

1. WSTĘP.

Praca ta zawiera podstawowe wiadomości, przydatne podczas pracy z systemem operacyjnym PC DOS w wersjach: 3.10 i 3.20 i została napisana na podstawie instrukcji "Disk Operating System Version 3.20 References" i "Microsoft MS-DOS version 3.10 Operating System". W stosunku do pierwszej wymienionej instrukcji wykonano następujące zmiany:

- dodano rozdział zawierający charakterystykę mikrokomputerów z rodziny IBM PC, zawierający m.in. dokładny opis funkcjonalny klawiatury,
- scharakteryzowano strukturę systemu operacyjnego PC DOS,
- zmieniono układ niektórych rozdziałów, dotyczy to zwłaszcza rozdziałów pt. 'ZBIÓR ROZKAZÓW PRZETWARZANIA WSA-DOWEGO' i 'PROGRAM KONSOLIDACJI ZBIORÓW LINK',
- pominięto rozdział dotyczący komunikatów generowanych przez system operacyjny PC DOS.

Praca nie zawiera szczegółowych informacji dotyczących wewnętrznej organizacji systemu operacyjnego PC DOS oraz danych dotyczących asemblera ASMB6. Informacje te można znaleźć w podanej literaturze.

W porównaniu do wersji PC DOS 3.10, wersja 3.20 została rozszerzona o rozkazy REPLACE i XCOPY. Dodano także program instalacyjny DRIVER.SYS (patrz rozdział pt. 'KONFIGUROWANIE SYSTEMU OPERACYJNEGO') oraz rozszerzono zakres czynności wykonywanych rozkazami: ATTRIB, COMMAND, DISKCOMP, DISKCOPY, FORMAT, SELECT i instrukcja SHELL. Wszystkie te zmiany zostały odnotowane w opisach wymienionych rozkazów.

Instrukcja może być przydatna także dla osób posługujących się systemem operacyjnym MS-DOS w wersji 3.10, niewiele różniącym się od systemu PC DOS 3.10. Różnice te zostały odnotowane w opisach poszczególnych rozkazów.

2. CHARAKTERYSTYKA MINIKOMPUTERÓW Z SERII IBM PC.

2.1. Wprowadzenie

Mikrokomputer IBM-PC pojawił się w sierpniu 1981 roku. Posiadał 64kB pamięci operacyjnej na pakiecie systemowym i 480kB pamięci na pakiecie dodatkowym. Jako pamięć masową wykorzystywano początkowo magnetofon, a później stacje dyskietkowe. Mikrokomputer IBM-PC/XT pojawił się w marcu 1983 roku. Posiadał 256kB pamięci operacyjnej na pakiecie systemowym i 384kB pamięci na pakiecie dodatkowym. Zrezygnowano w nim z wyjścia magnetofonowego, a jako pamięć masowej, oprócz stacji dyskietkowych, zaczęto używać dysków twardych typu WINCHESTER. Kolejne ulepszenie stanowił mikrokomputer IBM PC/AT. Zastosowano w nim szybszy mikroprocesor INTEL 8086 (poprzednio wymienione modele zawierały mikroprocesory INTEL 8088), pozwalający na efektywniejsze wykorzystywanie wszelkich operacji kosztem skomplikowania układów jednostki centralnej.

Mikrokomputery z rodziny IBM PC charakteryzują się dużą różnorodnością zestawu sprzętowego. Do ich standardowego wyposażenia można zaliczyć:

1. Pamięć operacyjną rozszerzoną do pełnego zakresu wynikającego z możliwości adresowych mikroprocesorów: INTEL 8086 i INTEL 8088, tj. do 1MB (z czego na potrzeby użytkownika do 640 kB). W niektórych modelach IBM PC/AT montowana jest rozszerzona pamięć operacyjna, do 4MB.
2. Stacje dyskietkowe jako wymienna pamięć masowa. Mogą być stosowane stacje dyskietkowe jednostronne, umożliwiające zapis/odczyt 160/180kB na pojedynczej dyskietce, stacje dyskietkowe dwustronne, umożliwiające zapis/odczyt 320/360kB i 720kB na pojedynczej dyskietce oraz stacje dyskietkowe o dużej gęstości zapisu, pozwalające korzystać z dyskietek o pojemności 1.2MB. Najczęściej spotykaną konfiguracją w mikrokomputerach IBM PC/XT są dwie stacje dwustronnego zapisu 320/360kB, natomiast w mikrokomputerach IBM PC/AT: po jednej stacji 320/360kB i 1.2Mb.

3. Stacje dysków twardych typu WINCHESTER jako stała pamięć masowa o pojemności 20MB i większej.
4. Koprocesor arytmetyczny INTEL 8087, pozwalający wykonywać, w kilku formatach danych, podstawowe operacje zmiennoprzecinkowe takie jak: dodawanie, odejmowanie, mnożenie, dzielenie, logarytmowanie oraz funkcje tangens i arcus tangens.
5. Klawiatura, najczęściej w układzie obowiązującym w USA.
6. Porty równoległe typu EPSON-CENTRONICS, umożliwiające równoległe podłączenie drukarek, porty asynchronicznej transmisji szeregowej, pozwalające na dwukierunkowy transfer danych z programowym ustawianiem parametrów transmisji.
7. Monitor monochromatyczny o rozdzielczości 720*348 punktów, tekst 25 linii po 80 znaków w linii (znak w matrycy 14*8 kropek).
8. Monitor kolorowy o rozdzielczości 640*200 punktów.
9. Urządzenia dodatkowe takie jak: głośnik, zegar bateryjny, joystick itp.

Podstawowe składniki komputera z rodziny IBM PC:

1. Pakiet podstawowy SBC (ang. Single Board Computer). Zawiera on mikroprocesor INTEL 8088 (8086) i ewentualnie koprocesor arytmetyczny INTEL 8087, pamięć RAM oraz pamięć ROM z programem BIOS i interpreterem języka BASIC, pakiety obsługi klawiatury, głośnika itp.
2. Pakiet HGC (ang. Hercules Graphic Card), przeznaczony do sterowania monitorem monochromatycznym i drukarką.
3. Pakiety FDC (ang. Floppy Disk Controller), przeznaczone do sterowania stacjami dysków elastycznych,
4. Pakiet IOC (ang. Input/Output Card), przeznaczony do obsługi układów wejścia/wyjścia oraz zegara.

2.2. Krótka charakterystyka mikroprocesorów INTEL 8086 i INTEL 8088.

Mikroprocesory INTEL 8086 (IBM PC/AT) i INTEL 8088 (IBM PC/XT) posiadają 16-bitowy arytmometr i 20-bitową szynę adresową (adresy od 00000H do FFFFFFFH). Oba wykonują te same funkcje, oba posiadają jednakowy zestaw instrukcji i oba mogą wykonywać te same programy. Podstawową różnicę między nimi stanowi szyna danych: INTEL 8088 korzysta z szyny 8-bitowej, INTEL 8086 z 16-bitowej. Skutkiem tego układ elektroniczny współpracujący z mikroprocesorem INTEL 8088 jest prostszy i tańszy niż w przypadku układu współpracującego z mikroprocesorem INTEL 8086, ale jednocześnie jest znacznie wolniejszy.

Rejestry ogólne (16 bitowe) to: AX, BX, CX, DX. Można je dzielić na 'porcje' 8 bitowe (AH, AL, BH, BL, CH, CL, DH, DL). Dodatkowo rejestr BX może być używany jako rejestr pomocniczy, a rejestr CX CX jako licznik.

Rejestry wskaźnikowe (16 bitowe) to:

- DI - indeks przeznaczenia
- SI - indeks źródłowy
- SP - wskaźnik stosu
- BP - wskaźnik bazowy

Rejestry segmentowe (16 bitowe) to:

- CS - rejestr segmentu programu
- DS - rejestr segmentu danych
- SS - rejestr segmentu stosu
- ES - rejestr segmentu dodatkowego

Rejestr specjalny IP - wskaźnik instrukcji

Rejestr flag - arytmometr umieszcza w nim informacje dla dekodera instrukcji. Flagi są następujące:

- | | |
|-----------------------------|-------------------------------------|
| 0-przeniesienie, | 7-znak, |
| 1-nie używany, | 8-praca krokowa, |
| 2-parzystość, | 9-zezwole nie na przerwania, |
| 3-nie używany, | 10-kierunek, |
| 4-pomocnicze przeniesienie, | 11-nadmiar (przekroczenie zakresu), |
| 5-nie używany, | 12-15-nie używane. |
| 6-zero, | |

Pamięć operacyjna jest adresowana przez mikroprocesor szyną adresową 20-bitową. Szesnastobitowy adres efektywny (wynikający z trybu adresowania) sumuje się z przesunięta o 4 bity 16-bitową zawartością odpowiedniego rejestru segmentu, tworząc 20-bitowy adres fizyczny. Rozkład oprogramowania w pamięci operacyjnej przedstawiono w podrozdziale pt. 'Rozkład oprogramowania w pamięci operacyjnej'.

2.3. Klawiatura

'Mózgiem' klawiatury jest mikroprocesor INTEL 8048. Zadaniem klawiatury jest:

- testowanie klawiatury na sygnał z jednostki centralnej,
- przyjmowanie klawiszy,
- komunikacja z komputerem (procesor 8088 + pamięć operacyjna).

Klawisze na klawiaturze są ponumerowane (dla 8048) od 1 do 83. Każdy klawisz ma swój numer, który za pośrednictwem mikroprocesora 8048 jest przesyłany do komputera (nie są wysyłane kody ASCII znaków odpowiadające klawiszom). Informacje dotyczą klawiszy aktualnie wciśniętych i wyciśniętych. W tym drugim przypadku mikroprocesor 8048 wysyła do komputera numer klawisza plus 128. Wciśnięcie klawisza powoduje wysłanie przerwania do mikroprocesora 8088, który podejmuje jego obsługę. Do obsługi przerwania spowodowanego wciśnięciem klawisza istnieją dwie procedury:

- przerwaniowa obsługa klawiatury,
- procedura przekazywania znaków programom.

8. SYSTEMY OPERACYJNE: PC DOS 3.10, 3.20 oraz MS DOS 3.10.

U W A G A :

Znak '/' umieszczony pomiędzy oznaczeniami dwóch klawiszy oznacza jednoczesne wciśnięcie tych klawiszy. Znak '-' umieszczony pomiędzy oznaczeniami dwóch klawiszy oznacza, że klawisze te należy wcisnąć w podanej kolejności, ale nie jednocześnie.

Klawisze na klawiaturze można podzielić na 4 grupy:

1. Klawisze funkcyjne, w kolorze szarym, znajdujące się z lewej strony klawiatury, oznaczone F1-F10. Nadawaniem ich funkcji zajmuje się program użytkowy.

- a. System operacyjny PC DOS nadaje im następujące znaczenie:

'F1' - odtwarza poprzedni rozkaz (znak po znaku),

'F2' - wraz z dowolnym znakiem z klawiatury ASCII odtwarza całą poprzednią linię, aż do napotkania tego znaku,

'F3' - odtwarza całą poprzednią linię,

'F4' - wraz z dowolnym znakiem z klawiatury ASCII odtwarza całą poprzednią linię od tego znaku do jej końca,

'F5' - rejestrowanie linii do poprawienia,

'F6' - wpisuje kod specjalny (kod kombinacji klawiszy 'Ctrl/Z' (1AH) oznaczający koniec zbioru tekstowego),

'F7' - wpisuje kod specjalny 00H, oznaczający koniec ciągu znaków w kodzie ASCII.

- b. Interpreter języka BASIC nadaje im następujące znaczenie:

- 'F1' - wyświetla na ekranie monitora instrukcję 'LIST', a po wciśnięciu klawisza 'Enter' listuje program,
- 'F2' - powoduje wystartowanie programu od jego początku,
- 'F3' - wyświetla na ekranie monitora instrukcję 'LOAD', należy podać wówczas nazwę programu, który po wciśnięciu klawisza 'Enter', zostanie załadowany z pamięci masowej do pamięci operacyjnej,
- 'F4' - wyświetla na ekranie monitora instrukcję 'SAVE', należy podać wówczas nazwę programu, który po wciśnięciu klawisza 'Enter', zostanie przepisany z pamięci operacyjnej do pamięci masowej,
- 'F5' - wyświetla na ekranie monitora instrukcję 'CONT'. Aby restartować program zatrzymany komendą STOP należy wówczas wcisnąć kombinację klawiszy 'Ctrl/Break',
- 'F6' - wyświetla na ekranie monitora 'LPT1',
- 'F7' - wyświetla na ekranie monitora instrukcję 'TRON'. Klawisz ten jest używany do listowania instrukcji programu aktualnie wykonywanej przez komputer,
- 'F8' - wyświetla na ekranie monitora instrukcję 'TROFF'. Klawisz ten powoduje skasowanie polecenia wydanego klawiszem 'F8',
- 'F9' - wyświetla na ekranie monitora instrukcję 'KEY'. Nacisnięcie tego klawisza zmienia znaczenie klawisza funkcyjnego wciśniętego bezpośrednio po nim,
- 'F10' - wyświetla na ekranie monitora instrukcję 'SCREEN'. Klawisz ten powoduje przejście monitora z trybu graficznego do trybu znakowego.

2. Klawisze ASCII, w kolorze jasnym, odpowiadające klawiaturze maszyny do pisania (alfabet, cyfry, znaki specjalne). Obsługa tej części klawiatury zajmuje się zasadni-

10. SYSTEMY OPERACYJNE: PC DOS 3.10, 3.20 oraz MS DOS 3.10.

czy program BIOS.

3. Klawisze ekranowe (lub numeryczne), zwykle w kolorze szarym, chociaż także jasnym.

- a. jeśli pracują jako klawisze ekranowe (ich obsługa zajmuje się program BIOS), to przesuwają kursor po ekranie monitora:

- '↑' - przesuwa kursor o jedną linię do góry,
- '↓' - przesuwa kursor o jedną linię w dół,
- '→' - przesuwa kursor o jedną pozycję w prawo,
- '←' - przesuwa kursor o jedną pozycję w lewo,
- 'Home' - przesuwa kursor w lewy górny róg ekranu,
- 'End' - przesuwa kursor na pozycję ostatniego znaku w linii,
- 'PgUp' - zastosowanie definiuje system operacyjny lub program użytkowy, zwykle przesuwa kursor na początek tekstu,
- 'PgDown' - zastosowanie definiuje system operacyjny lub program użytkowy, zwykle przesuwa kursor na koniec tekstu.

- b. jeśli pracują jako klawisze numeryczne to spełniają funkcje jak klawisze kalkulatora: '0-9', '+', '-', '.', 'CR'.

4. Klawisze specjalne:

- a. 'Ctrl' - w stosunku do klawiszy z klawiatury ASCII jego obsługa jest zawarta w programie BIOS. Służy on do generowania kodów ASCII z zakresu od 00 do 1FH, czyli kodów funkcyjnych:

'Ctrl/G' generuje kod 00H - NULL,

'Ctrl/G' generuje kod 07H - sygnał dźwiękowy,

- 'Ctrl/H' generuje kod 08H - usuń znak,
- 'Ctrl/I' generuje kod 09H - tabulacja pozioma,
- 'Ctrl/J' generuje kod 0AH - przejście do następnej linii,
- 'Ctrl/K' generuje kod 0BH - tabulacja pionowa,
- 'Ctrl/L' generuje kod 0CH - przejście do następnej strony,
- 'Ctrl/M' generuje kod 0DH - powrót do pierwszej kolumny,
- 'Ctrl/Z' generuje kod 1AH - koniec zbioru tekstowego,
- 'Ctrl/C' generuje kod 1BH - ESC kody specjalne.

Klawisz 'Ctrl' jest bez blokady, tzn. nie może być zablokowany w pozycji 'wciśnięty na stałe'.

- b. 'Alt' - klawisz ten służy do wprowadzania znaków ASCII poprzez podawanie ich dziesiętnych kodów z klawiatury numerycznej od najstarszej cyfry. Można go także wykorzystywać wraz z dowolnym klawiszem, oznaczonym literą od A do Z, do wprowadzania instrukcji języka BASIC.
- c. 'Shift' (prawy i lewy) - powoduje przejście do górnego pocztu znaków (duże litery, 'górne' znaki z klawiszy 'cyfrowych'). Klawisz ten jest bez blokady tzn. nie może być zablokowany w pozycji 'wciśnięty na stałe'.
- d. 'Caps Lock' - powoduje przejście do górnego pocztu znaków, ale tylko w stosunku do klawiszy odpowiadających literom. Klawisz ten można zablokować w pozycji 'wciśnięty' (funkcje realizowane przez BIOS).

- e. 'Num Lock' - klawisz ten służy do przełączania funkcji klawiszy z klawiatury numerycznej: pierwsze wciśnięcie pozwala klawiszami do sterowania kursorem wprowadzać cyfry, drugie wciśnięcie powoduje powrót do funkcji sterowania kursorem klawiszami klawiatury numerycznej itd. (funkcje realizowane przez BIOS).
- f. 'Scroll Lock' - pierwsze wciśnięcie powoduje że kursor zawsze pozostaje w tej samej linii na ekranie monitora, a klawisze przesuujące kursor 'w górę' lub 'w dół' tekstu przesuują teraz sam tekst względem linii z kursorem. Drugie wciśnięcie to powrót do poprzedniego trybu pracy (funkcje te nie są realizowane wewnątrz BIOSA).
- g. 'Ins' - klawisz ten jest wykorzystywany do sterowania wstawianiem danych w linię tekstu. Jego wciśnięcie powoduje ustawienie trybu pracy polegającego na tym, że wstawienie znaku w miejsce, w którym stoi kursor powoduje jednocześnie przesunięcie wszystkich znaków na prawo od kursora o jeden znak w prawo (w zakresie linii z kursorem). Powtórne wciśnięcie tego klawisza ustawia mod pracy polegający na zastępowaniu znakiem wpisywanym znaku na którym stoi kursor.
- h. 'Del' - wciśnięcie tego klawisza kasuje znak na którym stoi kursor z jednoczesnym przesunięciem tekstu linii na prawo od kursora o jedną pozycję w lewo.
- i. 'Esc' - w systemie operacyjnym PC DOS wciśnięcie tego klawisza usuwa linię lub instrukcję.
- j. '←' - wciśnięcie tego klawisza usuwa pierwszy znak z lewej strony kursora przesuując jednocześnie cały tekst z prawej strony kasowanego znaku o jedną pozycję w lewo.
- k. '⇐' - klawisz tabulacji poziomej (punkt zatrzymania co 8 znaków).

1. 'Enter' - wciśnięcie tego klawisza powoduje powrót kursora do pierwszej kolumny (wysłanie instrukcji do komputera).

Kombinacje klawiszy:

1. 'Ctrl/Alt/Del' - programowy restart systemu operacyjnego, wykonywany przez program BIOS.
2. 'Ctrl/Num Lock' - zatrzymanie wykonywania programu lub instrukcji. Aby kontynuować zawieszoną czynność należy wcisnąć dowolny klawisz z klawiatury.
3. 'Ctrl/Scroll Lock' - zatrzymanie wykonywania programu i identyfikacja linii instrukcji, na której nastąpiło to zatrzymanie.
4. 'Ctrl/Home' - usuwanie całej zawartości ekranu i przemieszczenie kursora do lewego górnego rogu.
5. 'Ctrl/PrtSc' - wszystko to, co będzie wyświetlane na ekranie, będzie także wysyłane na drukarkę (DOS).
6. 'Shift/PrtSc' - wydrukowanie aktualnej zawartości ekranu (BIOS).
7. 'Ctrl/Enter' - generowanie kodu OAH - nowa linia (DOS).
8. 'Ctrl/S' - zatrzymanie wyświetlania tekstu na ekranie (zawieszenie wykonywania programu, który wyświetla tekst na ekranie). Wznowienie nastąpi po wciśnięciu klawisza dowolnego znaku.

Polska klawiatura:

Aby można było korzystać z polskiego alfabetu trzeba:

1. Zmodyfikować procedury BIOSA obsługujące klawiaturę, a więc zmodyfikować znaczenie poszczególnych klawiszy według normy PN-79/F-02000 (norma dotycząca klawiatury).
2. Wprowadzić tzw. martwy klawisz, czyli możliwość drukowania znaku bez przesuwu karetki, a więc dodrukowywanie 'ogonków' do typowo polskich liter. Twórcy IBM-a rozszerzyli standardowy kod ASCII o następne 128 znaków, aby można było wprowadzać znaki charakterystyczne dla różnych alfabetów. I tak wprowadzono:

14. SYSTEMY OPERACYJNE: PC DOS 3.10, 3.20 oraz MS DOS 3.10.

- tzw. tablicę łacina I - zawierająca znaki alfabetów używanych w Europie Zachodniej,
- tzw. tablicę łacina II - zawierająca znaki alfabetów używanych w Europie Wschodniej (kody od 128 do 255).

Przejdzie z tablicy łacina II do tablicy łacina I powoduje kombinacja klawiszy: 'Ctrl/ALT/F1', a z tablicy łacina I do tablicy łacina II - kombinacja klawiszy: 'Ctrl/ALT/F2'.

3. Aby można było wyświetlać zmodyfikowane znaki na ekranie monitora trzeba przeprogramować generator znaków (znajduje się na pakiecie HGC).
4. Aby można było drukować zmodyfikowane znaki konieczna jest także modyfikacja generatora znaków umieszczonego w drukarce.

3. WPROWADZENIE DO SYSTEMU OPERACYJNEGO PC DOS.

3.1. Struktura systemu operacyjnego PC DOS.

System operacyjny PC DOS składa się z 6 zasadniczych części, czyli programów (zespołów procedur):

1. Program BIOS (ang. Basic Input/Output System) stanowiący integralną część każdego systemu operacyjnego, funkcjonującego w mikrokomputerach z rodziny IBM PC. Program BIOS jest zapisany w pamięci stałej typu ROM od adresu FE000H do adresu FFFFFH i zawiera szereg procedur oraz tablic z danymi. Program BIOS po włączeniu zasilania albo po wykonaniu restartu systemu operacyjnego wciśnięciem kombinacji klawiszy 'Ctrl/Alt/Del' przejmuje kontrolę nad systemem. Wykonywane są wtedy testy pamięci typu RAM oraz urządzeń podłączonych do jednostki centralnej. Następnie z pamięci masowej do pamięci operacyjnej zostaje ściągnięty program o nazwie BOOTSTRAP, służący do wykonywania restartu systemu operacyjnego. Jeśli w pamięci masowej nie ma założonego nośnika z programem BOOTSTRAP lub jeśli wystąpił błąd odczytu tego programu, wtedy program BIOS przekazuje kontrolę nad systemem interpreterowi języka BASIC, także zapisanemu w pamięci stałej typu ROM. Podczas pracy systemu operacyjnego program BIOS zajmuje się obsługą urządzeń peryferyjnych takich jak klawiatura, monitor ekranowy, stacje dysków miękkich, asynchronicznych portów szeregowych i portów równoległych.
2. Program BOOTSTRAP - umieszczany zawsze w stałym miejscu pamięci masowej: na dysku elastycznym - ślona 0, ścieżka 0, sektor 1, na dysku twardym typu WINCHESTER - 1 cylinder partycji DOSa. Program BOOTSTRAP, zajmujący 512 bajtów (1 sektor), jest zbyt mały, aby wykonywać samodzielnie skomplikowane czynności, toteż jego podstawowym zadaniem jest odszukanie tzw. zbiorów ukrytych (ang. hidden files) systemu operacyjnego PC DOS: IBMBIO.COM i IBMDOS.COM, zapisanych w z góry określonych miejscach w pamięci masowej i ściągnięcie ich do pamięci operacyjnej. Programy BOOTSTRAP z różnych wersji systemów opera-

cyjnych mogą się nieznacznie różnić od siebie. Różnica między nimi wynika z faktu, że zmianom ulegają przede wszystkim zbiory systemowe, ściągane programem BOOTSTRAP do pamięci operacyjnej, a więc zmienia się ich długość i miejsce zapisu.

3. Program IBMBIO.COM - stanowiący rozszerzenie i uzupełnienie programu BIOS. Jego zadaniem jest pośredniczenie pomiędzy procedurami BIOSa i programem IBMDOS.COM. Ponadto zajmuje się korygowaniem błędów programu BIOS. Program BIOS, przed zapisaniem w pamięci stałej typu ROM, poddawany jest bardzo dokładnej analizie, ale po zapisaniu nie może być już zmieniany. Jeśli wystąpią w nim jakiegokolwiek błędy, to ich korygowanie można wykonać tylko zmieniając program IBMDOS.COM (pod pojęciem 'błędy' nie należy tutaj rozumieć błędów logicznych czy tym bardziej składniowych, lecz ograniczone możliwości działania w stosunku do zmian sprzętowych i programowych). Program IBMBIO.COM zajmuje się więc obsługą dalszych urządzeń peryferyjnych takich jak: stacje dysków twar- dych, stacji dyskietkowych o dużej gęstości zapisu, plotterów itp. Ponadto, podczas restartu systemu operacyjnego, program IBMBIO.COM dodatkowo wykonuje obróbkę zbioru CONFIG.SYS, kształtującego konfigurację sprzętową systemu (dołączanie urządzeń indywidualnych).
4. Program IBMDOS.COM - zajmujący się wpisywaniem adresów procedur użytkowych, inicjalizowaniem zmiennych i ściąganiem do pamięci operacyjnej programów umieszczonych w zbiorze COMMAND.COM. Ponadto zarządza on zbiorami na dyskach twardych i dyskietkach, tj. kontroluje operowanie zbiorami i danymi w zbiorach.
5. Program COMMAND.COM - będący interpreterem rozkazów systemu operacyjnego PC DOS. Program ten dzieli się na dwie części:
 - część rezydentna (trwała), pozostająca w pamięci przez cały czas pracy systemu operacyjnego. Zajmuje się ona obsługą procedur użytkowych (przerwania: INT 22, INT 23, INT 24) oraz powtórным ładowaniem do pamięci części nietrwalej.
 - część nietrwała, używana podczas restartu systemu operacyjnego do kontroli i wykonywania rozkazów zawartych w zbiorze AUTOEXEC.BAT. Zawiera ona programy obsługi rozkazów wewnętrznych i interpreter rozkazów systemu operacyjnego PC DOS (wykonywanie rozkazów wewnętrznych oraz ściąganie do pamięci operacyjnej rozkazów zewnętrznych i ich wykonywanie).

6. Rozkazy zewnętrzne - są to programy zapisane w pamięci masowej, nie stanowiące integralnej części systemu operacyjnego PC DOS, lecz jego uzupełnienie. Rozkazami tymi są m.in. DISKCOMP, FORMAT, EDLIN, BASIC etc.

3.2. Rozkład oprogramowania w pamięci operacyjnej

Adres początkowy	Pamięć operacyjna po inicjalizacji systemu operacyjnego PC DOS	Pamięć operacyjna w momencie wykonywania rozkazów zewnętrznych
00000H	adresy procedur użytkowych programu BIOS	adresy procedur użytkowych programu BIOS
00080H	miejsce na adresy procedur	miejsce na adresy procedur
00400H	zmienne, z których korzysta program BIOS	zmienne, z których korzysta program BIOS
00500H	zmienne systemu operacyjnego PC DOS	zmienne systemu operacyjnego PC DOS
00600H	część rezydentna programu IBMBIO.COM	część rezydentna programu IBMBIO.COM
(*)	program IBMDOS.COM (najważniejsze procedury użytkowe)	program IBMDOS.COM (najważniejsze procedury użytkowe)
(*)	część rezydentna programu COMMAND.COM	część rezydentna programu COMMAND.COM
(*)	.	rozkaz zewnętrzny, czyli program użytkowy
(*)	część nietrwała programu COMMAND.COM	część nietrwała programu COMMAND.COM

Ciąg dalszy na następnej stronie...

Rozkład oprogramowania w pamięci operacyjnej c.d.

Adres początkowy	Pamięć operacyjna po inicjalizacji systemu operacyjnego PC DOS	Pamięć operacyjna w momencie wykonywania rozkazów zewnętrznych
A0000H	rozszerzenie pamięci ekranu	rozszerzenie pamięci ekranu
B0000H	pamięć ekranu dla kontrolerów tekstowych	pamięć ekranu dla kontrolerów tekstowych
BB000H	pamięć ekranu dla kontrolerów grafiki kolorowej	pamięć ekranu dla kontrolerów grafiki kolorowej
C0000H	procedury obsługi pamięci ekranu	procedury obsługi pamięci ekranu
CB000H	procedury obsługi dysków stałych	procedury obsługi dysków stałych
CC000H	ew. użytkowe uzupełnienia programu BIOS	ew. użytkowe uzupełnienia programu BIOS
F6000H	interpreter języka BASIC	interpreter języka BASIC
FE000H	program BIOS	program BIOS
FFFF0H	restart systemu operacyjnego po włączeniu zasilania	restart systemu operacyjnego po włączeniu zasilania

(*) - w tym miejscu adres początkowy może być różny.

3.3. Rozpoczynanie, kontynuacja i przerywanie pracy z systemem operacyjnym PC DOS.

Rozpoczynanie pracy z systemem PC DOS:

- restart 'zimny' - włączenie zasilania
- restart 'gorący' - jednoczesne wciśnięcie klawiszy 'Ctrl/Alt/Del'

- testy urządzeń składowych mikrokomputera (zajmują się tym procedury BIOSA)
- inicjalizacja systemu operacyjnego, czyli ściąganie do pamięci operacyjnej programu BOOTSTRAP (przez procedurę BIOSA)

Prace w systemie PC DOS można rozpoczynać po pojawieniu się komunikatu:

A>

Postać tego komunikatu można zmieniać modyfikując odpowiednio zbiór AUTOEXEC.BAT. Litera A oznacza tutaj pierwszą stację dysków miękkich A: (w przypadku dysku twardego typu WINCHESTER będzie to litera C). Zgłoszenie systemu już po wykonaniu restartu można zmodyfikować, wykorzystując rozkaz PROMPT (patrz opis).

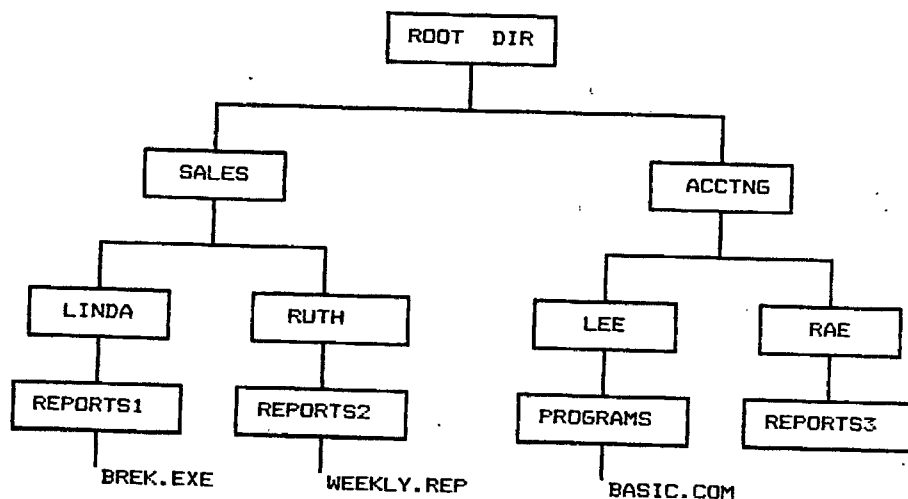
W instrukcji często pojawia się określenie 'urządzenie robocze' ('stacja robocza'). Pod tym pojęciem należy rozumieć to urządzenie, które domyślnie przyjmuje system operacyjny PC DOS, jeśli w linii komendy jakiegoś rozkazu zostanie pominięty opcjonalny parametr określający nazwę urządzenia. Urządzenie robocze można zmienić, po wyświetleniu zgłoszenia systemu, poprzez podanie oznaczenia literowego tego urządzenia, znaku dwukropka (:) oraz kończąc to wciśnięciem klawisza 'Enter' (A: oznacza pierwszą stację dysków miękkich, B: oznacza drugą stację dysków miękkich, C: oznacza pierwszą stację dysku twardego itp.).

Przerywanie pracy z mikrokomputerem:

- poczekać na zgłoszenie się systemu operacyjnego (ewentualnie poczekać aż zgaśnie dioda kontrolna stacji dysków),
- usunąć dyski elastyczne ze stacji,
- wyłączyć zasilanie monitora i mikrokomputera.

3.4. Katalogi w systemie operacyjnym PC DOS.

Katalogi w systemie operacyjnym PC DOS są zorganizowane w tzw. drzewa katalogowe. Organizacja tego typu pozwala na efektywny dostęp do wszystkich zbiorów, gdyż umożliwia tworzenie np. katalogów tematycznych, zawierających zbiory o określonym przeznaczeniu, zbiory należące do jednego użytkownika etc. W danym katalogu mogą być także zapisane zbiory będące katalogami podrzędnymi w stosunku do tego katalogu. Przykład drzewa katalogowego przedstawiono na poniższym rysunku:



Istnieją dwa następujące typy katalogów:

1. Katalogi podstawowe (ang. root directory), tworzone na każdym dysku lub dyskietce podczas formatowania. Katalog podstawowy jest katalogiem nadrzędnym w stosunku do każdego innego katalogu na dysku lub dyskietce (na każdym z tych nośników może znajdować się tylko jeden katalog podstawowy). W katalogach podstawowych mieści się ograniczona ilość zbiorów, zależnie od zaformatowania i rodzaju nośnika. I tak:
 - katalog podstawowy na dyskietce zaformatowanej jednostronnie, o pojemności 160/180kB, może pomieścić do 64 zbiorów i katalogów podrzędnych,
 - katalog podstawowy na dyskietce zaformatowanej dwu-

stronnie, o pojemności 320/360kB, może pomieścić do 112 zbiorów i katalogów podrzędnych,

- katalog podstawowy na dyskietce o dużej gęstości zapisu, o pojemności 1.2MB, może pomieścić do 224 zbiorów i katalogów podrzędnych,
- katalog podstawowy na dysku twardym może pomieścić do 512 zbiorów i katalogów podrzędnych.

Podane liczby dotyczą tylko pojemności katalogu podstawowego, a nie ilości wszystkich zbiorów i katalogów podrzędnych możliwych do zapisania na danym nośniku. Katalogi podstawowe są oznaczane znakiem '\' (ang. backslash).

2. Katalogi podrzędne, tworzone przez użytkownika i oznaczone przez niego nazwą. Nazwa katalogu podrzędnego może składać się od 1 do 8 znaków, za wyjątkiem:

. " / \ [] : ! < > + = ; ,

oraz żadnego ze znaków ASCII o kodzie mniejszym niż 20H.

Każdy katalog podrzędny może zawierać dowolną liczbę zbiorów i innych podkatalogów, ograniczoną tylko pojemnością nośnika. Ponadto każdy katalog podrzędny zawsze zawiera dwa następujące zbiory specjalne:

- zbiór o nazwie . <DIR> (słowo <DIR> oznacza, że dany zbiór jest katalogiem), pozwalający na szybkie odwołanie się do katalogu roboczego na danym urządzeniu, np. rozkaz ERASE . spowoduje usunięcie wszystkich zbiorów z katalogu roboczego na danym urządzeniu.
- zbiór o nazwie .. <DIR> pozwalający na szybkie zlokalizowanie katalogu nadrzędnego (macierzystego) w stosunku do katalogu roboczego, np. jeśli katalogiem roboczym jest katalog LINDA, to do zbioru WEEKLY.REP, zapisanego w katalogu REPORTS2 można dotrzeć na dwa sposoby:

\SALES\RUTH\REPORTS2 WEEKLY.REP

..\RUTH\REPORTS2 WEEKLY.REP

Oprócz tego, dość często używanym określeniem jest tzw. 'katalog roboczy' (ang. default directory). Katalog roboczy jest to taki katalog, który domyślnie przyjmuje system operacyjny PC DOS, jeśli w linii komendy jakiegoś rozkazu czy instrukcji nie podano nazwy katalogu. Każde urządzenie umożliwiające zapis/odczyt informacji (dyski twarde, dyskietki, tzw dyski wirtualne) w każdej chwili posiada swój własny katalog roboczy. Nazwę katalogu roboczego można zmieniać rozkazem CHDIR (patrz opis). Po wykonaniu restartu systemu operacyjnego, katalogi robocze wszystkich urządzeń stają się automatycznie katalogami podstawowymi.

3.4.1. Zmiana katalogu roboczego.

Zmianę katalogu roboczego szczegółowo opisano w rozdziale dotyczącym rozkazu CHDIR. Tutaj zostaną jedynie przedstawione przykłady wykorzystujące pokazaną wcześniej strukturę drzewa katalogowego.

Jeśli katalogiem roboczym na urządzeniu C: jest katalog podstawowy, a użytkownik potrzebuje np. zmodyfikować zbior WEEKLY.REP, to aby dotrzeć do niego trzeba zmienić katalog roboczy następującym rozkazem CHDIR:

```
C>CHDIR \SALES\RUTH\REPORTS2
```

Jeśli katalogiem roboczym jest katalog RUTH, to aby stał się nim katalog REPORTS1 należy użyć następującego rozkazu CHDIR:

```
C>CHDIR REPORTS2
```

Aby przywrócić katalog podstawowy jako katalog roboczy, należy użyć następującego rozkazu CHDIR:

```
C>CHDIR \
```

Taka forma przywrócenia katalogu podstawowego jako katalogu roboczego obowiązuje, bez względu na to jakim katalogiem jest obecny katalog roboczy.

Należy zwrócić uwagę, że specyfikacja ciągu katalogów, tak jak w pierwszym powyższym przykładzie, nie może zawierać więcej niż 63 znaki.

3.4.2. Określenie ścieżki-do-zbioru.

Z pojęciem katalogu roboczego ściśle wiąże się pojęcie tzw. ścieżki-do-zbioru (ang. path to a file). System operacyjny PC DOS pozwala na odwoływanie się do zbiorów zapisanych poza katalogiem roboczym bez zmiany tego katalogu. Jest to możliwe przez wyspecyfikowanie od katalogu roboczego wszystkich katalogów, aż do 'osiągnięcia' katalogu z poszukiwanym zbiorem. Przykładem takiej ścieżki-do-zbioru WEEKLY.REP z katalogu podstawowego jest:

```
\SALES\RUTH\REPORTS2
```

3.4.3. Zastosowanie rozkazu PATH.

Rozkaz PATH został omówiony szczegółowo w rozdziale pt. 'ROZKAZY SYSTEMU OPERACYJNEGO PC DOS'. Tutaj opis zostanie ograniczony do najważniejszych punktów.

Rozkaz PATH służy do przekazania systemowi operacyjnemu informacji, gdzie należy poszukiwać programów (czyli zbiorów o rozszerzeniu .EXE, .BAT lub .COM, patrz opis w dalszej części), jeśli nie znajdują się one w katalogu roboczym. Poszukiwania takie system operacyjny PC DOS będzie wykonywał w kolejnych, wyspecyfikowanych w danej ścieżce-do-zbioru katalogach, przy czym ścieżka do zbioru może rozpoczynać się albo z katalogu podstawowego (wtedy na początku występuje znak '\') albo z katalogu roboczego. Jeśli rozkazy np. BASIC.COM i BREK.EXE zapisane na urządzeniu C: odpowiednio w katalogach PROGRAMS i REPORTS1, mają być wywoływane z dowolnego katalogu roboczego, to przy istniejącej strukturze drzewa katalogowego, należy przedtem zainicjować wykonanie następującego rozkazu PATH:

```
C>PATH C:\SALES\LINDA\REPORTS1;C\ACCTNG\LEE\PROGRAMS
```

3.4.4. Tworzenie katalogów podrzędnych.

Katalogi podrzędne mogą być tworzone rozkazem MKDIR (patrz opis). Idea utworzenia katalogu podrzędnego, w danym katalogu, z innego katalogu polega na wyspecyfikowaniu, za słowem kluczowym MKDIR, ścieżki-do-zbioru z katalogu roboczego do tego katalogu, w którym ma zostać utworzony katalog podrzędny. Ostatnim ogniwem tej ścieżki jest nazwa tworzonego katalogu.

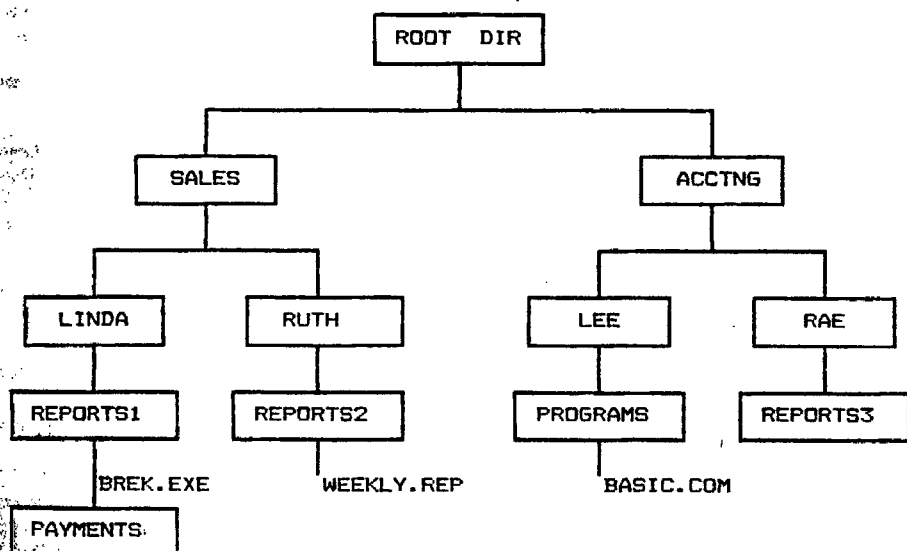
Przykładowo przyjmując opisana wcześniej strukturę drzewa katalogowego, aby z katalogu podstawowego jako katalogu roboczego utworzyć podkatalog PAYMENTS w katalogu REPORTS1 należy użyć następującego rozkazu MKDIR:

```
C>MKDIR \SALES\LINDA\REPORTS\PAYMENTS
```

Jeśli katalogiem roboczym jest katalog REPORTS1 to wystarczy następujący rozkaz MKDIR:

```
C>MKDIR PAYMENTS
```

W wyniku wykonania jednego z powyższych rozkazów aktualna struktura drzewa katalogowego będzie miała następującą postać:



Dość często katalog podrzędny można utworzyć w każdym katalogu, a nie tylko na 'najniższym' poziomie drzewa katalogowego.

3.4.5. Usuwanie katalogów podrzędnych.

Katalogi podrzędne można usuwać tylko rozkazem RMDIR (patrz opis). Usuwany katalog podrzędny musi być pusty, tzn. nie może zawierać żadnego zbioru ani katalogu podrzędnego. Oznacza to, że listowanie jego zawartości rozkazem DIR wykazuje tylko obecność dwóch zbiorów specjalnych:

. <DIR>

.. <DIR>

Katalog podrzędny można usuwać z bezpośredniego, roboczego katalogu nadrzędnego w stosunku do tego katalogu, lub z dowolnego katalogu o kilka poziomów 'wyższego'. W tym drugim przypadku usuwany katalog jest ostatnim wyspecyfikowanym katalogiem w ścieżce do zbioru. Katalogów: systemowego i roboczego nie można usunąć.

Aby przywrócić pierwotną strukturę drzewa katalogowego, zmodyfikowaną omówionymi w poprzednim podrozdziale rozkazami MKDIR, należy użyć następujących rozkazów RMDIR:

- jeśli katalogiem roboczym jest katalog podstawowy to:

```
C>RMDIR\SALES\LINDA\REPORTS\PAYMENTS
```

- jeśli katalogiem roboczym jest katalog REPORT1, to:

```
C>RMDIR PAYMENTS
```

W obu przypadkach usunięty zostanie katalog podrzędny PAYMENTS.

3.4.6. Wyświetlanie informacji dotyczących aktualnego katalogu roboczego i aktualnej struktury drzewa katalogowego.

Dla wyświetlenia aktualnego katalogu roboczego na danym urządzeniu można posłużyć się rozkazem CHDIR, wywoływanym tylko ze specyfikacją tego urządzenia (lub bez żadnych parametrów jeśli chodzi o uzyskanie informacji dotyczących urządzenia roboczego).

Aktualną strukturę drzewa katalogowego na danym urządzeniu, wraz z informacjami dotyczącymi zawartości poszczególnych katalogów, można uzyskać przy pomocy rozkazu TREE.

Oba te rozkazy omówiono szczegółowo w rozdziale pt. 'ROZKAZY SYSTEMU OPERACYJNEGO PC DOS', toteż w tym miejscu nie będą przedstawiane.

3.5. Zbiory w systemie operacyjnym PC DOS.

System operacyjny PC DOS zajmuje się między innymi:

- przechowywaniem zbiorów,
- obróbką zbiorów,
- wykonywaniem rozkazów zewnętrznych (zbiorów programowych).

3.5.1. Nazwy zbiorów w systemie operacyjnym PC DOS.

Nazwa zbioru pozwala systemowi operacyjnemu PC DOS na znalezienie tego zbioru. Pełny format nazwy jest następujący:

(stacja:)(ścieżka)(nazwa zbioru).(rozszerzenie)

gdzie:

1. Stacja określa urządzenie na którym dany zbiór znajduje się, lub na które zostanie przesłany. Określenie stacji polega na podaniu pojedynczej litery oznaczającej dane urządzenie i dwukropka za tą literą np. A:. W nazwie można nie podawać stacji, system operacyjny domyślnie przyjmie wówczas stację roboczą.
2. Ścieżka określa drogę, którą można dotrzeć do danego zbioru z aktualnego katalogu roboczego. Ścieżka składa się z ciągu do 63 znaków będących nazwami kolejnych katalogów oddzielonych od siebie znakiem '\'. Jeśli zaczyna się od znaku '\', to system operacyjny rozpoczyna określenie drogi do poszukiwanego zbioru od katalogu podstawowego, w pozostałych przypadkach od katalogu roboczego.
3. Nazwa zbioru (dalej występująca po prostu jako 'nazwa') określa konkretny zbiór na danym nośniku i w danym katalogu. Nazwa może tworzyć ciąg od jednego do osmiu znaków nie zawierający żadnego z następujących znaków:

. " / \ [] : ! < > + = ; ,

ani żadnego ze znaków ASCII o kodzie mniejszym niż 20H.

4. Rozszerzenie jest ciągiem od zera do trzech znaków i pozwala na łatwiejszą identyfikację przeznaczenia danego zbioru. Rozszerzenie nie może zawierać żadnego z następujących znaków:

" / \ [] : ! < > + = ; ,

ani żadnego ze znaków ASCII o kodzie mniejszym niż 20H.

3.5.2. Znaki specjalne.

W systemie operacyjnym PC DOS istnieje możliwość jednoczesnego odwoływania się do grup zbiorów. Do tego celu służą dwa znaki: '?' i '*'. Znak '?' umieszczony wewnątrz nazwy zbioru lub jego rozszerzenia zastępuje dowolny znak, który powinien znajdować się w tym miejscu. Przykładowo: AB?D.EFG jest równoważne wymienieniu np. ABCD.EFG, ABDD.EFG, AB1D.EFG itd. Podobnie, znak '*' umieszczony wewnątrz nazwy zbioru lub jego rozszerzenia (albo nawet zamiast nazwy lub rozszerzenia) zastępuje każdy znak, który powinien znajdować się w tym miejscu i wszystkie pozostałe znaki na prawo od tego miejsca. Przykładowo: AB*.XYZ jest równoważne wymienieniu np. ABCDE.XYZ, ABC357.XYZ, ABIDE.XYZ, ABIIOU.XYZ, AB.XYZ itd.

3.5.3. Specjalne rozszerzenia.

W systemie operacyjnym PC DOS istnieją trzy specjalne rozszerzenia dla nazw zbiorów:

- rozszerzenie .COM dla zbiorów będących programami całkowicie relokowalnymi,
- rozszerzenie .EXE dla zbiorów będących programami nierelokowalnymi,
- rozszerzenie .BAT dla zbiorów zawierających zestawy rozkazów przeznaczonych do przetwarzania wsadowego.

3.5.4. Nazwy zastrzeżone w systemie operacyjnym PC DOS.

W systemie operacyjnym PC DOS istnieje kilka nazw zastrzeżonych tylko dla urządzeń zewnętrznych. Są to:

- CON: oznacza wejście z konsoli (wejście standardowe) oraz wyjście na konsolę (wyjście standardowe),
- AUX: lub COM1 oznacza pierwsze łącze szeregowe RS-232C,
- COM2: oznacza drugie łącze szeregowe RS-232C,
- PRN: lub LPT1: oznacza pierwsze łącze równoległe CENTRONICS (działające tylko jako wyjście),
- LPT2: oznacza drugie łącze równoległe CENTRONICS (działające tylko jako wyjście),
- NUL: oznacza urządzenie nieistniejące, fizycznie stosowane do testowania.

3.6. Wtórna definicja standardowych urządzeń wejściowych i wyjściowych

Po każdym restarcie system operacyjny DOS jako standardowe wejście traktuje klawiaturę konsoli, a jako standardowe wyjście - ekran monitora konsoli. Ponieważ aplikacja konkretnego programu może zakładać, że np. dane do tego programu będą czytane ze zbioru znajdującego się na dysku i wyniki mają być drukowane na drukarce a nie wyświetlane na ekranie, więc system operacyjny zapewnia możliwość tzw. wtórnej definicji konsoli (ang. redirect).

Wtórna definicję standardowych urządzeń wejściowych i wyjściowych można wykonać dopisując do instrukcji DOSa następujące ciągi:

1. >[d:][ścieżka]nazwa[.rozszerzenie]

Ciąg ten spowoduje utworzenie [na wskazanym urządzeniu d:] zbioru o wyspecyfikowanej nazwie [i rozszerzeniu] [do którego dostęp z aktualnego katalogu roboczego wyznacza wyspecyfikowana ścieżka] w którym będzie zapisywane wszystko to, co normalnie byłoby wyświetlane na ekranie monitora konsoli.

2. >>[d:][ścieżka]nazwa[.rozszerzenie]

Ciąg ten spowoduje dopisanie (lub utworzenie, jeśli nie istnieje) do zbioru [na wskazanym urządzeniu d:] o wyspecyfikowanej nazwie [i rozszerzeniu] [do którego dostęp z aktualnego katalogu roboczego wyznacza wyspecyfikowana ścieżka] tego wszystkiego, co normalnie byłoby wyświetlane na ekranie monitora konsoli.

3. <[d:][ścieżka]nazwa[.rozszerzenie]

Ciąg ten spowoduje, że wszystko to, co powinno być czytane z klawiatury konsoli będzie teraz czytane ze zbioru o wyspecyfikowanej nazwie [i rozszerzeniu] [znajdującego się na wskazanym urządzeniu d:] [do którego dostęp z aktualnego katalogu roboczego wyznacza wyspecyfikowana ścieżka].

Przykłady:

DIR >LPT1:

- zawartość katalogu roboczego będzie drukowana na drukarce podłączonej do wyjścia równoległego LPT1:

DEBUG <ROZKAZ.TXT >LPT1:

- program debuggera będzie pobierał rozkazy ze zbioru ROZKAZ.TEX, a efekt działania wysyłał na drukarkę podłączoną do wyjścia równoległego LPT1:.

3.7. Przetwarzanie potokowe.

Przetwarzanie potokowe (ang. piping) w systemie operacyjnym PC DOS pozwala na traktowanie standardowych wyjść jednego programu jako standardowych wejść innego programu. W tym celu system operacyjny tworzy tymczasowe, pomocnicze zbiory pośrednie na urządzeniu roboczym, w katalogu roboczym. Programy, które będą poddawane przetwarzaniu potokowemu nie mogą ani kasować, ani modyfikować tych zbiorów. Przetwarzanie potokowe jest więc ciągiem programów wykonujących automatycznie wtórną definicję standardowych wejść i wyjść systemu operacyjnego. Nazwy programów występujących w linii komendy muszą być oddzielone od siebie znakiem '|'.
.

Przykład:

Niech program o nazwie SORT dokonuje przeglądu i modyfikacji konta użytkownika na urządzeniu A:. Wówczas odpowiednia komenda uruchamiająca ten program będzie miała następująca postać:

```
A>DIR | SORT
```

Rezultat działania programu SORT będzie przesłany na standardowe wyjście tj. monitor ekranowy konsoli. Jeśli komenda będzie miała postać:

```
A>DIR | SORT FILE.TXT
```

wówczas na urządzeniu roboczym, w katalogu użytkownika będzie utworzony zbiór o nazwie FILE.TXT zawierający wyniki działania programu SORT.

Oczywiście w linii komendy można podać nie dwa, a więcej nazw programów, które będą poddane przetwarzaniu potokowemu.

3.8. Filtry w systemie operacyjnym PC DOS.

Filtrem w systemie operacyjnym PC DOS nazwano programy lub komendy, które czytają dane ze standardowego wejścia (zazwyczaj z klawiatury konsoli), modyfikują je, a rezultat przesyłają na standardowe wyjście systemu (zazwyczaj monitor konsoli). W systemie operacyjnym PC DOS istnieją trzy rodzaje filtrów, będących rozkazami zewnętrznymi: FIND, MORE i SORT. Zostały one opisane w rozdziale pt. 'ROZKAZY SYSTEMU OPERACYJNEGO PC DOS'.

4. PRZYGOTOWANIE DYSKU STAŁEGO DO PRACY W SYSTEMIE OPERACYJNYM PC DOS.

4.1. Wprowadzenie.

Jeśli mikrokomputer IBM PC/XT lub AT wyposażono w stały dysk twardy typu WINCHESTER, to przed podjęciem jego eksploatacji należy go do niej odpowiednio przygotować. Wskazane jest aby czynność tę wykonała osoba doświadczona, która miała przedtem okazję poznać system operacyjny PC DOS.

W rozdziale tym przedstawiono sposób odwoływania się do poszczególnych stacji dyskowych i dyskietkowych, a także opisano szczegółowe algorytmy postępowania w następujących przypadkach:

1. Zastępowania starej wersji systemu operacyjnego PC DOS na dysku twardym jego nową wersją 3.20 (rozkaz REPLACE).
2. Zastępowania starych wersji dodatkowych programów systemowych PC DOS ich nowymi wersjami (rozkaz REPLACE).
3. Tworzenia, zmiany itd. partycji DOSa na dysku twardym (rozkaz FDISK).
4. Zastosowania rozkazu SELECT dla instalacji systemu operacyjnego PC DOS na dysku twardym.
5. Wykonywania restartu systemu operacyjnego z dysku twardego.

System operacyjny PC DOS pozwala na odwoływanie się do stacji dyskowych lub dyskietkowych poprzez wyspecyfikowanie odpowiednich liter. Słowo 'odwołanie' należy rozumieć jako sposób wywołania odpowiedniego programu obsługi danego urządzenia (ang. device driver). W systemie operacyjnym PC DOS przyjęto, że poprzez litery A i B można odwoływać się odpowiednio: do pierwszej i drugiej stacji dysków miękkich, natomiast litery C i D służą do oznaczania pierwszego i drugiego dysku twardego typu WINCHESTER, wszystkie zainstalowane wewnątrz obudowy mikrokomputera. Istnieją wprowadzić rozkazy systemu operacyjnego PC DOS, pozwalające w czasie pracy zmienić to przyporządkowanie (np. rozkaz ASSIGN), ale nie

obejmują one czynności opisanych w tym rozdziale. Należy pamiętać, że litery A, B i C zawsze służą do odwoływania się do odpowiednio: pierwszej i drugiej stacji dysków miękkich oraz pierwszego dysku twardego, nawet wtedy, gdy w konfiguracji sprzętowej brak jest któregoś z wymienionych urządzeń.

4.2. Algorytm zastępowania na dysku twardym starej wersji systemu operacyjnego PC DOS wersja 3.20.

Stara wersję systemu operacyjnego PC DOS można zastąpić nową wersją 3.20, korzystając z rozkazu REPLACE (patrz opis w rozdziale pt. 'ROZKAZY SYSTEMU OPERACYJNEGO PC DOS') według następującego algorytmu:

1. Założenie dyskietki z systemem operacyjnym PC DOS w wersji 3.20 na pierwszą stację dysków miękkich A: i wykonanie restartu systemu operacyjnego poprzez jednoczesne wciśnięcie kombinacji klawiszy 'Ctrl/Alt/Del'.
2. Po odpowiednich komunikatach wpisanie aktualnej daty i czasu, a następnie wywołanie rozkazu SYS o następującej postaci:

SYS C:

Spowoduje on przepisanie zbiorów systemowych z dyskietki na urządzeniu A: na dysk twardy C:.

3. Zastąpienie istniejących starych wersji zbiorów systemu operacyjnego PC DOS ich nowymi wersjami przy pomocy następującego rozkazu REPLACE:

REPLACE A:*.* C:\S\R

4. Dodanie nowych zbiorów systemowych wersji 3.20 do zbiorów zapisanych już na dysku twardym następującym rozkazem REPLACE:

REPLACE A:*.* C:\[ścieżka]\A

gdzie opcjonalny parametr 'ścieżka' określa ścieżkę od katalogu podstawowego do katalogu podrzędnego, zawierającego zbiory systemowe (patrz opis struktury drzewa katalogowego).

U W A G A :

Powyższy algorytm można wykonać tylko wtedy, gdy na dysku twardym zapisano uprzednio inny system operacyjny PC DOS. Jeśli jest to 'czysty' dysk, to należy postępować według algorytmów zawartych w podrozdziale pt. 'Podział dysku twardego na partycje DOSa. Tworzenie, zmiana, usuwanie i wyświetlanie informacji o partycjach'.

4.3. Algorytm zastępowania starych wersji dodatkowych programów systemowych ich nowymi wersjami.

Algorytm ten może być kontynuacją algorytmu opisanego w poprzednim podrozdziale.

1. Usunięcie z pierwszej stacji dysków miękkich A: dyskietki systemowej.
2. Wykonanie restartu systemu operacyjnego PC DOS wersja 3.20 z dysku twardego C:.
3. Po załadowaniu systemu operacyjnego do pamięci mikrokomputera, założenie na pierwszą stację dysków miękkich A: dyskietki z dodatkowymi zbiorami systemowymi.
4. Zmiana katalogu roboczego na dysku C: na katalog, w którym mają zostać zapisane dodatkowe zbiory systemowe.
5. Zainicjowanie wykonania następującego rozkazu REPLACE:

REPLACE A:*. * C:\S\R

6. Aby dopisać nowe, dodatkowe zbiory systemowe wersji 3.20 należy zainicjować następujący rozkaz REPLACE:

REPLACE A:*. * C:\[ścieżka]/A

gdzie opcjonalny parametr 'ścieżka' wyznacza ścieżkę od katalogu podstawowego do katalogu podrzędnego, w którym mają zostać zapisane zadane zbiory.

7. Po zakończonym kopiowaniu usunięcie dyskietki z dodatkowymi zbiorami systemowymi z pierwszej stacji dysków miękkich A:.

Po wykonaniu czynności opisanych w tym i poprzednim podrozdziale na dysku twardym C: znajduje się pełny system operacyjny PC DOS w wersji 3.20.

4.4. Podział dysku twardego na partycje. Tworzenie, zmiana, usuwanie i wyświetlanie informacji o partycjach.

Dysk twardy można podzielić na jedną, dwie, trzy lub cztery części zwane partycjami. W każdej partycji można zapisać tylko jeden system operacyjny. Do tworzenia, zmiany, usuwania i wyświetlania informacji o partycjach oraz do wybierania następnego dysku twardego do podziału służy specjalny rozkaz systemu operacyjnego PC DOS o nazwie FDISK. Rozkaz ten można zainicjować jak każdy 'zwykły' rozkaz systemowy, poprzez założenie dyskietki systemowej na pierwszą stację dysków miękkich A:, wykonanie restartu systemu operacyjnego, a następnie wywołanie rozkazu FDISK następującą komendą:

A>FDISK

Po wciśnięciu klawisza 'Enter' na ekranie monitora pojawiają się następujące komunikaty:

IBM Personal Computer
Fixed Disk Setup Program Version 3.20
(C)Copyright IBM Corp. 1983, 1985

FDISK Options

Current Fixed Disk Drive: 1

Choose one of the following:

1. Create DOS Partition
2. Change Active Partition
3. Delete Dos Partition
4. Display Partition Data
5. Select Next Fixed Disk Drive

Enter choice: []

Press ESC to return to DOS

gdzie:

1. Create DOS Partition - oznacza utworzenie partycji DOSa o określonej wielkości.
2. Change Active Partition - oznacza zmianę aktywnej partycji w celu użycia innego systemu operacyjnego.
3. Delete DOS Partition - oznacza usunięcie partycji DOSa.
4. Display Partition Data - oznacza wyświetlenie informacji o partycjach na dysku twardym.
5. Select Next Fixed Disk Drive - oznacza wybranie następnego dysku twardego do podziału.

Wybranie opcji polega na wciśnięciu klawisza z jej numerem a następnie klawisza 'Enter'. Powrót do systemu operacyjnego PC DOS następuje po wciśnięciu klawisza 'Esc'.

Opcje 1, 2, 3 i 4 dotyczą mikrokomputera IBM PC z pojedynczym, stałym dyskiem twardym. Opcja 5 dotyczy mikrokomputera z co najmniej dwoma dyskami twardymi.

4.4.1. Opcja 1 - tworzenie partycji DOSa.

1. Jeśli system operacyjny PC DOS ma być jedynym systemem na dysku twardym, to cały dysk twardy powinien zostać przeznaczony dla tego systemu.
2. Jeśli system operacyjny PC DOS ma być jednym z co najwyżej czterech systemów operacyjnych na dysku twardym, to należy utworzyć na nim tyle partycji, ile jest systemów.

Po dokonaniu wyboru opcji 1 rozkazu FDISK na ekranie monitora zostaje wyświetlony następujący komunikat:

Create DOS Partition

Current Fixed Disk Drive: 1

Do you wish to use the entire fixed disk
for DOS (Y/N).....? [Y]

Wciśnięcie klawisza z literą 'Y' oznacza, że cały dysk twardy ma zostać przeznaczony dla pojedynczego systemu operacyjnego PC DOS, natomiast wciśnięcie klawisza z literą 'N' oznacza, że dysk twardy zostanie podzielony na kilka partycji, każda dla innego systemu operacyjnego.

38. SYSTEMY OPERACYJNE: PC DOS 3.10, 3.20 oraz MS DOS 3.10.

Algorytm tworzenia pojedynczej partycji na dysku twardym dla pojedynczego systemu operacyjnego PC DOS:

Jeśli użytkownik zdecyduje się przeznaczyć cały dysk twardy dla pojedynczego systemu operacyjnego PC DOS, to wciska klawisz z literą 'Y'. Na ekranie monitora pojawia się następujący komunikat:

System will now restart
Insert DOS diskette in drive A:
Press any key when ready . . .

Należy wówczas:

1. Założyć na pierwszą stację dysków miękkich A: dyskietkę z systemem operacyjnym i wcisnąć klawisz dowolnej litery, który zainicjuje restart systemu operacyjnego z urządzenia A:.
2. Po odpowiednich komunikatach generowanych przez system należy wyspecyfikować aktualną datę i czas. Utworzenie partycji DOSa zostaje zasygnalizowane zgłoszeniem się systemu operacyjnego na ekranie monitora.

U W A G A :

Przed tworzeniem partycji należy zaformatować dysk.

3. Dalszy algorytm postępowania opisano w podrozdziale pt. "Zastosowanie rozkazu SELECT dla instalacji systemu operacyjnego PC DOS na dysku twardym".

Algorytm tworzenia kilku partycji na dysku twardym:

Jeżeli użytkownik zamierza zapisać na dysku twardym kilka systemów operacyjnych, to musi wykonać podział tego dysku na kilka partycji, każda z przeznaczeniem na inny system. Po wciśnięciu klawisza z literą 'N' na ekranie monitora pojawia się następujący komunikat:

Total fixed disk space is xxxx cylinders
Maximum available space is xxxx
cylinders at xxxx.

przy czym informacja 'Total fixed disk space' określa ile cylindrów znajduje się na całym dysku twardym, a 'Maximum available space' określa ile cylindrów na dysku twardym zostało przeznaczonych na partycje DOSa. Następnie na ekranie monitora zostaje wyświetlone kolejne pytanie dotyczące rozmiarów partycji przeznaczonej dla systemu operacyjnego PC DOS:

Enter partition size.....: [xxxx]

- jeśli użytkownik zamierza przeznaczyć całą dostępną przestrzeń na dysku twardym na partycje DOSa, to wciska tylko klawisz 'Enter',
- jeśli użytkownik nie zamierza przeznaczać całej dostępnej przestrzeni na dysku twardym na partycje DOSa, to określa ile cylindrów ma ona zajmować, a następnie wciska klawisz 'Enter'.

Kolejne pytanie dotyczy początkowego numeru cylindra na dysku twardym, od którego będzie rozpoczynała się partycja DOSa:

Enter starting cylinder number...: [xxxx]

Początkowy numer cylindra na dysku twardym zależy od określonych wcześniej rozmiarów partycji DOSa.

Po wykonaniu jednego z powyższych algorytmów na dysku twardym została utworzona partycja DOSa, ale przed jej wykorzystaniem należy:

- zmienić aktywną partycję (opcja 2 rozkazu FDISK),
- zainstalować system operacyjny PC DOS w tej partycji (patrz podrozdział pt. 'Zastosowanie rozkazu SELECT dla

instalacji systemu operacyjnego PC DOS na dysku twardym

Jeśli początek danej partycji wypada w miejscu, gdzie nośnik magnetyczny uległ uszkodzeniu, to rozkaz FDISK automatycznie przesunie ten początek w inne miejsce. Operacja ta zmniejsza jednak zadeklarowany rozmiar partycji.

4.4.2. Opcja 2 - zmiana aktywnej partycji.

Dysk twardy może być podzielony na kilka partycji, z których tylko jedna pozostaje aktywna. Oznacza to, że w przypadku zapisania na dysku twardym kilku systemów operacyjnych kontrolę nad systemem przejmuje ten, którego partycja jest aktywna (tylko ten system restartuje się). Wybranie opcji 2 rozkazu FDISK spowoduje wyświetlenie na ekranie monitora szeregu komunikatów. Przykładowy taki komunikat przedstawia poniższy rysunek:

```
Change Active Partition

Current Fixed Disk Drive: 1

Partition  Status  Type  Start  End  Size
   1         A    DOS   000   149   150
   2         N non-DOS  150   304   155
   3         N    xxx   xxx   xxx   xxx

Total disk space is 305 cylinders

Enter the number of the partition you
want to make active.....: [ ]
```

gdzie:

Partition - oznacza numer partycji,

Status - oznacza stan partycji, A-partycja aktywna,
N-partycja nieaktywna,

Type - oznacza typ partycji: partycja DOSa lub nie,

Start/End - oznacza numery cylindrów: początku i końca
partycji,

Size - oznacza rozmiar partycji w cylindrach.

Postępowanie użytkownika ogranicza się tutaj do określenia numeru partycji, którą zamierza uaktywnić i wciśnięciu klawisza 'Enter'.

Dalsze postępowanie należy realizować zgodnie z algorytmem przedstawionym w podrozdziale pt. 'Zastosowanie rozkazu SELECT dla instalacji systemu operacyjnego PC DOS na dysku twardym'.

4.4.3. Opcja 3 - usunięcie aktualnej partycji DOSa.

Opcja 3 rozkazu FDISK pozwala na usunięcie z dysku twardego partycji DOSa. Należy jednak pamiętać że wykonanie opcji 3 niszczy bezpowrotnie wszystkie dane zapisane w tej partycji, a także usuwa jej granice. Po usunięciu partycji DOSa z dysku twardego, system może zostać powtórnie załadowany do pamięci operacyjnej z dyskietki systemowej założonej na pierwszą stację dyskietkową A:.

Wybranie opcji 3 rozkazu FDISK spowoduje wyświetlenie na ekranie monitora szeregu komunikatów. Przykładowy taki komunikat przedstawiono na poniższym rysunku:

Delete DOS Partition

Current Fixed Disk Drive: 1

Partition	Status	Type	Start	End	Size
1	N	dos	000	304	305

Total fixed disk space is 305 cylinders

Warning! Data in the DOS partition
will be lost. Do you wish to
continue.....? [N]

Postępowanie użytkownika ogranicza się tutaj:

- do wciśnięcia klawisza 'Enter', jeśli nie zamierza usunąć aktualnej partycji DOSa (jest to odpowiedź domyślna, więc nie jest konieczne wciskanie klawisza z literą 'N'),
- do wciśnięcia klawisza 'Y', a następnie 'Enter', jeśli zamierza usunąć aktualną partycję DOSa. Zapisany w tej partycji system operacyjny zostanie wtedy skasowany.

Po usunięciu aktualnej partycji DOSa inny system operacyjny może zostać ściągnięty do pamięci z dyskietki systemowej, założonej na pierwszej stacji dysków miękkich A: lub poprzez uaktywnienie partycji z innym systemem operacyjnym, zapisanym na dysku twardym.

4.4.4. Opcja 4 - wyświetlanie informacji o istniejących na dysku twardym partycjach.

Opcja 4 rozkazu FDISK pozwala na wyświetlenie informacji dotyczących istniejących na dysku twardym partycjach. Informacje takie są także wyświetlane po wybraniu opcji 2 i 3.

Wybranie opcji 4 rozkazu FDISK spowoduje wyświetlenie na ekranie monitora szeregu komunikatów. Przykładowy taki komunikat pokazano na poniższym rysunku:

Display Partition Information

Current Fixed Disk Drive: 1

Partition	Status	Type	Start	End	Size
1	A	DOS	000	199	200
2	N	non-DOS	200	304	105

Total fixed disk space is 305 cylinders

gdzie:

Partition - oznacza numer partycji,

Status - oznacza stan partycji, A-partycja aktywna,
N-partycja nieaktywna,

Type - oznacza typ partycji: partycja DOSa lub nie,

Start/End - oznacza numery cylindrów: początku i końca
partycji,

Size - oznacza rozmiar partycji w cylindrach.

4.4.5. Opcja 5 - wybranie następnego dysku stałego do podziału na partycje.

Jeśli mikrokomputer IBM PC posiada w konfiguracji sprzętowej
więcej niż jeden dysk twardy, to można użyć opcji 5 rozkazu FDISK
do wykonania podziału następnego dysku twardego na partycje.
Wciśnięcie klawisza z cyfrą 5 spowoduje wyświetlenie takiego sa-
mego menu postępowania jak w przypadku pierwszego dysku twardego,
tym razem w odniesieniu do następnego dysku.

4.5. Zastosowanie rozkazu SELECT dla instalacji systemu operacyjnego PC DOS na dysku twardym.

Po utworzeniu partycji DOSa na dysku twardym można, przy pomo-
cy rozkazu SELECT, zaformatować tę partycję, a następnie zainsta-
lować w niej system operacyjny PC DOS. Rozkaz SELECT (patrz opis
w rozdziale pt. 'ROZKAZY SYSTEMU OPERACYJNEGO PC DOS'), oprócz
instalacji samego systemu, pozwala jednocześnie na zapisanie
informacji o tym, jakiego układu klawiatury oraz jakiego formatu
daty i czasu będzie można używać w zainstalowanym systemie.
Szczegółowe dane na ten temat można znaleźć w opisie instrukcji
COUNTRY oraz rozkazów: KEYBxx i SELECT.

U W A G A :

1. Podczas wykonywania rozkazu SELECT partycja DOSa na dysku twardym jest formatowana, toteż należy się liczyć ze skasowaniem zapisanej w niej poprzedniej wersji systemu i wszystkich innych zbiorów.
2. Jeśli partycja DOSa na dysku twardym jest mniejsza niż 10MB i została zaformatowana przy pomocy rozkazów PC DOS 3.20, to nie można zainstalować w niej wersji systemu operacyjnego o numerze mniejszym niż 3.00.

Algorytm instalowania systemu operacyjnego PC DOS na dysku twardym w partycji DOSa przy pomocy rozkazu SELECT:

1. Założenie dyskietki z systemem operacyjnym PC DOS wersja 3.20, na pierwszą stację dysków miękkich A:, wykonanie restartu systemu i zainicjowanie wykonania następującego rozkazu SELECT:

A>[ścieżka]SELECT C:[ścieżka] xxx yy

gdzie:

[ścieżka], przed słowem kluczowym SELECT, określa ścieżkę do zbioru, gdzie znajduje się program obsługi rozkazu SELECT,

C: określa dysk twardy, w którym zostanie zainstalowany system operacyjny,

[ścieżka] określa katalog, w którym zostaną zapisane zbiory systemowe. Pominięcie tego parametru spowoduje domyślne przyjęcie katalogu podstawowego,

xxx określa kod kraju, którego format daty i czasu będzie formatem obowiązującym w zainstalowanym systemie operacyjnym,

yy określa układ klawiatury kraju, który będzie obowiązywał w zainstalowanym systemie operacyjnym.

Bliższe dane dotyczące wymienionych parametrów można znaleźć w opisie rozkazu SELECT, w rozdziale pt. 'ROZKAZY SYSTEMU OPERACYJNEGO PC DOS'.

2. Zainicjowanie wykonania rozkazu SELECT spowoduje wyświetlenie następującego komunikatu ostrzegawczego:

```
SELECT is used to install DOS the first  
time. SELECT erases everything on the  
specified target and then installs DOS.  
Do you want to continue (Y/N)?
```

Aby kontynuować instalację należy wcisnąć klawisz z literą 'Y'.

3. Jeśli program obsługi klawiatury o kodzie określonym parametrem 'yy' znajduje się na dyskietce systemowej, to skocz do punktu 4. W przeciwnym wypadku system żąda potwierdzenia jego obecności na innym nośniku, wyświetlając następujący komunikat:

```
Is KEYBxx.COM on another  
diskette (Y/N)?
```

Należy wcisnąć klawisz z literą 'Y'.

4. System operacyjny wyświetla następujący komunikat:

```
WARNING, ALL DATA ON NON-REMOVABLE DISK  
DRIVE C: WILL BE LOST!  
Proceed with Format (Y/N)?
```

Żądając kolejnego potwierdzenia zamiaru skasowania całej informacji wpisanej do partycji DOSa na dysku twardym. Należy wcisnąć klawisz z literą 'Y'. Rozpoczyna się formatowanie dysku twardego.

5. Kolejne etapy formatowania są 'ilustrowane' odpowiednimi komunikatami określającymi numer głowicy i cylindra. Po zakończeniu formatowania system wyświetla następujący komunikat, żądając wprowadzenia (lub nie) etykiety dysku twardego (patrz opis rozkazu LABEL i FORMAT):

Format complete
System transferred

Volume label (11 characters, ENTER for none)?

Po wprowadzeniu etykiety pojawiają się kolejne komunikaty informacyjne:

xxxxxx bytes total disk space
xxxxxx bytes used by system
xxxxxx bytes in bad sectors
xxxxxx bytes available on disk

6. Jeśli na dyskietce systemowej brak jest programu obsługi układu klawiatury o kodzie określonym parametrem 'yy' to skocz do punktu 7. Jeśli program ten znajduje się na dyskietce systemowej to rozpoczyna się kopiowanie kolejnych zbiorów systemowych na dysk C:. Nazwy kopiowanych zbiorów system wyświetla na ekranie monitora. Po zakończonym kopiowaniu zgłasza się system operacyjny:

A>

Skocz do punktu 10.

7. Jeśli program obsługi układu klawiatury o kodzie określonym parametrem 'yy' znajduje się na innej dyskietce niż dyskietka systemowa, to system następującym komunikatem:

Insert KEYByy.COM diskette in drive A:
Strike any key when ready

żąda założenia tej dyskietki do pierwszej stacji dyskietek A:, w miejsce dyskietki systemowej i potwierdzenie tego wciśnięciem klawisza dowolnej litery.

8. Po skopiowaniu programu obsługi układu klawiatury KEYByy.COM kolejnym komunikatem o następującej treści:

Insert DOS diskette in Drive A:
Strike any key when ready

system operacyjny żąda powtórnej zamiany dyskietek pierwszej stacji dysków miękkich A: na dyskietkę z systemem operacyjnym.

9. Po wymianie dyskietek i wciśnięciu klawisza dowolnej litery rozpoczyna się kopiowanie zbiorów systemowych z dyskietki na urządzeniu A: na dysk twardy C:. Nazwy kolejno przepisywanych zbiorów są wyświetlane na ekranie

monitora. Po zakończonym kopiowaniu zgłasza się system operacyjny PC DOS:

A>

10. Aby przepisać dodatkowe zbiory systemowe, których brak jest na dyskietce systemowej, należy wymienić dyskietkę w pierwszej stacji dysków miękkich A: na dyskietkę z tymi zbiorami i zainicjować wykonanie następującego rozkazu COPY:

COPY A:*. * C:

Nazwy kolejno kopiowanych zbiorów są wyświetlane na ekranie monitora. Po zakończonym kopiowaniu system operacyjny został zainstalowany w partycji DOSa na dysku twardym C:, skąd można wykonać jego restart.

4.6. Restartowanie systemu operacyjnego PC DOS z dysku twardego.

Zainstalowanie systemu operacyjnego PC DOS na dysku twardym C:, opisane w poprzednich podrozdziałach, oznacza, że po włączeniu zasilania mikrokomputera lub po wciśnięciu kombinacji klawiszy 'Ctrl/Alt/Del' system będzie restartowany (ładowany do pamięci operacyjnej) właśnie z tego dysku twardego. Trzeba pamiętać, że przed wykonaniem restartu należy otworzyć zatrzask blokujący dyskietkę w pierwszej stacji dysków miękkich A:.

5. KONFIGUROWANIE SYSTEMU OPERACYJNEGO.

Zbiór o nazwie CONFIG.SYS, znajdujący się w katalogu podstawowym, na urządzeniu, z którego wykonuje się restart systemu operacyjnego, jest specjalnym zbiorem systemowym kształtującym konfigurację systemu. Podczas restartu system operacyjny PC DOS odczytuje poszczególne polecenia zawarte w tym zbiorze, interpretuje je, a następnie wykonuje. Zmiana zbioru CONFIG.SYS podczas sesji pracy pod systemem operacyjnym PC DOS ukształtuje nową konfigurację systemu dopiero po jego ponownym restarcie. Jeśli podczas restartu system operacyjny nie odnajdzie zbioru CONFIG.SYS, to podstawia za odpowiednie parametry instrukcji konfiguracyjnych system ich wartości domyślne. Zbiór CONFIG.SYS umożliwia:

1. Ustawianie funkcji kombinacji klawiszy 'Ctrl/Break' (instrukcja BREAK).
2. Określanie liczby buforów dyskowych w systemie (instrukcja BUFFERS).
3. Określanie formatu daty i czasu, w którym dane te będą wpisywane (instrukcja COUNTRY).
4. Ładowanie do pamięci operacyjnej programów instacyjnych poszczególnych urządzeń (tzw. device drivers - instrukcja DEVICE).
5. Określanie liczby zbiorów, które mogą być kontrolowane przez tzw. bloki kontrolne zbioru (ang. file control blocks - instrukcja FCBS).
6. Określanie liczby zbiorów, które jednocześnie mogą być otwarte (instrukcja FILES).
7. Określanie maksymalnej liczby urządzeń dyskowych, dostępnych dla systemu operacyjnego PC DOS (instrukcja LASTDRIVE).
8. Wyprecyzowanie zbioru, który zastąpi standardowy interpreter rozkazów COMMAND.COM (instrukcja SHELL).

Zbiór CONFIG.SYS można utworzyć przy pomocy dowolnego edytora, bądź bezpośrednio z klawiatury konsoli, wykorzystując rozkaz COPY:

COPY CON nazwa.BAT 'Enter'

. . . ciąg kolejnych instrukcji konfiguracyjnych
. . . system operacyjny, każdy zakończony
. . . wciśnięciem klawisza 'Enter'

klawisz 'F6', a następnie 'Enter'

W razie pomyłki można przerwać wpisywanie, bez zachowania dotychczas wpisanego fragmentu, wciskając kombinację klawiszy 'Ctrl/Break'.

5.1. Instrukcja BREAK.

Format:

BREAK [ON|OFF]

Opis:

Domyślnie (BREAK=OFF) system operacyjny PC DOS sprawdza, czy użytkownik przy pomocy wciskania kombinacji klawiszy 'Ctrl/Break' chce przerwać wykonywanie jakiegoś polecenia (rozkażu, instrukcji) tylko w trakcie wykonywania funkcji związanych ze standardowym wejściem (zwykle klawiatura), standardowym wyjściem (zwykle ekran monitora), standardowym urządzeniem drukującym itp. Instrukcja:

BREAK ON

powoduje, że system operacyjny sprawdza, czy na klawiaturze wciśnięto kombinację klawiszy 'Ctrl/Break' podczas wykonywania każdej funkcji. Można w ten sposób przerwać wykonywanie programów wykonujących czynności nie związane ze standardowymi urządzeniami wejścia/wyjścia. Instrukcja BREAK bez podania parametrów wyświetla aktualny stan kontroli klawiszy 'Ctrl/Break' ustawiony ostatnią instrukcją tego typu.

5.2. Instrukcja BUFFERS.

Format:

BUFFERS=xx

Opis:

Instrukcja BUFFERS określa ilość buforów systemowych, które będą zarezerwowane podczas restartu systemu operacyjnego. Parametr xx może być liczbą dziesiętną od 1 do 99. Domyślnie system operacyjny określa liczbę buforów na 2 - dla mikrokomputera IBM PC/XT, lub na 3 - dla mikrokomputera IBM PC/AT.

Bufor dyskowy jest ciągłym obszarem pamięci operacyjnej wykorzystywanej przez system operacyjny PC DOS podczas czytania/pisania danych z/na dysk. Np. jeśli wykonywany jest odczyt 128-bajтового rekordu ze zbioru zapisanego na dysku, wówczas system operacyjny przepisuje cały sektor z tym rekordem do bufora systemowego, odszukuje w nim żądany rekord i dopiero wtedy przesyła go do właściwego miejsca w pamięci operacyjnej. Im więcej miejsca zarezerwowano w pamięci operacyjnej na bufory, tym więcej danych można w nich przechowywać. Podczas wykonywania operacji czytania/pisania danych z/na dysk system operacyjny w pierwszym kroku zawsze przegląda zawartość buforów, poszukując w nich żądanych danych. To pozorne komplikowanie operacji czytania i pisania danych zwiększa prędkość wykonywania poszczególnych poleceń (dostęp do pamięci operacyjnej jest znacznie szybszy niż do pamięci masowej). Optymalna liczba buforów może zmieniać się w zależności od charakteru programu. Ponieważ każdy dodatkowy bufor wymaga 512 bajtów pamięci operacyjnej RAM przeznaczonych na część rezydentną systemu operacyjnego, to jednocześnie o tę wartość maleje rozmiar pamięci RAM przeznaczony dla użytkownika. A zatem dla określenia optymalnej liczby buforów należy wziąć pod uwagę:

- charakter programów, które najczęściej będą wykonywane,
- wielkość pamięci operacyjnej wbudowanej w komputer,
- czy w konfiguracji sprzętowej występują dyski twarde (potrzeba wówczas minimum trzy bufory).

Przykład:

BUFFERS=5 - instrukcja ta, znajdująca się w zbiorze CONFIG.SYS, ustawia 5 buforów systemowych.

5.3. Instrukcja COUNTRY.

Format:

COUNTRY=xxx

Opis:

Instrukcja COUNTRY określa format daty i czasu dla wybranego kraju. Parametr xxx symbolizuje kod wybranego kraju, zgodnie z poniższą tabelą:

KRAJ	KOD KRAJU	KRAJ	KOD KRAJU
Australia	061	kraje Środko-	786
Belgia	032	wego Wschodu	
Kanada	002	Holandia	031
(franc.)		Norwegia	047
Dania	045	Portugalia	351
Finlandia	358	Hiszpania	034
Francja	033	Szwecja	046
RFN	049	Szwajcaria	041
Włochy	039	W. Brytania	044
Izrael	972	USA	001

Przykład:

COUNTRY=001 - po wykonaniu tej instrukcji (po wykonaniu restartu systemu operacyjnego PC DOS, zawierającego zbiór CONFIG.SYS z tą instrukcją) podczas wprowadzania daty i czasu należy użyć formatu obowiązującego w USA, tj. mm-dd-yy i hh:mm:ss, gdzie: mm-miesiąc, dd-dzień, yy-rok, hh-godzina, mm-minuta, ss-sekunda.

5.4. Instrukcja DEVICE.

Format:

DEVICE=[d:]ścieżka[nazwa[.rozszerzenie]]

Opis:

Instrukcja DEVICE umożliwia określenie urządzenia, katalogu i nazwy zbioru instalacyjnego, zawierającego program obsługi określonego urządzenia (ang. device driver). Podczas restartu systemu operacyjnego zbiór ten ładowany jest do pamięci operacyjnej, umożliwiając wykorzystywanie tego urządzenia. System operacyjny PC DOS, dostarczany przez producenta, zawiera programy instalacyjne, obsługujące standardowe urządzenia: wejściowe (klawiaturę), wyjściowe (monitor), drukujące, oraz zegar, stacje dysków elastycznych i stacje dysków twardych. Nie jest zatem konieczne podawanie dla nich dodatkowych instrukcji DEVICE. Instrukcja ta jest jednak wymagana przy instalowaniu dostarczanych wraz z systemem operacyjnym programów:

- programu ANSI.SYS, obsługującego rozszerzony ekran i klawiatury (extended screen and keyboard),
- programu DRIVER.SYS obsługującego zewnętrzne i wewnętrzne urządzenia dyskowe,
- programu VDISK.SYS obsługującego zasymulowane w pamięci operacyjnej urządzenia dyskowe.

Obsługa wewnętrznych i zewnętrznych urządzeń dyskowych
(tylko w wersji PD DOS 3.20):

U W A G A :

Poniżej przyjęto, że określenie 'stacja dyskowa' oznacza stację dysku twardego (typu WINCHESTER) lub stację dysku miękkiego. Pod pojęciem 'wewnętrzna stacja dyskowa' jest rozumiana stacja dyskowa zamontowana na stałe wewnątrz obudowy mikrokomputera. Określenie 'zewnętrzna stacja dyskowa' oznacza stację dyskową, która znajduje się poza obudową mikrokomputera, ale jest podłączona do niego.

Podczas pracy z mikrokomputerem IBM PC/XT lub AT zachodzi czasem konieczność wykorzystania pojedynczej stacji dyskowej do transferu danych pomiędzy kilkoma różnymi nośnikami (dyskietkami), lub posługiwania się zewnętrzną stacją dyskową.

W systemie operacyjnym PC DOS przyjęto, że:

- poprzez litere A można odwoływać się do pierwszej wewnętrznej stacji dysków miękkich,
- poprzez literę B można odwoływać się do drugiej wewnętrznej stacji dysków miękkich,
- poprzez litere C i dalsze można odwoływać się do stacji dysków twardych, wbudowanych w mikrokomputer, zewnętrznych stacji dyskowych, a w szczególnym przypadku - do pierwszej i drugiej wewnętrznej stacji dysków miękkich.

W czasie restartu system operacyjny PC DOS sprawdza istnienie wewnętrznych stacji dyskowych, a następnie przegląda zbiór CONFIG.SYS pod kątem uzyskania informacji o:

- istnieniu zewnętrznych stacji dyskowych,
- żądaniu użytkownika zapewnienia możliwości odwoływania się do tej samej stacji dyskowej poprzez różne oznaczenia literowe.

Informacje te użytkownik przekazuje do systemu operacyjnego poprzez wstawienie do zbioru CONFIG.SYS instrukcji DEVICE, instalującej z odpowiednimi parametrami specjalny zbiór (program) o nazwie DRIVER.SYS.

Format:

DEVICE=DRIVER.SYS/D:ddd[/T:ttt][/S:ss][/H:hh][/C][/N][/F:f]

Opis:

Zbiór DRIVER.SYS znajdujący się na dysku systemowym pozwala na wykorzystywanie zewnętrznych i wewnętrznych urządzeń dyskowych poprzez odwoływanie się do nich przez odpowiednie oznaczenia literowe. Wymienione w linii komendy parametry mają następujące znaczenie:

1. Parametr /D:ddd określa liczbę odpowiadającą zainstalowanemu urządzeniu dyskowemu. Liczba ta może zmieniać się od 0 do 255, w ten sposób, że:

/D:0 odpowiada pierwszej wewnętrznej stacji dys-

kietkowej, do której w specyfikacjach rozkazów systemu operacyjnego PC DOS można odwoływać się poprzez literę A,

/D:1 odpowiada drugiej stacji dyskietkowej,

/D:2 odpowiada trzeciej (zawsze zewnętrznej) stacji dyskietkowej,

/D:128 odpowiada pierwszej stacji dysków twardych,

/D:129 odpowiada drugiej stacji dysków twardych.

2. Parametr /T:ttt określa liczbę ścieżek na pojedynczej stronie dysku lub dyskietki. Wartość ttt może zmieniać się od 1 do 999, domyślnie wynosi 80.
3. Parametr /S:ss określa liczbę sektorów na ścieżkę. Wartość ss może zmieniać się od 1 do 99, domyślnie wynosi 9.
4. Parametr /H:hh określa maksymalną liczbę głowic 'obsługujących' daną stację dyskową. Wartość hh może zmieniać się od 1 do 99, domyślnie wynosi 2.
5. Parametr /C oznacza, że może być konieczna zmiana stacji dyskietkowej (zmiana w sensie podłączenia jednej stacji zamiast drugiej),
6. Parametr /N oznacza, że w urządzeniu dyskowym nie można wymieniać nośnika (np. dysk twardy typu WINCHESTER).
7. Parametr /F:f określa typ urządzenia zgodnie z poniższą tabelką:

Wartość f	Typ urządzenia
0	160/180kB
0	320/360kB
1	1.2MB
2	720kB

Przykład 1:

W zestawie sprzętowym mikrokomputera IBM PC/XT znajdują się następujące urządzenia: wewnętrzna stacja dysków miękkich, wewnętrzna stacja dysku twardego i zewnętrzna stacja dysków miękkich umożliwiające posługiwanie się dyskietkami 720kB. Dla zapewnienia możliwości posługiwanie się tą stacją zewnętrzną należy w zbiorze CONFIG.SYS umieścić następującą instrukcję:

```
DEVICE=DRIVER.SYS/D:2
```

a następnie wykonać restart systemu operacyjnego.

Przykład 2:

W zestawie sprzętowym mikrokomputera IBM PC/AT znajdują się następujące urządzenia: 2 wewnętrzne stacje dysków miękkich i wewnętrzna stacja dysku twardego. Pierwsza stacja dysków miękkich umożliwia posługiwanie się dyskietkami 1.2MB, druga - dyskietkami 320/360kB. Instrukcja:

```
DEVICE=DRIVER.SYS/D:0/T:80/S:15/H:2/C/F:1
```

zapewnia możliwość odwoływania się, w rozkazach systemu operacyjnego PC DOS, poprzez litery A i D do pierwszej stacji dysków miękkich. Oznacza to, że po restarcie systemu operacyjnego ze zbiorem CONFIG.SYS, zawierającym te instrukcje np. rozkaz:

```
A>COPY file1 D:
```

skopiuje zbiór o nazwie file1, zapisany w katalogu roboczym na dyskietce znajdującej się w stacji dyskietkowej A:, na inną dyskietkę, założoną następnie na tą samą stację. System operacyjny PC DOS odpowiednimi komunikatami będzie sterował wymianą dyskietek w tej stacji.

Przykład 3:

Dla zapewnienia możliwości posługiwanie się pojedynczą zewnętrzną stacją dysków miękkich dla transferu danych pomiędzy różnymi nośnikami (zestaw sprzętowy jak w przykładzie 1) należy w zbiorze CONFIG.SYS umieścić dwie instrukcje DEVICE o następującej postaci:

```
DEVIVE=DRIVER.SYS/D:2
```

```
DEVIVE=DRIVER.SYS/D:2
```

Po restarcie systemu operacyjnego ze zbiorem CONFIG.SYS zawierającym tę instrukcję, do zewnętrznej stacji dysków miękkich można odwoływać się poprzez litery D i E, umożliwiając, tak jak w przykładzie 2, wymianę nośnika.

Symulacja w pamięci operacyjnej urządzeń dyskowych:

```
DEVICE=[d:][\ścieżka]\VDISK.SYS[ comment][ bbb][ comment][ sss]
[ comment][ ddd][ /EC:m]
```

Opis:

Zbiór VDISK.SYS, znajdujący się na dysku systemowym, pozwala na symulowanie urządzenia dyskowego w pamięci operacyjnej RAM. Urządzenie to zostaje zainstalowane instrukcją DEVICE w zbiorze CONFIG.SYS jako tzw. dysk wirtualny (ang. virtual disk). Dysk wirtualny pozwala na szybsze, w porównaniu do "zwykłego" dysku, wykonywanie wszelkich operacji związanych z odczytem i zapisem danych. Możliwe jest zainstalowanie więcej niż jednego dysku wirtualnego, używając dla każdego z nich osobnej instrukcji DEVICE. Każdy z nich zostanie oznaczony kolejną literą, różną jednak od liter oznaczających istniejące w konfiguracji sprzętowej "zwykłe" urządzenia dyskowe. W przypadku mikrokomputerów IBM PC/AT w których dodatkowo zamontowano rozszerzoną pamięć RAM (ponad 1 MB), dyski wirtualne mogą być instalowane w tej pamięci. Dla każdego dysku wirtualnego można określić rozmiar pamięci zajmowany przez ten dysk, wielkości sektorów na nim, liczbę katalogów możliwych do zainstalowania, a także etykietę identyfikującą. Zawartość dysków wirtualnych będzie zawsze tracona po wyłączeniu zasilania komputera lub po wykonaniu restartu systemu operacyjnego. Dysk wirtualnego nie trzeba formatować.

U W A G A :

Każdy kolejny dysk wirtualny, oprócz zajęcia określonego obszaru pamięci operacyjnej przez "sam dysk", powoduje dodatkowe zwiększenie o 800 bajtów pamięci RAM zajmowanej przez część rezydentną systemu operacyjnego PC DOS. Pamięć ta zostaje zajęta przez program obsługi tego dysku (ang. disk driver).

Wymienione w linii komendy parametry mają następujące znaczenie:

1. Parametr "bbb" określa wielkość dysku wirtualnego w kilobajtach. Wartość bbb może zmieniać się od 1kB do 64kB (najwyżej pełnego rozmiaru dostępnej dla użytkownika pamięci operacyjnej). Domyślnie system operacyjny przyjmuje bbb=64kB.

- jeśli w chwili instalowania dysku wirtualnego

dostępna dla użytkownika pamięć operacyjna jest mniejsza niż 64kB, to program VDISK wyświetli komunikat o błędzie i nie zainstaluje dysku,

- jeśli rozmiar pamięci zadeklarowanej dla dysku wirtualnego jest mniejszy niż 1kB lub większy niż dostępna dla użytkownika w chwili instalacji pamięć RAM, to program VDISK zainstaluje dysk wirtualny zajmujący 64kB tej pamięci,
- jeśli po zainstalowaniu dysku wirtualnego o określonym przez użytkownika rozmiarze zostałoby do wykorzystania mniej niż 64kB pamięci operacyjnej, to program VDISK automatycznie zmniejszy rozmiar tego dysku do wartości, przy której pozostanie wolnych co najmniej 64kB pamięci RAM.

Wartość parametru bbb obejmuje nie tylko rozmiar pamięci operacyjnej przeznaczonej na zbiory użytkownika, lecz także na zbiory i buforów związane z ich obsługą (katalogi, FAT czyli tablicę określającą położenie poszczególnych zbiorów na dysku etc).

2. Parametr 'sss' określa rozmiar sektora dysku wirtualnego w bajtach. Parametr 'sss' może przyjmować wartości: 128, 256 lub 512, domyślnie system operacyjny przyjmuje sss=128, także wtedy, gdy podano wartość różną od wymienionych powyżej. Jeśli użytkownik zamierza użyć dysku wirtualnego do przechowywania zbiorów o niewielkich rozmiarach, to wskazane jest zadeklarowanie mniejszych sektorów dla uniknięcia zbędnych strat w wykorzystaniu pamięci operacyjnej (w sektorze można zapisać informacje pochodzącą tylko z jednego zbioru, a więc jeśli zbiór zajmuje np. 150 bajtów, a zadeklarowano rozmiar sektora równy 512 bajtów, to niewykorzystanych pozostanie 362 bajtów).
3. Parametr 'ddd' określa liczbę zbiorów, które można zapamiętać na dysku wirtualnym. Parametr 'ddd' może przyjmować wartości od 2 do 512, domyślnie system operacyjny przyjmuje ddd=64. Program VDISK nie trzyma się 'sztywno' zadeklarowanej wartości parametru 'ddd':
 - ilość możliwych do zapisywania zbiorów może zostać zwiększona do wartości wynikającej z zadeklarowanego rozmiaru sektora, np. jeśli użytkownik zadeklaruje ddd=10, a rozmiar sektora wynosi 128, to program VDISK przyjmie, że na dysku wirtualnym można zapisać 12 zbiorów (12 zbiorów, każdy po 32 bajty zajmuje 384 bajty, czyli dokładnie 3 sektory),

- jeśli dysk wirtualny, określony powyżej, jest zbyt mały dla przechowywania FAT (File Allocation Table - tablica określająca położenie poszczególnych zbiorów na dysku), katalogu z zapisanymi zbiorami i dwóch dodatkowych sektorów, to liczba zbiorów możliwych do zapisania na dysku wirtualnym będzie automatycznie zmniejszana o jeden tyle razy, aż możliwy będzie zapis (w chwili, gdy liczba ta osiągnie wartość 1, a nadal brak jest miejsca na zapis, to program VDISK wyświetli komunikat o błędzie i nie zainstaluje dysku wirtualnego).
4. Parametr 'comment' (koniecznie ze spacją poprzedzającą komentarz) jest komentarzem zawierającym od 20 do 126 dowolnych znaków ASCII za wyjątkiem znaku '/'.
5. /E - klucz ten sygnalizuje, że dysk wirtualny ma zostać zainstalowany w obszarze rozszerzonej pamięci RAM (ponad 1Mb). Klucza tego można używać tylko w mikrokomputerze IBM PC/AT z rozszerzoną pamięcią. W rozszerzonej pamięci można instalować kilka dysków wirtualnych, przy pomocy kilku instrukcji DEVICE=VDISK.SYS.
- m jest maksymalną liczbą sektorów (o rozmiarze sss), które naraz przesyła program VDISK.SYS. Wartość ta może wynosić 1, 2, 3, 4, 5, 6, 7 lub 8, domyślnie wynosi 8.

U W A G A :

W przypadku zamontowania dysku wirtualnego w rozszerzonej pamięci operacyjnej, podczas transferu danych zostaje zawieszona obsługa przerw. Jeśli częstotliwość tych przerw jest dostatecznie duża, wówczas mogą być one 'zgubione' i pozostawione całkowicie bez obsługi.

Przykład:

```
DEVICE=VDISK 320 512  
DEVICE=VDISK 160 512 128
```

Powyższe instrukcje ze zbioru CONFIG.SYS tworzą dwa dyski wirtualne: jeden o pojemności 320 kB, zawierający 512 sektorów i drugi: o pojemności 160 kB zawierający 512 sektorów, na którym można zapisać 128 zbiorów.

5.5. Instrukcja FCBS.

Format:

FCBS=m,n

Opis:

Instrukcja FCBS pozwala określić liczbę tzw. bloków kontrolnych zbioru (ang. FCBs - File Control Blocks), które mogą być jednocześnie otwarte przez system operacyjny PC DOS. Blok kontrolny zbioru FCBs jest to fragment pamięci operacyjnej, rezerwowany podczas wywoływania każdego zbioru, zawierający wszystkie niezbędne informacje dotyczące tego zbioru: gdzie został umieszczony, jaki program z niego korzysta etc.

Wymienione w linii komendy parametry mają następujące znaczenie:

1. Parametr 'm' określa całkowitą liczbę zbiorów otwartych w danej chwili a kontrolowanych przez FCBs. Parametr ten można zmieniać od 1 do 255, wartość domyślna wynosi 4.
2. Parametr 'n' określa liczbę zbiorów kontrolowanych przez FCBs, których nie może automatycznie zamknąć system operacyjny PC DOS, o ile dany program próbuje korzystać więcej niż z m zbiorów równocześnie kontrolowanych przez FCBs. Pierwszych n zbiorów kontrolowanych przez FCBs jest wówczas zabezpieczonych przed zamknięciem. Parametr 'n' można zmieniać od 0 do 255, wartość domyślna wynosi 0.

Niektóre programy aplikacyjne wykorzystują bloki kontrolne zbiorów do tworzenia, otwierania, kasowania, czytania i zapisywania zbiorów. System operacyjny PC DOS czuwa nad ostatnio używanymi blokami kontrolnymi zbiorów. Jeśli dany program próbuje otworzyć więcej niż m zbiorów kontrolowanych przez FCBs to reakcja systemu

operacyjnego zależy od tego, czy możliwe jest wykorzystywanie pojedynczego zbioru przez różne programy (ang. file sharing):

- jeśli jest możliwe, to system operacyjny PC DOS zamyka ostatnio używany FCBs i otwiera nowy zbiór,
- jeśli nie jest możliwe, to liczba zbiorów, które mogą być jednocześnie otwarte w tym samym czasie, nie jest ograniczona. Instrukcja FCBs ma więc zastosowanie tylko wtedy, gdy pojedynczy zbiór może być wykorzystywany przez różne programy.

Wartość parametru m musi być większa lub równa wartości parametru n . Jeśli $m=n$ i dany program próbuje otworzyć więcej niż m zbiorów to system operacyjny nie może zamknąć żadnego zbioru aby otworzyć nowy. W przypadku wykorzystywania przez dany program do kontroli tego samego zbioru co najmniej dwóch FCBs-ów, system operacyjny traktuje, że użyty został tylko jeden FCBs. Należy ponadto pamiętać, że umieszczenie instrukcji FCBs w zbiorze CONFIG.SYS zwiększa zajętość pamięci operacyjnej, zajmowanej przez rezydentną część systemu PC DOS.

Przykład:

FCBS=8,4 - instrukcja FCBs znajdująca się w zbiorze CONFIG.SYS określa liczbę zbiorów, możliwych do otwarcia przez FCBs, na 8. Jeśli program otwiera więcej niż 8 zbiorów, wtedy pierwsze cztery otwarte zbiory nie będą automatycznie zamykane przez system PC DOS.

5.6. Instrukcja FILES.

Format:

FILES=xx

Opis:

Instrukcja FILES ustawia maksymalną liczbę zbiorów, które można jednocześnie używać. Ilość zbiorów określa parametr xx. Może on przyjmować wartości od 8 do 255, domyślnie przyjmowane jest $xx=8$.

Dostęp do zbioru, związany z jego czytaniem, zapisem zamknięciem można uzyskać poprzez poinformowanie systemu operacyjnego PC DOS który zbiór będzie wykorzystywany. System operacyjny utworzy wówczas w polu pamięci operacyjnej, zarezerwowanej dla jego potrzeb podczas restartu, blok kontrolny tego zbioru. Wielkość tego pola (a więc w konsekwencji maksymalna liczba zbiorów, które można jednocześnie wykorzystywać) zależy od wartości podanej instrukcją FILES. Wartość domyślna FILES=8 jest wystarczająca dla większości zastosowań, jakkolwiek niekiedy zachodzi konieczność jej zmiany. Każdy dodatkowy zbiór ponad 8, wyspecyfikowany w instrukcji FILES, powoduje zarezerwowanie dodatkowych 48 bajtów pamięci operacyjnej, co jednocześnie zmniejsza o tę wartość ilość pamięci pamięci RAM dostępnej dla użytkownika.

Przykład:

FILES=16 - umożliwia pracę z szesnastoma zbiorami jednocześnie.

5.7. Instrukcja LASTDRIVE.

Format:

LASTDRIVE=x

Opis:

Instrukcja LASTDRIVE określa maksymalną liczbę urządzeń dyskowych dostępnych dla systemu operacyjnego. Parametr 'x' może być litera od A do Z, wartość domyślna wynosi E. Użyta litera będzie nazwą ostatniego urządzenia, które zaakceptuje system operacyjny PC DOS. Minimalna liczba urządzeń, którą można określić instrukcją LASTDRIVE, jest równa aktualnej liczbie urządzeń dyskowych, podłączonych w systemie. Instrukcja zostanie zignorowana przez system operacyjny, jeśli liczba 'x' jest mniejsza od liczby przyłączonych urządzeń w aktualnej konfiguracji sprzętowej.

Przykład:

LASTDRIVE=P - instrukcja ta, wymieniona w zbiorze CONFIG.SYS, określi liczbę dostępnych urządzeń dyskowych dla systemu operacyjnego na 16 (P jest szesnastą literą alfabetu).

5.8. Instrukcja SHELL.

Format:

SHELL=[d:][ścieżka]nazwa[.rozszerzenie][/E:xxxxx][/P]

Opis:

Instrukcja SHELL pozwala na określenie nazwy i miejsca zapisu zbioru będącego podstawowym interpreterem rozkazów, który zostanie zainicjowany podczas restartu systemu operacyjnego. Programista, który zamierza zainicjować swój własny interpreter rozkazów, musi zapewnić obsługę przerwań: 22H, 23H i 24H dla czytania i wykonywania rozkazów. Ponieważ programy obsługi rozkazów wewnętrznych systemu operacyjnego PC DOS oraz interpreter umożliwiający przetwarzanie wsadowe (ang. batch processor) znajdują się w zbiorze COMMAND.COM, to po zainicjowaniu drugiego interpretera, nie będą dostępne aż do chwili ich skopiowania do tego interpretera.

Poszczególne klucze występujące w linii komendy instrukcji SHELL mają następujące znaczenie:

klucz /E:xxxxx (tylko w wersji PC DOS 3.20) pozwala na zarezerwowanie w pamięci operacyjnej miejsca niezbędnego do przechowywania tablicy z opisem otoczenia programisty (patrz rozkaz SET). Parametr 'xxxxx' określa dziesiętną liczbę całkowitą z przedziału [160, 32768]. Przed zarezerwowaniem pamięci, system operacyjny zawsze zaokrągla tę liczbę do najbliższej większej liczby całkowitej podzielnej przez 16. Jeśli parametr 'xxxxx' określa liczbę większą niż 32768 lub mniejszą niż 160, wtedy system operacyjny przyjmuje odpowiednio: 32768 lub 160.

klucz /P (tylko w wersji PC DOS 3.20) spowoduje, że interpreter COMMAND.COM wykona, zaraz po załadowaniu do pamięci operacyjnej, rozkazy zawarte w zbiorze rozkazów przetwarzania wsadowego o nazwie AUTOEXEC.BAT.

6. ROZKAZY SYSTEMU OPERACYJNEGO PC DOS.

W systemie operacyjnym PC DOS istnieją dwa typy rozkazów:

1. Rozkazy wewnętrzne, działające natychmiast po wywołaniu, gdyż stanowią one integralną część systemu operacyjnego. Rozkazy te są ładowane do pamięci operacyjnej komputera podczas restartu systemu operacyjnego i nie wymagają powtórznego ściągnięcia po zainicjowaniu ich wykonania.
2. Rozkazy zewnętrzne, znajdujące się w pamięci masowej jako zbiory, a zatem przed ich wykonaniem muszą być ściągnięte do pamięci operacyjnej. Zbiory te mogą posiadać jedno z trzech następujących rozszerzeń: .BAT, .COM lub .EXE. Przy wywoływaniu rozkazów zewnętrznych nie trzeba podawać ich rozszerzenia.

Wykonując rozkaz w systemie operacyjnym PC DOS należy pamiętać o tym, że:

- system operacyjny oczekuje na podanie rozkazu zgłaszając się literowym symbolem stacji roboczej ze znakiem '>' np. A> (zgłoszenie można zmienić dokonując zmian w zbiorze AUTOEXEC.BAT lub wykorzystując rozkaz PROMPT),
- nazwę rozkazu zewnętrznego mogą poprzedzać dane dotyczące miejsca jego zapisania (tzn. urządzenie i katalog),
- nazwa rozkazu poprzedza zwykle jeden lub więcej parametrów,
- nazwy rozkazów mogą być pisane zarówno małymi jak i dużymi literami,
- nazwy rozkazów i parametry muszą być oddzielone od siebie odpowiednimi separatorami. Separatorami mogą być spacja, przecinek, średnik, znak równości lub znak tabulacji poziomej.
- nazwa zbioru, występująca w linii rozkazu, składa się z następujących czterech części: nazwy stacji, ścieżki, 'właściwej' nazwy zbioru i rozszerzenia tego zbioru. Nazwa stacji jest pojedynczą literą z dwukropkiem, ścieżka to ciąg (co najwyżej 63 znaków) nazw kolejnych katalogów oddzielonych od siebie znakiem '\' (tzw. backslash), nazwa to ciąg od 1 do 8 znaków, a rozszerzenie - od 0 do 3 znaków.

- nie wolno oddzielać separatorami żadnej z trzech części tworzącej identyfikator zbioru, tj.

(urządzenie:)(nazwa)(.rozszerzenie)

zawsze (o ile jest wymagana ich obecność) muszą być pisane 'razem',

- aby zatrzymać wykonywanie rozkazu należy wcisnąć kombinację klawiszy 'Ctrl/Break'. Znaczenie wciśnięcia tych klawiszy jest rozpoznawane przez system wtedy, gdy znajduje się on w trybie pracy umożliwiającym czytanie z klawiatury, lub wyświetlanie na ekranie terminala. A zatem wciśnięcie tych klawiszy nie zawsze jest równoważne z natychmiastowym zatrzymaniem wykonywania rozkazu (patrz opis rozkazu BREAK),
- inicjacja wykonywania rozkazu następuje zawsze po wciśnięciu klawisza 'Enter',
- po zainicjowaniu rozkazu zewnętrznego (programu) system operacyjny, aby go automatycznie ściągnąć do pamięci operacyjnej, poszukuje go w aktualnym katalogu roboczym, na urządzeniu roboczym. Jeśli rozkaz nie 'znajdzie się' w tym katalogu, poszukiwania są kontynuowane w katalogach zdefiniowanych ostatnim rozkazem PATH (patrz opis),
- wystąpienie w linii rozkazu znaku (znaków): '<', '>', lub '!' spowoduje wtórna definicję standardowych urządzeń wejściowych lub wyjściowych, albo tzw. przetwarzanie potokowe (ang. piping),
- w opisach linii komend poszczególnych rozkazów mogą wystąpić następujące oznaczenia:

! oznacza wybór pomiędzy wartościami dwóch argumentów,

... oznacza, że może wystąpić kilka argumentów o identycznym formacie,

[] oznacza, że parametry ujęte w te nawiasy są opcjonalne.

6.1. Rozkaz ASSIGN (Assign Drive).

Format:

[d:][ścieżka]ASSIGN [x[=]y[...]]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz ASSIGN pozwala zmienić sposób odwoływania się do urządzeń dyskowych, wszelkie odwołania do urządzenia 'x' są wykonywane na urządzeniu 'y'.

Poszczególne parametry występujące w linii komendy rozkazu ASSIGN mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym ASSIGN, określa miejsce, w którym zapisano program obsługi rozkazu ASSIGN,

x określa urządzenie, do którego, przed wykonaniem rozkazu ASSIGN, są kierowane odwołania,

y określa urządzenie, do którego, po wykonaniu rozkazu ASSIGN, będą kierowane odwołania.

Urządzenia: x i y muszą fizycznie istnieć w danej konfiguracji sprzętowej. Urządzeniami tymi mogą być: stacje dyskietkowe, dyski twarde lub tzw. dyski wirtualne (patrz opis instrukcji DEVICE w rozdziale poświęconym konfiguracji systemu operacyjnego). W linii komendy rozkazu ASSIGN po nazwach urządzeń x i y nie trzeba pisać dwukropka. Rozkaz ASSIGN bez parametrów przywraca normalny stan odwoływania się do poszczególnych urządzeń.

Zmiana odwołań do poszczególnych urządzeń nie może mieć miejsca w przypadku rozkazów: LABEL, JOIN, SUBST i PRINT. Ponadto rozkazy: FORMAT, DISKCOMP i DISKCOPY ignorują zmiany wykonane rozkazem ASSIGN.

Przykłady:

Rozkaz ASSIGN o następującej postaci:

```
A>ASSIGN A=C B=C
```

spowoduje przypisanie wszystkich działań na stacjach dyskiety-
nych A: i B: do dysku stałego C:. Program obsługi rozkazu ASSIGN
znajduje się w katalogu roboczym, na urządzeniu roboczym.

Rozkaz ASSIGN o następującej postaci:

```
C>B:\USER ASSIGN
```

przywraca normalny stan odwoływania się do poszczególnych urzą-
dzeń. Program obsługi rozkazu ASSIGN znajduje się w katalogu
\\USER na urządzeniu B:.

6.2. Rozkaz ATTRIB (Attribute).

Format:

```
[d:][ścieżka]ATTRIB [+R|-R][+A|-A][d1:][ścieżka1]nazwa  
[.rozszerzenie]
```

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz ATTRIB jest używany do ustawiania lub usuwania atrybutu
RO (Read Only - tylko czytanie), a także do ustawiania lub ga-
szenia tzw. bitu archiwalnego danego zbioru. Ustawiony dla danego
zbioru atrybut RO oznacza, że zbiór ten może być tylko odczytywa-
ny, natomiast nie można go usunąć z dysku bez wcześniejszej zmia-
ny atrybutu na przeciwny. Bit archiwalny jest rodzajem wskaźnika,
używany do oznaczania danego zbioru, najczęściej w przypadku
wykonywania kopii zapasowych (rozkaz BACKUP /M i XCOPY /M).

Poszczególne parametry występujące w linii komendy rozkazu ATTRIB mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym ATTRIB, określa miejsce, w którym znajduje się program obsługi rozkazu ATTRIB,

[d1:][ścieżka][nazwa][.rozszerzenie] określa zbiór lub zbiory (wolno stosować znaki specjalne '?' i '*'), których atrybuty będą zmieniane,

+R oznacza ustawienie atrybutu RO (Read Only) dla wyspecyfikowanego zbioru (zbiorów),

-R oznacza usunięcie atrybutu RO (Read Only) dla wyspecyfikowanego zbioru (zbiorów),

+A (tylko w wersji PC DOS 3.20) oznacza ustawienie bitu archiwalnego dla wyspecyfikowanego zbioru (zbiorów),

-A (tylko w wersji PC DOS 3.20) oznacza zgaszenie bitu archiwalnego dla wyspecyfikowanego zbioru (zbiorów).

Rozkaz ATTRIB bez argumentów +R, -R, +A, -A wyświetla dla danego zbioru (zbiorów) aktualne stany wszystkich atrybutów.

Przykłady:

Rozkaz ATTRIB o następującej postaci:

```
C>B:ATTRIB +R ABC.DAT
```

ustawia atrybut RO dla zbioru ABC.DAT zapisanego w katalogu roboczym na urządzeniu roboczym. Program obsługi rozkazu ATTRIB znajduje się w katalogu roboczym na urządzeniu B:.

Rozkaz ATTRIB o następującej postaci:

```
B>A:\USER ATTRIB *.*
```

wyświetla atrybuty wszystkich zbiorów zapisanych w katalogu roboczym, na urządzeniu roboczym B:. Program obsługi rozkazu ATTRIB znajduje się w katalogu \USER na urządzeniu A:.

Rozkaz ATTRIB o następującej postaci:

B>ATTRIB -A C:ZBR.TXT

gasi bit archiwalny zbioru ZBR.TXT zapisanego w katalogu roboczym na urządzeniu C:. Program obsługi rozkazu ATTRIB znajduje się w katalogu roboczym, na urządzeniu roboczym B:.

Rozkaz ATTRIB o następującej postaci:

C>ATTRIB TEST.DOC

wyświetla aktualny atrybut zbioru TEST.DOC, zapisanego w katalogu roboczym na urządzeniu roboczym.

6.3. Rozkaz BACKUP.

Format:

[d:][ścieżka]BACKUP d1:[ścieżka][nazwa[.rozszerzenie]]
d2:[/S][/M][/A][/D:mm-dd-yy]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz BACKUP pozwala na wykonywanie kopii zapasowej zbioru lub grupy zbiorów (całych katalogów lub zawartości całych dysków) z jednego dysku na drugi. Kopiowanie można wykonywać:

- z dysku twardego na dyskietki,
- z dyskietki na dyskietki,
- z dyskietki na dysk twardy,
- z dysku twardego na dysk twardy.

Poszczególne parametry występujące w linii komendy rozkazu BACKUP mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym BACKUP, określa miejsce w którym znajduje się program obsługi rozkazu BACKUP,

d1:[ścieżka][nazwa][.rozszerzenie] określa zbiór lub grupę zbiorów, dla których będą wykonywane kopie zapasowe. Brak nazwy zbioru spowoduje skopiowanie całego katalogu, brak nazwy katalogu spowoduje skopiowanie zawartości całego dysku lub dyskietki znajdującej się na urządzeniu d1;

d2: określa urządzenie, na którym będą zapisywane kopie zapasowe. Nazwa tego urządzenia musi być różna od nazwy urządzenia źródłowego d1:.

Sposób wykonywania rozkazu BACKUP mogą wymuszać dodatkowo następujące klucze:

klucz /S wymusza wykonanie również kopii zapasowych katalogów podrzędnych w stosunku do wyspecyfikowanego katalogu,

klucz /M powoduje wykonanie kopii zapasowych tych zbiorów, które modyfikowano od czasu wykonania rozkazem BACKUP ich kopii zapasowych,

klucz /A wymusza dopisanie kopii zapasowych wyspecyfikowanych zbiorów do zbiorów już istniejących na dysku lub dyskietce wyjściowej (rozkaz BACKUP bez klucza /A kasuje 'starą' zawartość całego dysku lub dyskietki wyjściowej),

klucz /D:mm-dd-yy (lub: /D:dd-mm-yy lub: /D:yy-mm-dd, zależnie od formatu daty ustawionego instrukcją COUNTRY lub rozkazem SELECT) wymusza utworzenie kopii zapasowych tylko tych zbiorów, które zapisano lub zmodyfikowano od dnia określonego wyspecyfikowaną datą.

W systemie operacyjnym MS DOS wersja 3.10 występują dodatkowo następujące klucze:

klucz /L wymusza utworzenie w katalogu podstawowym zbioru o nazwie BACKUP.LOG, który zawiera raport o wykonanych, tym rozkazem BACKUP, kopiach zapasowych zbiorów. Raport ten zawiera nazwy, rozszerzenia, miejsca zapisu i czas utworzenia kopii zapasowych poszczególnych zbiorów.

klucz /T:hh:mm:ss wymusza odtwarzanie tylko tych zbiorów, które zostały utworzone lub zmodyfikowane od wyspecyfikowanego czasu,

klucz /P wymusza zapisywanie tylu zbiorów, ile jest to tylko możliwe, na dyskietkę założoną w wyspecyfikowane

urządzenie. Jednocześnie rozkaz BACKUP z tym kluczem pozwala na tworzenie na dyskietce wyjściowej katalogów podrzędnych, o ile pozwoli to zwiększyć upakowanie zbiorów.

Uwagi dotyczące posługiwania się rozkazem BACKUP:

1. W specyfikacjach nazw i rozszerzeń zbiorów można używać znaków specjalnych '?' i '*'. Np. rozkaz:

BACKUP C:*. * A:/S

spowoduje wykonanie kopii zapasowych całego katalogu podstawowego (ang. root directory) wraz z katalogami niższych poziomów, z urządzenia C: na urządzenie A:.

2. Rozkaz BACKUP nie jest odpowiednikiem rozkazu COPY. Rozkaz COPY powoduje dokładne przepisanie całego zbioru źródłowego, natomiast rozkaz BACKUP przepisuje tylko te zbiory, których nie zapisano innym rozkazem BACKUP. W przypadku konieczności wykonania kopii zapasowych zbiorów zapisanych rozkazem BACKUP, muszą one zostać uprzednio odtworzone rozkazem RESTORE (patrz opis rozkazu RESTORE).
3. Jeśli w systemie pojedyncze zbiory mogą być równocześnie wykorzystywane przez różne zadania (ang. file sharing - patrz opis rozkazu SHARE), to kopie zapasowe tych zbiorów można wykonywać tylko wtedy, gdy możliwy jest dostęp do nich (żadne zadanie nie korzysta z nich). W przeciwnym razie wykonywanie rozkazu BACKUP zostanie przerwane, a na ekranie monitora pojawi się komunikat:

PATHNAME\FILENAME.EXT

Not able to backup at this time.

gdzie PATHNAME\FILENAME.EXT określa o jakim zbiorze jest mowa.

4. Każda nowa dyskietka, przed wykonaniem rozkazu BACKUP, musi zostać zaformatowana.
5. Jeśli na dyskietce, na której mają zostać wykonane kopie zapasowe, zapisano wcześniej inne zbiory, to podczas wykonywania rozkazu BACKUP zostaną one skasowane, chyba że użyto klucza /A.
6. Rozkaz BACKUP zapewnia wykorzystanie pełnej pojemności dyskietki wyjściowej, nawet wtedy, gdy jakiś zbiór nie mieści się w całości. W takim przypadku ta część zbioru,

która nie zmieściła się, zostaje zapisana na innej dyskietce (system operacyjny PC DOS żąda wymiany dyskietki). Należy być świadomym zachowania kolejności zakładania poszczególnych dyskietek podczas odtwarzania podzielonego zbioru przy pomocy rozkazu RESTORE. Ułatwieniem jest to, że rozkaz BACKUP wyświetla nazwę każdego zbioru, dla którego wykonuje kopie zapasowe.

7. Jeśli urządzeniem wyjściowym jest dysk twardy, to kopie zapasowe będą tworzone w katalogu o nazwie \BACKUP. W przypadku dyskietek kopie będą zapisywane w katalogu podstawowym.
8. Jeśli którakolwiek z dyskietek: źródłowa lub wyjściowa ma założoną protekcję zapisu, to system operacyjny PC DOS zarządza jej wymiany (w przypadku dyskietki źródłowej rozkaz BACKUP zaznacza na niej dla których zbiorów były wykonywane kopie zapasowe).
9. Rozkazu BACKUP nie powinno się wykonywać w połączeniu z rozkazem JOIN, gdyż może to prowadzić do powstania na dysku wyjściowym takiej struktury drzewa katalogowego, do której nie można później zastosować rozkazu RESTORE.

W zależności od wyniku wykonania rozkazu BACKUP, system operacyjny PC DOS ustawia parametr ERRORLEVEL instrukcji warunkowej IF (przetwarzanie wsadowe) na następujące wartości:

- 0 - poprawne zakończenie wykonywania rozkazu BACKUP,
- 1 - nie odnaleziono zbioru dla wykonania jego kopii zapasowej,
- 2 - zablokowany dostęp do pewnych zbiorów, dla których miały zostać wykonane kopie zapasowe (zbiory te jednocześnie wykorzystywały inne zadania).
- 3 - wykonywanie rozkazu BACKUP zostało przerwane przez użytkownika poprzez wciśnięcie kombinacji klawiszy 'Ctrl/Break'.
- 4 - wykonywanie rozkazu BACKUP zostało przerwane przez jakiś błąd.

Przykłady:

Rozkaz BACKUP o następującej postaci:

BACKUP A:FILE.TXT C:

spowoduje wykonanie kopii zapasowej zbioru o nazwie FILE.TXT znajdującego się na urządzeniu A:. Kopia będzie wykonana na dysku twardym C:.

Rozkaz BACKUP o następującej postaci:

BACKUP A: C:/D:8-21-84

spowoduje wykonanie kopii zapasowych wszystkich zbiorów zapisanych na urządzeniu A:, które powstały, lub które zmodyfikowano w dniu 21 sierpnia 1984 roku i później. Kopia będzie wykonana na dysku twardym C:.

Rozkaz BACKUP o następującej postaci:

BACKUP A: B:/A

spowoduje dopisanie do kopii zapasowych, istniejących na dyskietce założonej na urządzenie B:, kopii zapasowych wszystkich zbiorów z dyskietki założonej na urządzenie A:.

6.4. Rozkaz BREAK (Control Break).

Format:

BREAK [ON|OFF]

Typ:

Rozkaz wewnętrzny.

Opis:

Domyślnie (BREAK=OFF) system operacyjny PC DOS sprawdza, czy użytkownik przy pomocy wciskania kombinacji klawiszy 'Ctrl/Break' chce przerwać wykonywanie jakiegoś polecenia (rozkażu, instrukcji) tylko w trakcie wykonywania funkcji związanych ze standardowym wejściem (zwykle klawiatura), standardowym wyjściem (zwykle ekran monitora), standardowym urządzeniem drukującym itp. Rozkaz:

BREAK ON

powoduje, że system operacyjny sprawdza, czy na klawiaturze wciśnięto kombinację klawiszy 'Ctrl/Break' podczas wykonywania każdej funkcji. Można w ten sposób przerwać wykonywanie programów wykonujących czynności nie związane ze standardowymi urządzeniami wejścia/wyjścia. Rozkaz BREAK, bez podania parametrów, wyświetla aktualny stan kontroli klawiszy 'Ctrl/Break' ustawiony ostatnim rozkazem tego typu.

Rozkazu BREAK najczęściej używa się w zbiorze konfigurującym system operacyjny PC DOS o nazwie CONFIG.SYS (rozkaz ten omówiono także w rozdziale dotyczącym konfigurowania systemu operacyjnego).

6.5. Rozkaz CHDIR (Change Directory).

Format:

CHDIR [d:][ścieżka]

lub:

CD [d:][ścieżka]

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz CHDIR pozwala na zmianę katalogu roboczego na wyspecyfikowanym lub roboczym urządzeniu oraz umożliwia wyświetlenie na ekranie monitora aktualnej ścieżki-do-zbioru (katalogu).

Poszczególne parametry występujące w linii komendy rozkazu CHDIR mają następujące znaczenie:

[d:] określa urządzenie, na którym ma zostać zmieniony katalog roboczy. Jeśli pominięto parametr 'd:', to program obsługi rozkazu CHDIR domyślnie przyjmie urządzenie robocze.

[ścieżka] określa katalog (ścieżkę-do-zbioru), który po wykonaniu rozkazu CHDIR stanie się katalogiem roboczym na danym urządzeniu. Specyfikacja parametru 'ścieżka' nie może zawierać więcej niż 63 znaki. Jeśli pominięto

ten parametr, to program obsługi rozkazu CHDIR wyświetli aktualną ścieżkę-do-zbioru na urządzeniu ds. (lub urządzeniu roboczym, jeśli pominięto także parametr 'd:').

Uwagi dotyczące posługiwania się rozkazem CHDIR:

1. Aktualny katalog roboczy można zmieniać na każdy inny katalog istniejący na danym urządzeniu, poprzez wyspecyfikowanie w linii komendy rozkazu CHDIR pełnej ścieżki do tego zbioru, rozpoczynając od katalogu podstawowego. O tym, że ścieżka-do-zbioru jest określana z katalogu podstawowego użytkownik informuje program obsługi rozkazu CHDIR poprzez wpisanie jako pierwszego znaku po specyfikacji urządzenia (lub po słowie kluczowym CHDIR, jeśli pominięto parametr 'd:') znaku '\\'. Np. rozkaz CHDIR o następującej postaci:

A>CD B:\LEVEL1\LEVEL2

spowoduje, że bez względu na to który katalog był do tej pory katalogiem roboczym, tym katalogiem stanie się katalog LEVEL2 (zapisany w katalogu LEVEL1, a ten z kolei w katalogu podstawowym).

2. Jeśli katalogiem roboczym jest katalog LEVEL2, to rozkaz CHDIR o następującej postaci:

A>CD B:LEVEL3

spowoduje, że katalogiem roboczym stanie się katalog LEVEL3, zapisany w katalogu LEVEL2. W tym przypadku aktualna ścieżka-do-zbioru z katalogu podstawowego będzie miała następującą postać:

\\LEVEL1\LEVEL2\LEVEL3

Powyższy rozkaz CD jest w pewnym sensie równoważny rozkazowi:

* CD B:\LEVEL1\LEVEL2\LEVEL3

z tym, że ten ostatni umożliwia zmianę katalogu roboczego na LEVEL3 z dowolnego katalogu, a nie tylko z katalogu LEVEL2.

3. Podobnie jak w powyższych przykładach, można wykonywać zmiany katalogów roboczych z katalogów niższego poziomu na katalogi wyższego poziomu.
4. 'Prawdziwy' katalog roboczy na danym urządzeniu może

zostać zamaskowany przez zainicjowanie instrukcji ASSIGN, SUBST lub JOIN.

6.6. Rozkaz CHKDSK (Check Disk).

Format:

[d:][ścieżka]CHKDSK [d1:][ścieżka1][nazwa[.rozszerzenie]]
[/F][/V]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz CHKDSK pozwala na wykonanie analizy struktury katalogów, zbiorów oraz tzw. FAT (ang. File Allocation Table), czyli tablicy określającej m.in. położenie poszczególnych zbiorów na dysku (dyskietce), ustalając położenie pustych sektorów oraz miejsc, gdzie nośnik magnetyczny uległ uszkodzeniu. Po wykonaniu tych czynności, system operacyjny wyświetla stosowny raport, zawierający dane dotyczące stanu sprawdzanego dysku (dyskietki) oraz pamięci operacyjnej.

Poszczególne parametry występujące w linii komendy rozkazu CHKDSK mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym CHKDSK, określa miejsce, w którym znajduje się program obsługi rozkazu CHKDSK,

[d1:] określa urządzenie zawierające dysk (dyskietkę) poddawany analizie. Pominięcie parametru 'd1:' spowoduje domyślne przyjęcie urządzenia roboczego,

[ścieżka1][nazwa[.rozszerzenie]] określa katalog i nazwę zbioru. Podanie parametrów 'nazwa' i '.rozszerzenie' spowoduje wyświetlenie ilości nieciągłych obszarów zajmowanych przez dany zbiór,

klucz /F pozwala na wykonanie korekty błędów znalezionych w katalogu lub FAT, przy czym korekta zostanie wykonana na urządzeniu d1: lub urządzeniu roboczym. Jeśli w linii komendy rozkazu CHKDSK pominięto klucz /F, to

będą podejmowane wszystkie czynności, których wymaga przeprowadzenie korekty, ale bez jej wykonywania. W przypadku wyspecyfikowania katalogu podrzędnego, do którego nie można dotrzeć, system operacyjny wyświetli następujący komunikat:

```
Cannot CHKDSK to d1:ścieżka1,  
tree past this point not processed.
```

klucz /V spowoduje wyświetlenie nazw, rozszerzeń oraz katalogów tych wszystkich zbiorów, które na analizowanym dysku lub dyskietce zostały zapisane w sposób nieciągły.

Program obsługi rozkazu CHKDSK, natychmiast po wciśnięciu klawisza 'Enter', rozpoczyna analizę zapisu na dyskietce znajdującej się w wyspecyfikowanej lub roboczej stacji, a zatem nie ma żadnej możliwości wymiany tej dyskietki. Ma to szczególne znaczenie w systemach, które w swojej konfiguracji sprzętowej zawierają tylko jedną stację dyskietkową. Należy wtedy pamiętać, aby parametr 'd1:' nie określał urządzenia roboczego, gdyż spowoduje to wykonanie analizy zapisu na dyskietce systemowej.

Uwagi dotyczące posługiwania się rozkazem CHKDSK:

1. Przed rozpoczęciem wykonywania jakichkolwiek zmian na analizowanym dysku lub dyskietce, związanych z ewentualną korektą, program obsługi rozkazu CHKDSK żąda potwierdzenia 'Y' lub zaprzeczenia 'N', oba zakończone wciśnięciem klawisza 'Enter', zamiaru wykonania tych poprawek.
2. Jeśli rozkaz CHKDSK został wywołany z wyspecyfikowaną w linii komendy nazwą zbioru, to wyświetli liczbę nieciągłych obszarów zajmowanych przez ten zbiór. Sprawdzenie zbiorów, zapisanych w sposób nieciągły, powoduje wydłużenie czasu wykonywania analizy, gdyż zbiory te nie mogą być czytane z zachowaniem kolejności poszczególnych sektorów.
3. W linii komendy rozkazu CHKDSK można używać znaki specjalne '?' i '*' w specyfikacjach nazw i rozszerzeń zbiorów.
4. Jeśli program obsługi rozkazu CHKDSK wykryje na analizowanym nośniku uszkodzone obszary, to żąda od użytkownika zgody na podjęcie próby odzyskania tych zbiorów, które częściowo zapisano w uszkodzonych miejscach. Jeśli uzyska potwierdzenie, a rozkaz CHKDSK zainicjowano z kluczem /F, wtedy odzyskiwana informacja będzie zapisywana do zbioru o nazwie:

FILEnnnn.CHK

gdzie nnnn określa kolejną liczbę, rozpoczynającą się od wartości 0000. Zbiory FILEnnnn.CHK tworzone są w katalogu podstawowym na urządzeniu roboczym i, po zakończeniu wykonywania rozkazu CHKDSK, mogą być poddawane weryfikacji co do przydatności zawartej w nich informacji (na temat odzyskiwania zbiorów bliższe informacje można znaleźć w opisie rozkazu RECOVER).

5. Raport, generowany przez program obsługi rozkazu CHKDSK, można zapisywać w określonym zbiorze, bez wyświetlania jego zawartości na ekranie monitora, np.

A>CHKDSK B:>FILE.TXT

spowoduje utworzenie zbioru o nazwie FILE.TXT, zawierającego raport z analizy dyskietki założonej na stację B:.

Przykłady:

Przykładowy raport z wykonania analizy dysku może mieć następującą postać:

```
179712 bytes total disk space
 18944 bytes in 3 hidden files
   512 bytes in 1 directories
 26112 bytes in 4 user files
134144 bytes available on disk
```

```
196608 bytes total memory
170736 bytes free
```

Należy zwrócić uwagę, że raport obejmuje także 3 zbiory ukryte: IBMBIO.COM, IBMDOS.COM oraz zbiór z nazwą etykiety dysku. Ponadto pewne programy aplikacyjne także mogą tworzyć własne zbiory ukryte.

Rozkaz CHKDSK o następującej postaci:

A>CHKDSK A:

spowoduje wykonanie analizy dyskietki założonej na pierwszą stację dysków miękkich A: i wyświetlenie na ekranie monitora stosownego raportu.

Rozkaz CHKDSK o następującej postaci:

A>CHKDSK A:/F

spowoduje wykonanie analizy dyskietki założonej na pierwszą stację dysków miękkich A:, a następnie korekty znalezionych na niej błędów. Po zakończeniu korekty, na ekranie monitora zostanie wyświetlony stosowny raport.

Rozkaz CHKDSK o następującej postaci:

A>CHKDSK A:/V

spowoduje wykonanie analizy dyskietki założonej na pierwszą stację dysków miękkich A:, wyświetlenie stosownego raportu oraz podanie nazw, rozszerzeń i katalogów tych zbiorów, które zostały zapisane na tej dyskietce w sposób nieciągły.

Rozkaz CHKDSK o następującej postaci: .

A>CHKDSK C:*. *

spowoduje wykonanie analizy dysku twardego C:, wyświetlenie stosownego raportu oraz podanie nazw, rozszerzeń i katalogów tych zbiorów, które zostały zapisane w sposób nieciągły. Ta ostatnia informacja zostanie wymuszona nie kluczem /V, lecz specyfikacją *.*.

Przykładowy fragment raportu, dotyczący zbiorów zapisanych w sposób nieciągły, może mieć następująca postać:

```
C:\FILE1.TXT
  Contains 2 non-contiguous blocks
C:\FILE1.TXT
  Contains 4 non-contiguous blocks
C:\FILE1.TXT
  Contains 3 non-contiguous blocks
```

6.7. Rozkaz CLS (Clear Screen).

Format:

CLS

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz CLS czyści ekran monitora i ustawia kursor w jego lewym górnym rogu oraz przywraca obraz czarno-biały, o ile kolor tła i znaków nie został ustawiony przez funkcję 'rozszerzonego sterowania ekranem i klawiaturą' (ang. 'extended screen and keyboard control').

6.8. Rozkaz COMMAND (Secondary Command Processor).

Format:

COMMAND [d:][ścieżka][/P][C string][/E:xxxxx]

Opis:

Rozkaz COMMAND pozwala zainicjować drugi interpreter rozkazów (ang. secondary command processor).

Poszczególne parametry, występujące w linii komendy rozkazu COMMAND, mają następujące znaczenie:

[d:][ścieżka] określa urządzenie i katalogi, w których system operacyjny będzie poszukiwał drugiego interpretera rozkazów. Jeśli interpreter ten nie zostanie znaleziony w wyspecyfikowanych katalogach (parametr 'ścieżka'), to system operacyjny będzie prowadził poszukiwania w katalogach, których nazwy znajdują się w tablicy opisującej otoczenie programisty (ang. environment, patrz rozkaz SET),

klucz /P spowoduje, że nowy interpreter rozkazów pozostanie na stałe w pamięci operacyjnej, aż do chwili wykonania restartu systemu operacyjnego. Oznacza to, że bez wykonania restartu systemu operacyjnego nie będzie można odwoływać się do pierwszego interpretera rozkazów,

[/C string] pozwala na powrót do pierwszego interpretera rozkazów po wykonaniu rozkazu określonego parametrem 'string'. Np. zainicjowanie rozkazu COMMAND o następującej postaci:

A>COMMAND /C DIR B:

spowoduje, że drugi interpreter rozkazów zostanie załadowany do pamięci operacyjnej, następnie wykona rozkaz DIR B: i spowoduje powrót do pierwszego interpretera,

klucz /E:xxxxx (tylko w wersji PC DOS 3.20) pozwala na zarezerwowanie w pamięci operacyjnej miejsca niezbędnego do przechowywania tablicy z opisem otoczenia programisty (patrz rozkaz SET). Parametr 'xxxxx' określa dziesiętną liczbę całkowitą z przedziału [160, 32768]. Przed zarezerwowaniem pamięci system operacyjny zawsze zaokrągla tę liczbę do najbliższej, większej liczby całkowitej podzielnej przez 16. Jeśli parametr 'xxxxx' określa liczbę większą niż 32768 lub mniejszą niż 160, wtedy system operacyjny przyjmuje odpowiednio: 32768 lub 160. Zainicjowanie rozkazu COMMAND bez klucza /E z parametrem 'xxxxx' spowoduje, że kopia nowego interpretera rozkazów zostanie załadowana do pamięci operacyjnej i będzie mogła wykorzystywać tę tablicę z opisem otoczenia programisty, którą używał poprzedni interpreter rozkazów. Zmiany w opisie otoczenia programisty, wykonane rozkazem SET, przy zainicjowanym drugim interpreterze rozkazów dotyczą tylko tego interpretera i po powrocie do pierwszego interpretera są przez niego zignorowane (obowiązuje wówczas otoczenie programisty pierwszego interpretera).

Wyspecyfikowanie w linii komendy rozkazu COMMAND kluczy: /P i /C spowoduje zignorowanie znaczenia klucza /P.

Przykład:

Niech pierwszy interpreter rozkazów powoduje 'zwykle' zgłaszanie się systemu operacyjnego tj. kodem literowym urządzenia roboczego i znakiem '>'. Zainicjowanie drugiego interpretera rozkazów nic nie zmieni w sposobie zgłaszania się systemu. Zmiana w tym interpreterze sposobu zgłaszania się systemu operacyjnego, po powrocie do pierwszego interpretera, zostanie przez pierwszy interpreter zignorowana.

Jeśli drugi interpreter rozkazów został zainicjowany bez podania w linii komendy rozkazu COMMAND klucza /C z parametrem 'string', to powrót do pierwszego interpretera rozkazów nastąpi po wywołaniu specjalnego rozkazu EXIT.

6.9. Rozkaz COMP (Compare Files, nie występuje w wersji MS DOS 3.10).

Format:

[d:][ścieżka]COMP [d1:][ścieżka1][nazwa1[.rozszerzenie1]]
[d2:][ścieżka2][nazwa2[.rozszerzenie2]]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz COMP umożliwia porównywanie zawartości pierwszego zbioru (grupy zbiorów) z zawartością drugiego zbioru (grupy zbiorów), przy czym w odróżnieniu od rozkazu DISKCOMP, rozkaz COMP porównuje zbiory, a nie informacje zapisane na dwóch różnych dyskieciekach.

Poszczególne parametry występujące w linii komendy rozkazu COMP mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym COMP, określa miejsce, w którym znajduje się program obsługi rozkazu COMP,

[d1:][ścieżka1][nazwa1[.rozszerzenie1]] oznacza zbiór (grupę zbiorów) porównywany ze zbiorem (grupą zbiorów) określonych następnymi parametrami,

[d2:][ścieżka2][nazwa2[.rozszerzenie2]] oznacza zbiór (grupę zbiorów) porównywany ze zbiorami (grupą zbiorów) określonych poprzednimi parametrami.

Uwagi dotyczące posługiwania się rozkazem COMP:

1. Zbiory porównywane ze sobą mogą znajdować się na tych samych lub różnych urządzeniach, w tych samych lub różnych katalogach.
2. W specyfikacji linii komendy rozkazu COMP można używać znaków specjalnych '*' i '?', np. rozkaz COMP o następującej postaci:

A>COMP A:*.ASM B:*.BAK

spowoduje wykonanie porównania każdego zbioru o rozszerzeniu .ASM, zapisanego w katalogu roboczym na urządzeniu A: ze zbiorami o odpowiednio takich samych nazwach, ale rozszerzeniach .BAK, zapisanych w katalogu roboczym na urządzeniu B:.

3. Jeśli w linii komendy rozkazu COMP pominięto wszystkie parametry określające zbiory porównywane, lub pominięto tylko drugi parametr, to system operacyjny PC DOS zażąda ich wyspecyfikowania. Jeśli oba parametry zawierają tylko oznaczenia urządzeń lub urządzeń i katalogów, wtedy program obsługi rozkazu COMP domyślnie przyjmie za nazwy i rozszerzenia zbiorów *.*.
4. Podczas wykonywania operacji porównywania, program obsługi rozkazu COMP wyświetla nazwy aktualnie porównywanych zbiorów. System operacyjny może także wyświetlić odpowiedni komunikat o błędzie, jeśli:
 - do zbioru określonego pierwszym parametrem nie można 'dobrać' zbioru określonego drugim parametrem, w celu wykonania porównania. Ma to miejsce szczególnie wtedy, gdy w specyfikacji linii komendy rozkazu COMP użyto znaków specjalnych: '*' i '?',
 - zbiory porównywane mają różne rozmiary,
 - w linii komendy rozkazu COMP wyspecyfikowano nieistniejący katalog.
5. Rozkaz COMP pozwala wykonywać porównywanie wszystkich zbiorów zapisanych w jednym katalogu z odpowiadającymi im zbiorami zapisanymi w innym katalogu.
6. Jeśli program obsługi rozkazu COMP nie może 'dobrać', dla wykonania porównania, do zbioru (grupy zbiorów) określonych drugim parametrem żadnego zbioru (grupy zbiorów) określonych pierwszym parametrem, to specjalnym komunikatem zażąda wyspecyfikowania tego zbioru (grupy zbiorów).
7. Program obsługi rozkazu COMP wykonuje porównanie obu zbiorów bajt po bajcie. W przypadku wykrycia niezgodności w wartościach dwóch odpowiadających sobie bajtów zostaje wyświetlony komunikat o błędzie, zawierający informacje, które kolejne bajty, licząc od początku zbioru różnią się od siebie (czyli jest to offset li-

czony od początku zbioru), oraz wartości tych bajtów w obu zbiorach. Postać tego komunikatu jest następująca:

```
Compare error of OFFSET xxxxxxxx  
FILE1=xx  
FILE2=yy
```

gdzie FILE1 i FILE2 to określenia zbiorów.

Po zakończeniu porównywania, w zależności od jego rezultatów, zostaje wyświetlony jeden z następujących komunikatów:

- jeśli wystąpiły różnice, to komunikat ma następującą postać:

```
xx Mismatches - ending compare
```

gdzie xx jest liczbą par bajtów z obu zbiorów, które różniły się między sobą.

- jeśli oba zbiory były identyczne, to komunikat ma następującą postać:

```
Files compare OK
```

Po zakończonym porównywaniu wszystkich odpowiadających sobie zbiorów, program obsługi rozkazu COMP wyświetla kolejny komunikat:

```
Compare more files (Y/N)?_
```

Wciśnięcie klawisza z literą 'N' definitywnie kończy wykonywanie rozkazu COMP. Wciśnięcie klawisza z literą 'Y' umożliwia wyspecyfikowanie kolejnych zbiorów do porównywania (system zapyta o nie).

8. Program obsługi rozkazu COMP wykonuje porównywanie aż do chwili wykrycia w jednym ze zbiorów porównywanych znaku sterującego Ctrl/Z (1AH), oznaczającego koniec zbioru tekstowego (a zatem rozkaz COMP nie może być efektywnie wykorzystywany do porównywania dwóch zbiorów binarnych). Wykrycie tego znaku zawsze kończy operację porównywania, wykonywaną dla danej pary zbiorów. W przypadku, gdy znak Ctrl/Z nie wystąpił, system wyświetla następujący komunikat:

EOF mark not found

U W A G A :

Rozkaz COMP może generować komunikaty o niezgodnościach w obu zbiorach, nawet w przypadku, gdy są one identyczne. Ma to miejsce wtedy, gdy zbiory te nie są zakończone znakami Ctrl/Z. Wtedy kryterium zakończenia porównywania jest porównanie tylu par bajtów, ile wynosi rozmiar każdego zbioru (rozmiary muszą być identyczne patrz podpunkt poniżej) zapisany w odpowiadającym mu katalogu. System operacyjny PC DOS dla danego zbioru zaokrągla w górę jego rozmiar do pełnej wielokrotności 128 i dopiero tę wartość wpisuje do katalogu. A zatem w przypadku braku znaku Ctrl/Z porównanie może przekroczyć faktyczne końce obu zbiorów i wykazać istnienie niezgodności pomiędzy bajtami 'dopełniającymi' zbioru do pełnej wielokrotności 128.

9. Rozkaz COMP nie zostanie zainicjowany jeśli rozmiary obu porównywanych zbiorów, zapisane w katalogach są różne (patrz uwaga powyżej).
10. Jeśli w specyfikacji drugiego parametru wyszczególniono tylko nazwę urządzenia, to program obsługi rozkazu COMP domyślnie przyjmie, że porównanie ma dotyczyć zbiorów o takich samych nazwach i rozszerzeniach, jak podano w specyfikacji pierwszego parametru.
11. Rozkaz COMP nie wyświetla osobnego komunikatu zawieszającego wykonywanie operacji porównywania na czas wymiany dyskietki. A zatem, jeśli któryś z porównywanych zbiorów nie został zapisany na tym samym nośniku co program obsługi rozkazu COMP, to należy zainicjować ten rozkaz bez podawania parametrów w linii komendy. Spowoduje to załadowanie programu obsługi rozkazu COMP do pamięci operacyjnej i po wyświetleniu komunikatu żądającego wyspecyfikowania zbiorów źródłowych, umożliwi wymianę nośnika.

Przykłady:

Rozkaz COMP o następującej postaci:

A>COMP B:*.ASM C:

spowoduje wykonanie porównania wszystkich zbiorów o rozszerzeniach .ASM, zapisanych w katalogu roboczym na urządzeniu B:, ze zbiorami o takich samych nazwach i rozszerzeniach, zapisanych w katalogu roboczym na urządzeniu C:.

Rozkaz COMP o następującej postaci:

A>COMP A:\LEVEL1 A:\LEVEL2

spowoduje wykonanie porównania wszystkich zbiorów, zapisanych w katalogu LEVEL1 na urządzeniu A:, ze zbiorami o takich samych nazwach i rozszerzeniach, zapisanych w katalogu LEVEL2, także na urządzeniu A:.

6.10. Rozkaz COPY.

Format:

COPY [/A[/B][d1:][ścieżka1]nazwa1[.rozszerzenie1][[/A[/B][d2:][ścieżka2]nazwa2[.rozszerzenie2]][[/A[/B][V]

lub:

COPY [/A[/B][d1:][ścieżka1]nazwa1[.rozszerzenie1][[/A[/B][+[d2:][ścieżka2]nazwa2[.rozszerzenie2][[/A[/B]...][dn:][ścieżkan]nazwan[.rozszerzenien]][[/A[/B][V]

lub:

COPY [/A[/B][d1:][ścieżka1]nazwa1[.rozszerzenie1][[/A[/B][+[[,],d2:][ścieżka2]nazwa2[.rozszerzenie2][[/A[/B]...][dn:][ścieżkan]nazwan[.rozszerzenien]][[/A[/B][V]

czyli:

COPY (źródło) (przeznaczenie)

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz COPY pozwala na kopiowanie pojedynczego zbioru lub grupy zbiorów, zapisanych w różnych lub na tych samych urządzeniach, w różnych lub w tych samych katalogach. Pierwsze specyfikacje odnoszą się do zbiorów źródłowych, drugie - do zbioru (zbiorów) wynikowych.

U W A G A :

Rozkaz COPY nie jest odpowiednikiem rozkazu BACKUP. Rozkazy: BACKUP i XCOPY pozwalają na kopiowanie zarówno zbiorów jak i podkatalogów zapisanych w danym katalogu. Rozkaz COPY wykonuje tylko kopie zbiorów, a nie podkatalogów zapisanych w katalogu roboczym lub określonych w linii komendy.

Jeśli kopia ma zostać na tym samym urządzeniu, to nazwy zbiorów: źródłowego i wynikowego muszą być różne, lub zbiory te będą kopiowane pomiędzy różnymi katalogami (poprzez nazwę zbioru rozumie się tutaj jego nazwę 'właściwą' i rozszerzenie). Podczas operacji kopiowania zbiory mogą być także łączone ze sobą w jeden lub więcej zbiorów wynikowych. Rozkaz COPY umożliwia ponadto transfer danych pomiędzy wszystkimi urządzeniami systemowymi.

Poszczególne klucze występujące w linii komendy rozkazu COPY mają następujące znaczenie:

klucz /A:

- użyty w specyfikacji zbiorów źródłowych spowoduje, że będą one traktowane jako zbiory tekstowe (ASCII), a kopiowanie odbywać się będzie do momentu napotkania pierwszego znalezionej znacznika końca zbioru (CTRL/Z, czyli 1AH),
- użyty w specyfikacji zbioru wynikowego spowoduje dodanie znaku CTRL/Z (1AH) jako ostatniego znaku zbioru.

klucz /B:

- użyty w specyfikacji zbiorów źródłowych spowoduje, że będą one kopiowane w całości, bez względu na występowanie znaków końca zbioru (CTRL/Z),

- użyty w specyfikacji zbioru wynikowego spowoduje, że żaden znak końca zbioru (CTRL/Z) nie będzie umieszczony w kopii.

klucz /V wymusza wykonywanie weryfikacji poprawności utworzonej kopii. Sprawdzenie to polega na zbadaniu, czy kopia została utworzona poprawnie pod względem formalnym, a nie na porównywaniu jej ze zbiorem źródłowym. Jeśli w systemie zainicjowano uprzednio rozkaz VERIFY ON, to kontrola poprawności zapisu zostanie przeprowadzona dwukrotnie: raz w wyniku opcji /V, a drugi - właśnie dzięki rozkazowi VERIFY ON. Zastosowanie klucza /V powoduje wydłużenie czasu wykonywania rozkazu COPY, ze względu na dodatkową weryfikację.

U W A G A :

1. Kopia zbioru o atrybucie RO (ang. Read Only - tylko odczyt, patrz rozkaz ATTRIB), wykonana rozkazem COPY, nie zachowuje tego atrybutu.
2. Nie można używać rozkazu COPY do kopiowania danych (zbiorów) pomiędzy portami szeregowymi: COM1, COM2 i AUX.

Rozkaz COPY można wykonywać trzema różnymi sposobami:

1. Kopiowanie z zachowaniem nazw zbiorów.
2. Kopiowanie zbiorów ze zmianą nazw.
3. Kopiowanie zbiorów połączone z ich łączeniem w jeden lub kilka zbiorów wynikowych.

1. Kopiowanie z zachowaniem nazw zbiorów.

Sposób ten pozwala tak wykonywać kopie zbiorów, że zachowują one nazwy i rozszerzenia zbiorów źródłowych:

```
COPY [d1:][ścieżka1]nazwa[.rozszerzenie1]
```

, lub:

```
COPY [d1:][ścieżka1]nazwa[.rozszerzenie1] d2:[ścieżka2]
```

Zastosowanie pierwszego formatu umożliwia kopiowanie zbioru, który został wyspecyfikowany w linii komendy, na urządzenie robocze do katalogu roboczego. Drugi format jest uogólnieniem pierwszego, gdyż umożliwia skopiowanie wyspecyfikowanego zbioru na urządzenie d2:, do katalogu określonego parametrem 'ścieżka2'. W obu przypadkach transfer nie może odbywać się pomiędzy tymi samymi katalogami na jednym urządzeniu. Ponieważ nie podano ani nazwy, ani rozszerzenia zbioru wyjściowego, to zachowa on nazwę i rozszerzenie zbioru źródłowego.

Przykłady:

Rozkaz COPY o następującej postaci:

```
A>COPY B:MYPROG
```

spowoduje wykonanie kopii zbioru MYPROG z urządzenia B: na robocze urządzenie A: (do katalogu roboczego).

Rozkaz COPY o następującej postaci:

```
A>COPY *.* B:
```

spowoduje skopiowanie wszystkich zbiorów zapisanych w katalogu roboczym z urządzenia A: na urządzenie B:. Po wykonaniu kopii każdego zbioru, program obsługi rozkazu COPY wyświetla na ekranie monitora jego nazwę i rozszerzenie.

Rozkaz COPY o następującej postaci:

```
A>COPY B:\MYPROG B:\LEVEL1
```

spowoduje wykonanie kopii zbioru MYPROG z katalogu podstawowego do katalogu podrzędnego LEVEL1, znajdujących się na urządzeniu B:. Należy zwrócić uwagę, że przy tak wyspecyfikowanym rozkazie, katalog \LEVEL1 musi fizycznie istnieć w katalogu podstawowym, gdyż inaczej program obsługi rozkazu COPY potraktuje nazwę katalogu LEVEL1 jako nazwę kopii zbioru MYPROG i kopiowanie zostanie wykonane w ramach katalogu podstawowego.

2. Kopiowanie zbiorów ze zmianą nazw.

Sposób ten pozwala na takie wykonywanie kopiowania, że otrzymane kopie mają inne nazwy lub rozszerzenia niż ich zbiory źródłowe:

```
COPY [d1:][ścieżka1]nazwa1[.rozszerzenie1]
      [ścieżka2]nazwa2[.rozszerzenie2]
```

lub:

```
COPY [d1:][ścieżka1]nazwa1[.rozszerzenie1]
      d2:[ścieżka2]nazwa2[.rozszerzenie2]
```

Zastosowanie pierwszego formatu pozwala na kopiowanie zbiorów w ramach tego samego urządzenia oraz tych samych lub różnych katalogów. Drugi format uwzględnia dodatkowo wykonywanie kopii pomiędzy różnymi urządzeniami. W obu przypadkach wykonane kopie będą miały inne nazwy lub rozszerzenia niż ich zbiory źródłowe. Pominięcie opcjonalnych parametrów 'd1:', 'ścieżka1' lub 'ścieżka2' powoduje podstawienie ich wartości domyślnych, tj. urządzenia i katalogu roboczego.

Przykłady:

Rozkaz COPY o następującej postaci:

```
A>COPY MYPROG.ABC B:*.XXX
```

spowoduje wykonanie kopii zbioru źródłowego MYPROG.ABC na urządzeniu B: pod nazwą MYPROG, ale ze zmienionym rozszerzeniem .XXX.

Omawiany sposób pozwala nie tylko na transfer danych pomiędzy nośnikami magnetycznymi takimi jak dyski twarde czy dyskietki, ale także z uwzględnieniem następujących urządzeń: CON, AUX, COM1, COM2, PRN, LPT1, LPT2, LPT3 i NUL. Transfer taki powinien uwzględniać rodzaj urządzenia (np. PRN nie może występować jako urządzenie źródłowe) i nie może być wykonywany pomiędzy dwoma różnymi portami szeregowymi:

- COPY CON FILE.TXT - kopiowanie z konsoli CON na zbiór FILE.TXT,
- COPY CON AUX - kopiowanie z konsoli CON na port szeregowy AUX,
- COPY CON LPT1 - kopiowanie z konsoli CON na drukarkę LPT1,
- COPY FILE.TXT CON - kopiowanie zbioru FILE.TXT na monitor konsoli,
- COPY COM1 LPT1 - kopiowanie z portu szeregowego COM1 na drukarkę LPT1,
- COPY COM2 CON - kopiowanie z portu szeregowego COM1 na monitor konsoli,
- COPY COM1 COM2 - błędny rozkaz, nie wolno wykonywać kopiowania pomiędzy dwoma portami szeregowymi.

Aby utworzyć zbiór o nazwie FILE.TXT bezpośrednio z klawiatury, należy wywołać następujący rozkaz COPY:

A>COPY CON FILE.TXT

a następnie wpisać tekst zbioru bezpośrednio z klawiatury:

- 1 linia tekstu zakończona wciśnięciem klawisza 'Enter'
- 2 linia tekstu zakończona wciśnięciem klawisza 'Enter'

ostatnia linia tekstu zakończona wciśnięciem klawisza 'Enter'

Zakończenie wpisywania wciśnięciem klawisza 'F6', a następnie klawisza 'Enter'

3. Wykonywanie kopiowania zbiorów, połączone z ich łączeniem w jeden lub kilka zbiorów wynikowych:

Sposób ten umożliwia łączenie większej ilości różnych zbiorów w jeden lub kilka zbiorów wynikowych, poprzez dołączanie kolejnych zbiorów do końca tworzonego zbioru wynikowego:

```
COPY [/A][/B][d1:][ścieżka][nazwa[.rozszerzenie]][/A][B]
[+d1:][ścieżka][.rozszerzenie]][/A][B]...
[d:][ścieżka][nazwa[.rozszerzenie]][/A][B][V]
```

Przykłady:

Rozkaz COPY o następującej postaci:

A>COPY A.XYZ+B.ABC+B:C.TXT BIGFILE.TXT

spowoduje utworzenie w katalogu roboczym, na urządzeniu roboczym zbioru o nazwie BIGFILE.TXT ze zbiorów: A.XYZ, B.ABC i B:C.TXT.

Rozkaz COPY o następującej postaci:

A>COPY A.ASM+B.ASM

spowoduje utworzenie nowego zbioru A.ASM ze zbiorów A.ASM oraz B.ASM, przy czym stara wersja zbioru A.ASM zostanie skasowana.

Łączenie zbiorów ze sobą ma charakter łączenia ciągów znaków ASCII. A zatem znak sterujący 'Ctrl/Z' (1AH), oznaczający koniec zbioru i występujący w każdym zbiorze źródłowym, jest domyślnie, bez względu na swoje położenie wewnątrz tego zbioru, interpretowany jako znak końca zbioru. W przypadku łączenia zbiorów tekstowych interpretacja taka nie wnosi żadnego błędu, gdyż w każdym takim zbiorze znak 'Ctrl/Z' występuje tylko raz, właśnie na końcu zbioru. W przypadku zbiorów binarnych (np. o rozszerzeniu .COM) tego typu interpretacja prowadziłaby do kopiowania tylko fragmentu zbioru, do momentu napotkania pierwszego znaku o kodzie hexadecimalnym 1AH. Aby temu zapobiec kopiowanie i łączenie zbiorów binarnych musi być wykonywane z kluczem /B. Rozkaz COPY umożliwia ponadto łączenie ze sobą zbiorów tekstowych i binarnych poprzez stosowanie klucza /A dla zbiorów tekstowych i klucza /B dla zbiorów binarnych.

Rozkaz COPY o następującej postaci:

A>COPY A.XYZ+B.COM/B+B:C.TXT/A BIGFILE.TXT

spowoduje utworzenie zbioru tekstowego BIGFILE.TXT ze zbioru tekstowego B:C.TXT, zbioru binarnego B.COM i zbioru 'o niejasnym charakterze' A.XYZ. Należy zwrócić uwagę, że klucz /B, wyspecyfikowany po zbiorze binarnym B.COM, odnosi się także do zbioru A.XYZ.

Dla łączenia kilku zbiorów można wykorzystywać znaki specjalne: '?' i '*', np. rozkaz COPY o następującej postaci:

A>COPY *.LST COMBIN.PRN

spowoduje skopiowanie wszystkich zbiorów o rozszerzeniu .LST z katalogu roboczego w zbiór wynikowy COMBIN.PRN. Kolejność dopisywania poszczególnych zbiorów do zbioru wynikowego jest zdeterminowana przez kolejność ich występowania w katalogu roboczym.

Rozkazem COPY można tworzyć także kilka zbiorów wynikowych. Przykładem jest tutaj następujący rozkaz COPY:

A>COPY *.LST+*.REF *.PRN

Jeśli katalog roboczy zawiera następujące zbiory:

FILE1.LST
FILE2.LST
FILE3.REF
FILE4.REF

to zbiory FILE1.LST i FILE3.REF utworzą zbiór FILE1.PRN, a zbiory FILE2.LST i FILE4.REF utworzą zbiór FILE2.PRN (w katalogu pozostaną także zbiory źródłowe).

Kopiowanie przy pomocy znaków specjalnych '*' i '?' nie zawsze jest jednak wykonywane zgodnie z intencjami użytkownika. Jeśli w katalogu roboczym istnieje zbiór np. ALL.LST i zainicjowano następujący rozkaz COPY:

A>COPY *.LST ALL.LST

to powstanie nowy zbiór ALL.LST, który jednak nie będzie zawierał 'starego' zbioru ALL.LST. Jest to spowodowane tym, że nazwa każdego zbioru wejściowego jest porównywana z nazwą, jaką użytkownik zamierza nadać zbiorowi wyjściowemu. Jeśli nazwy te są takie same, to podczas kopiowania zbiór wejściowy o zbieżnej nazwie ze zbiorem wynikowym jest pomijany, a system operacyjny PC DOS wyświetla na ekranie monitora następujący komunikat:

Content of destination lost before copy

Wyjściem z tej sytuacji jest rozkaz COPY o następującej postaci:

A>COPY ALL.LST+*.LST

a więc realizujący kopiowanie bez zmiany nazwy. W tym przypadku specyfikacja *.LST nie obejmuje zbioru ALL.LST (wynika to z mechanizmu kontroli wbudowanego w program obsługi rozkazu COPY).

Podczas kopiowania zbiorów, bez ich łączenia ze sobą, zbiory wynikowe zachowują datę i czas utworzenia ich zbiorów źródłowych. Dla ich skopiowania z aktualną datą i czasem należy użyć rozkazu COPY o następującej postaci:

A>COPY B:XYZ.ASM+

Spowoduje to skopiowanie zbioru B:XYZ.ASM do katalogu roboczego na urządzeniu A:, z nową datą i czasem utworzenia. W przypadku 'uaktualniania' daty i czasu zapisu pojedynczego zbioru w obrębie tego samego katalogu i na tym samym urządzeniu można użyć następującego rozkazu COPY:

A>COPY B:XYZ.ASM+,, B:

Należy zwrócić uwagę, że rozkaz o postaci:

A>COPY B:*.*,,, B:

nie zmieni daty i czasu powstania wszystkich zbiorów, lecz skopiuje wszystkie zbiory z katalogu roboczego w jeden zbiór wyjściowy, nadając mu nazwę pierwszego skopiowanego zbioru ('stara' wersja tego zbioru zostanie skasowana, ale wszystkie pozostałe zbiory nadal będą istnieć).

U W A G A

W przypadku, gdy żaden zbiór źródłowy, wyspecyfikowany w linii komendy rozkazu COPY, nie został skopiowany, system wyświetla następujący komunikat:

0 file(s) copied

6.11. Rozkaz CTTY (Change Console).

Format:

CTTY device-name

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz CTTY pozwala na zmianę standardowej, wejściowo-wyjściowej konsoli na inną, podłączoną do jednego z wejściowo-wyjściowych portów szeregowych jako konsola pomocnicza. Parametr 'device-name' może określać jeden z portów szeregowych: AUX, COM1, COM2. W przypadku podstawienia za ten parametr nazwy CON zostanie przywrócona klawiatura i ekran monitora jako standardowe urządzenia wejściowo-wyjściowe.

Uwagi dotyczące posługiwania się rozkazem CTTY:

1. Rozkaz CTTY akceptuje nazwy tylko tych urządzeń, które zapewniają dwukierunkową transmisję danych. Oznacza to, że nie można podać nazwy np. PRN (drukarki), ponieważ system operacyjny PC DOS nie może odczytywać danych z tego urządzenia.
2. Rozkaz CTTY zapewnia efektywną pracę tylko tych programów, które odwołują się do funkcji samego systemu operacyjnego. Pewne programy napisane np. w języku BASIC nie odwołują się do funkcji PC DOS, a więc nie są w stanie wykorzystywać zmiany standardowych urządzeń wejściowych i wyjściowych.

Przykłady:

Rozkaz CTTY o następującej postaci:

CTTY COM1

spowoduje, że system operacyjny PC DOS przyporządkowuje jako standardowe urządzenie we/wy dowolne urządzenie przyłączone do złącza szeregowego COM1.

Rozkaz CTTY o następującej postaci:

CTTY CON

spowoduje, że system operacyjny PC DOS przywraca klawiaturę i ekran monitora jako standardowe urządzenie we/wy.

6.12. Rozkaz DATE.

Format:

DATE [mm-dd-yy]

lub:

DATE [dd-mm-yy]

lub:

DATE [yy-mm-dd]

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz DATE umożliwia wprowadzenie lub zmianę daty systemu. Poszczególne parametry występujące w linii komendy rozkazu DATE mają następujące znaczenie:

mm określa miesiąc. Określeniem miesiąca może być jedno lub dwucyfrowa liczba z zakresu od 1 do 12,

dd określa dzień. Określeniem dnia może być jedno lub dwucyfrowa liczba z zakresu od 1 do 31,

yy określa rok. Określeniem roku może być dwucyfrowa liczba z zakresu od 80 do 99 (80 - odpowiada rok 1980, 81 - rok 1981 itd.).

Uwagi dotyczące posługiwania się rozkazem DATE:

1. Jeśli wyspecyfikowano rozkaz DATE bez podania parametrów, to na ekranie monitora pojawi się następujący komunikat:

Current date is day mm-dd-yy
Enter new date (mm-dd-yy):_

Format daty (mm-dd-yy, dd-mm-yy, yy-mm-dd) zależy od parametrów instrukcji COUNTRY w zbiorze CONFIG.SYS.

2. Separatorami pomiędzy oznaczeniami dni, miesięcy i lat mogą być: znak minusa (-), slash (/), lub kropka dziesiętna (.). Np. dzień 23 września 1984 roku może być wyspecyfikowany trzema równoważnymi sposobami:

9-23-84
9/23/84
9.23.84

3. Jeśli data została wyspecyfikowana poprawnie pod względem formalnym, to zostanie wyświetlona na ekranie monitora. W przeciwnym wypadku system operacyjny PC DOS wyświetli następujący komunikat:

Invalid date
Enter new date (mm-dd-yy):_

4. System operacyjny PC DOS wyświetla dodatkową informację w postaci dnia tygodnia. W specyfikacji linii komendy rozkazu DATE dnia tygodnia nie trzeba podawać.
5. Zmianę daty można wykonywać ze standardowego urządzenia wejściowego (klawiatura) lub ze zbioru wsadowego. Dla zapewnienia możliwości uaktualniania daty przy każdym restarcie systemu, zbiór AUTOEXEC.BAT powinien zawierać rozkaz DATE.
6. Wciśnięcie klawisza "Enter" przed wprowadzeniem nowej daty powoduje zachowanie daty dotychczas obowiązującej w systemie.
7. W mikrokomputerze IBM PC/AT wyspecyfikowanie nowej daty nie zmienia czasu systemowego.

Przykład:

W poniższym przykładzie przedstawiono wyświetlenie i zmianę daty:

DATE

Current date is Mon 1-18-1984

Enter new date: 7/24/84_

6.13. Rozkaz DEL (Delete).**Format:****DEL [d:][ścieżka]nazwa[.rozszerzenie]****Typ:**

Rozkaz wewnętrzny.

Opis:

Rozkaz DEL (Delete) usuwa zawartość wyspecyfikowanego zbioru (zbiorów), jeśli nie posiada on ustawionego atrybutu RD (ang. Read Only - patrz opis rozkazu ATTRIB).

Poszczególne parametry występujące w linii komendy rozkazu DEL mają następujące znaczenie:

[d:] określa urządzenie, na którym znajduje się nośnik ze zbiorem przeznaczonym do usunięcia. Jest to parametr opcjonalny i w razie jego pominięcia system operacyjny PC DOS domyślnie przyjmie urządzenie robocze,

[ścieżka] określa katalog ze zbiorem przeznaczonym do usunięcia. Jest to także parametr opcjonalny i w razie jego pominięcia system operacyjny PC DOS domyślnie przyjmie katalog roboczy,

nazwa[.rozszerzenie] określają nazwę i rozszerzenie zbioru przeznaczonego do usunięcia. Można tutaj używać znaków specjalnych '?' i '*'.

Uwagi dotyczące posługiwania się rozkazem DEL:

1. Rozkaz DEL pozwala na usunięcie zarówno pojedynczego zbioru jak i całej grupy zbiorów. Jeśli zamiast nazwy i rozszerzenia wyspecyfikowano \$.\$, sygnalizując zamiar usunięcia wszystkich zbiorów zapisanych w katalogu, to system operacyjny PC DOS żąda dodatkowego potwierdzenia wyświetlając następujące pytanie:

Are you sure (Y/N)?

2. Rozkazu DEL nie można używać do usuwania katalogów podrzędnych. Do tego celu służy rozkaz RMDIR (patrz opis).
3. Należy zwrócić szczególną uwagę podczas wykonywania kasowania po zainicjowaniu rozkazów ASSIGN, JOIN i SUBST.

Kasowanie zbioru nie polega na skasowaniu jego zapisu z dysku lub dyskietki, lecz na odpowiednich zmianach w tzw. File Allocation Table, zwalniających miejsca zarezerwowane na ten zbiór. Zbiór taki może zostać odtworzony, o ile nie zapisano w tym samym miejscu innych informacji lub nie zaformatowano dysku lub dyskietki. Programy odtwarzające skasowane zbiory nie są opisane w tej instrukcji.

Przykłady:

Rozkaz DEL o następującej postaci:

DEL A:FIL.DAT

spowoduje usunięcie zbioru FIL.DAT, zapisanego na dyskietce załączonej na urządzenie A: w katalogu roboczym.

Rozkaz DEL o następującej postaci:

DEL C:*.BAS

spowoduje usunięcie wszystkich zbiorów o rozszerzeniu .BAS zapisanych na dysku C: w katalogu roboczym.

Rozkaz DEL o następującej postaci:

DEL C:\LEVEL1\LEVEL2

spowoduje usunięcie wszystkich zbiorów zapisanych na dysku C:, w katalogu podrzędnym LEVEL2.

6.14. Rozkaz DIR (Directory).

Format:

DIR [d:][ścieżka][nazwa[.rozszerzenie]][/P][/W]

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz DIR pozwala na wyświetlenie na ekranie monitora zawartości wyspecyfikowanego w linii komendy katalogu lub zawartości katalogu roboczego. W zależności od parametrów rozkazu DIR, wyświetlanie może objąć całą zawartość katalogu lub tylko określoną grupę zbiorów. Rozkazem DIR nie można wykazać obecności na danym nośniku dwóch ukrytych zbiorów systemowych:

IBMBIO.COM

IBMDOS.COM

gdyż nie są one przypisane do żadnego katalogu.

Poszczególne parametry występujące w linii komendy rozkazu DIR mają następujące znaczenie:

[d:][ścieżka][nazwa][.rozszerzenie] określa zbiór lub grupę zbiorów których nazwy, rozszerzenia i pewne dodatkowe informacje zostaną wyświetlone rozkazem DIR, o ile znajdują się one w wyspecyfikowanym (roboczym) katalogu. Dodatkowe informacje dotyczące każdego zbioru to czas i data jego utworzenia oraz jego długość wyrażona w bajtach.

klucz /P wymusza zawieszenie wyświetlania informacji w momencie zapełnienia całego ekranu, z następującym komunikatem:

Strike a key when ready ...

który oznacza, że wykonywanie rozkazu DIR będzie kontynuowane po wciśnięciu klawisza dowolnej litery.

klucz /W wymusza wyświetlenie zawartości katalogu w kilku "słupkach", co pociąga za sobą pominięcie informacji o datach i czasach utworzenia zbiorów oraz danych dotyczących ich długości.

Uwagi dotyczące posługiwania się rozkazem DIR:

1. Informacja wyświetlona przy pomocy rozkazu DIR zawiera nazwę urządzenia, etykietę dysku lub dyskietki oraz całkowitą ilość wolnego miejsca na dysku, wyrażoną w bajtach. Liczba określająca ilość wolnego miejsca na nośniku zostaje przy tym zaokrąglona do najbliższej, całkowitej wielokrotności liczby 1024. W linii zawierającej nazwę i rozszerzenie danego zbioru znajduje się także informacja o długości tego zbioru (długość wyrażona w bajtach jest liczbą dziesiętną) oraz data i czas jego utworzenia.
2. Program obsługi rozkazu DIR wyróżnia katalogi podrzędne, zapisane w wyspecyfikowanym (roboczym) katalogu wyświetleniem słowa <DIR> w miejscu, w którym w przypadku zbiorów, są podawane informacje o ich długościach i rozszerzeniach.
3. Program obsługi rozkazu DIR dopuszcza stosowanie w linii komendy znaków specjalnych '*' i '?'. Pominięcie w specyfikacji grupy zbiorów rozszerzenia jest traktowane jako wpisanie w tym miejscu znaku '*'.
4. Dla wyświetlenia informacji dotyczących tylko zbiorów bez rozszerzeń, należy w linii komendy rozkazu DIR wpisać, po nazwie zbioru, znak kropki (.). Wówczas program obsługi rozkazu DIR nie przyjmie domyślnie za rozszerzenie znaków specjalnego '*' lecz zinterpretuje to jako brak rozszerzenia.
5. Linia komendy rozkazu DIR może zawierać oba opcjonalne klucze /P i /W.
6. W pewnych przypadkach szybkość wyświetlania zawartości danego katalogu może wzrosnąć po zadeklarowaniu instrukcją BUFFER w zbiorze CONFIG.SYS (patrz rozdział pt. 'KONFIGUROWANIE SYSTEMU OPERACYJNEGO') co najmniej trzech buforów systemowych.

Przykłady:

Rozkaz DIR o następującej postaci:

A>DIR

spowoduje wyświetlenie na ekranie monitora zawartości całego katalogu roboczego z urządzenia A:. Przykładowy taki listing, w przypadku gdy katalog roboczy jest jednocześnie katalogiem podstawowym, pokazano poniżej:

Volume in drive A is MYDISK
Directory of A:\

FILE1	ASM	10368	7-20-83	12:13p
FILE3	ASM	1613	5-27-83	12:14p
9X		31	8-17-82	10:59a
LEVEL2	<DIR>		9-09-82	12:10p
FILE1		2288	9-02-82	12:25p

5 File(s) 141312 bytes free

U W A G A :

W powyższym przykładzie daty i czasy utworzenia poszczególnych zbiorów i podkatalogu LEVEL2 zostały podane w formacie obowiązującym w USA. W przypadku używania innego formatu, program obsługi rozkazu DIR dostosowuje się do niego.

Rozkaz DIR o następującej postaci:

A>DIR C:

spowoduje wyświetlenie na ekranie monitora zawartości całego katalogu roboczego z urządzenia C:.

Rozkaz DIR o następującej postaci:

A>DIR \LEVEL2

spowoduje wyświetlenie na ekranie monitora zawartości całego katalogu LEVEL2 z urządzenia roboczego. Przykładowy taki listing pokazano poniżej:

Volume in drive A is MYDISK
Directory of A:\LEVEL2

```
.      <DIR>                9-09-84  1:30p
..     <DIR>                9-09-84  1:30p
MYPROG COM      2463        7-30-84  8:55a
      3 File(s)  141312 bytes free
```

Należy zwrócić uwagę, że katalog LEVEL2, podobnie jak każdy katalog podrzędny, zawiera dwa specjalne katalogi:

```
. <DIR>
.. <DIR>
```

Katalog . <DIR> służy do odwoływania się do katalogu roboczego, natomiast katalog .. <DIR> służy do odwoływania się do katalogu wyższego poziomu, w którym zapisano katalog LEVEL2 (w tym przypadku katalogiem tym jest katalog podstawowy). I tak, jeśli katalogiem roboczym jest katalog LEVEL2, to aby wyświetlić zawartość katalogu podstawowego należy użyć rozkazu DIR o następującej postaci:

A>DIR ..

Rozkaz DIR o następującej postaci:

A>DIR FILE3.ASM

spowoduje wyświetlenie informacji o wszystkich zbiorach z rozszerzeniem .ASM, zapisanych w katalogu podstawowym na urządzeniu A:. Przykładowy taki komunikat podano poniżej:

Volume in drive A is MYDISK
Directory of A:\

```
FILE1  ASM      10368        7-20-83  12:13p
      1 File(s)  141312 bytes free
```

Rozkaz DIR o następującej postaci:

A>DIR *.ASM

spowoduje wyświetlenie informacji o wszystkich zbiorach z rozszerzeniem .ASM, zapisanych w katalogu roboczym na urządzeniu A:. Przykładowy taki komunikat pokazano poniżej:

Volume in drive A is MYDISK
Directory of A:\

FILE1	ASM	10368	7-20-83	12:13p
FILE3	ASM	1613	5-27-83	12:14p
2 File(s) 141312 bytes free				

Rozkaz DIR o następującej postaci:

A>DIR FILE1

jest równoważny rozkazowi:

A>DIR FILE1.*

i spowoduje wyświetlenie informacji o wszystkich zbiorach o nazwach FILE1, ale dowolnych rozszerzeniach, zapisanych w katalogu roboczym na urządzeniu A:. Przykładowy taki komunikat pokazano poniżej:

Volume in drive A is MYDISK
Directory of A:\

FILE1	ASM	10368	7-20-83	12:13p
FILE		1613	5-27-83	12:14p
2 File(s) 141312 bytes free				

Aby w powyższym przykładzie wyświetlić informacje dotyczące tylko zbiorów o nazwach FILE1, ale bez rozszerzeń, należy użyć następującego rozkazu DIR:

A>DIR FILE1

6.15. Rozkaz DISKCOMP (Compare Diskettes Only).

Format:

[d:][ścieżka]DISKCOMP [d1:[d2:]][/1]/[8]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz DISKCOMP pozwala na porównywanie informacji zapisanych na dwóch dyskietkach, znajdujących się na wyspecyfikowanych urządzeniach.

Poszczególne parametry występujące w linii komendy rozkazu DISKCOMP mają następujące znaczenie:

[d:] [ścieżka], przed słowem kluczowym DISKCOMP, określa miejsce, w którym znajduje się program obsługi rozkazu DISKCOMP,

[d1:] określa urządzenie, na którym znajduje się dyskietka źródłowa,

[d2:] określa urządzenie, na którym znajduje się dyskietka porównywana z dyskietką na urządzeniu d1:.

klucz /1 (nie występuje w wersji MS DOS 3.10) wymusza porównywanie tylko pierwszych stron obu dyskietek, nawet jeśli stacje dyskietkowe i same dyskietki są dwustronne,

klucz /8 (nie występuje w wersji MS DOS 3.10) pozwala na porównywanie dyskietek, z których co najmniej jedna (na urządzeniu d2:) została zaformatowana z 8 sektorami na ścieżce. Format ten jest wykorzystywany przez wersje PC DOS 1.10, wersja 3.20 domyślnie formatuje 9/15 sektorów na ścieżce (patrz rozkaz FORMAT).

Uwagi dotyczące posługiwania się rozkazem DISKCOMP:

1. Rozkaz DISKCOMP wykonuje porównania całych dyskietek, sektor po sektorze, a nie jak rozkaz COMP zbior po zbiorze. Oznacza to, że mogą być zasygnalizowane różnice, jeśli kopie dyskietki wykonano innym rozkazem niż DISKCOPY (np. rozkazem COPY lub XCOPY - w takim przypadku zawartość obu dyskietek jest jednakowa, ale rozmieszczenie zbiorów na obu nośnikach różne). Komunikat o wykrytych różnicach zawiera numer strony, ścieżki i sektora na ścieżce, w którym zapis różni się.
2. Rozkazu DISKCOMP można używać tylko do dyskietek. Próba wyspecyfikowania w linii komendy rozkazu DISKCOMP stacji dysków twardych lub dysku wirtualnego (patrz instrukcja DEVICE w rozdziale pt. "KONFIGUROWANIE SYSTEMU OPERACYJ-

NEGO') kończy się komunikatem o wykryciu błędu.

3. W linii komendy rozkazu DISKCOMP można wyspecyfikować oba jednakowe parametry 'd1:' i 'd2:'. W takim przypadku program obsługi rozkazu DISKCOMP odpowiednimi komunikatami, wyświetlanymi na ekranie monitora, będzie sterował wymianą porównywanych dyskietek na wskazanym urządzeniu. Użytkownik może również pominąć oba parametry 'd1:' i 'd2:'. Wówczas porównywane będą dyskietki kolejno zakładane na urządzenie robocze. Jeśli pominięty zostanie parametr 'd2:', to system operacyjny PC DOS domyślnie przyjmie zamiast niego urządzenie robocze.
4. Zakończenie porównywania system operacyjny sygnalizuje następującym komunikatem:

Compare more diskettes (Y/N)?_

Jeśli użytkownik naciśnie klawisz z literą 'Y', to będzie mógł wykonać następne porównywania zawartości innych dyskietek założonych na poprzednio wyspecyfikowane stacje, przy czym program obsługi rozkazu DISKCOMP będzie sterował wymianą dyskietek. Wciśnięcie klawisza z literą 'N' kończy wykonywanie rozkazu DISKCOMP.

5. Podczas wykonywania rozkazu DISKCOMP może wystąpić błąd odczytu zawartości jednej z dyskietek. Powoduje to wyświetlenie komunikatu określającego miejsce wystąpienia błędu oraz informacji o wykrytych, w związku z tym błędem, różnicach w zapisie informacji na porównywanych nośnikach. Następnie rozkaz DISKCOMP kontynuuje porównywanie.
6. Program obsługi rozkazu DISKCOMP automatycznie określa numery stron, ścieżek i sektorów z których informacje będą następnie porównywane. Czynność ta jest wykonywana na podstawie dyskietki, która została założona na stację d1:
 - jeśli dyskietka ze stacji d1: została zaformatowana jednostronnie, lub jeśli użyto klucza /1, to porównywanie dotyczy tylko pierwszych stron obu dyskietek, nawet jeśli druga z nich jest dyskietką dwustronną,
 - jeśli dyskietkę ze stacji d1: zaformatowano z 9 (lub z 15) sektorami na ścieżce, to program obsługi rozkazu DISKCOMP zakłada, że także na drugiej dyskietce, informacja została zapisana w takim właśnie formacie (chyba, że użyto klucza /8),
 - jeśli pierwsza stacja i założona na nią dyskietka

jest dyskietką dwustronną i nie wyspecyfikowano kłucza /1, to porównanie będzie obustronne. W takim przypadku system zasygnalizuje błąd zawsze wtedy, gdy druga stacja lub założona na nią dyskietka jest jednostronna.

7. Rozkaz DISKCOMP ignoruje przyporządkowania dokonane wcześniej rozkazami ASSIGN, JOIN i SUBST.

U W A G A :

Rozkaz DISKCOMP może generować następujący komunikat:

Insert disk with \COMMAND.COM in drive A
and strike any key when ready

Oznacza to żądanie założenia na urządzenie A: dyskietki z systemem operacyjnym i potwierdzenie wciśnięciem klawisza dowolnej litery.

Możliwości wykonywania rozkazu DISKCOMP
w zależności od zestawu sprzętowego:

1. Jeśli mikrokomputer posiada stacje dyskietkowe*jednostronne, to rozkaz DISKCOMP pozwala porównywać ze sobą tylko dyskietki jednostronne o pojemności 160/180kB.
2. Jeśli mikrokomputer posiada stacje dyskietkowe dwustronne, to rozkaz DISKCOMP pozwala na porównywanie:
 - dyskietki jednostronnej o pojemności 160/180kB z dyskietką jednostronną o pojemności 160/180kB,
 - dyskietki dwustronnej o pojemności 320/360kB z dyskietką dwustronną o pojemności 320/360kB,
 - dyskietki dwustronnej o pojemności 720kB (tylko w wersji PC DOS 3.20) z dyskietką dwustronną o pojemności 720kB.
3. Jeśli komputer posiada stacje dyskietkowe dwustronne o dużej gęstości zapisu, to rozkaz DISKCOMP pozwala na porównywanie:
 - dyskietki jednostronnej o pojemności 160/180kB z dy-

skietką jednostronną o pojemności 160/180kB,

- dyskietki dwustronnej o pojemności 320/360kB z dyskietką dwustronną o pojemności 320/360kB,
- dyskietki o dużej gęstości zapisu o pojemności 1.2MB z dyskietką o dużej gęstości zapisu o pojemności 1.2MB.

Rozkazu DISKCOMP nie można używać do porównywania dyskietek założonych na dwie stacje różnych typów. W takim przypadku wykonywanie rozkazu DISKCOMP zawsze kończy się następującym komunikatem:

Drive types (double, single-sided) or
diskette types not compatible

Przykłady:

Rozkaz DISKCOMP o następującej postaci:

A>DISKCOMP A: B:

spowoduje porównanie zawartości dwóch dyskietek założonych na urządzenia A: i B:.

Rozkaz DISKCOMP o następującej postaci:

A>DISKCOMP

spowoduje porównanie zawartości dwóch dyskietek zakładanych kolejno na urządzenie A: (urządzenie robocze), przy czym wymiana obu nośników będzie sterowana odpowiednimi komunikatami wyświetlanymi na ekranie monitora.

Rozkaz DISKCOMP o następującej postaci:

A>DISKCOMP B:

spowoduje porównanie zawartości dwóch dyskietek: pierwszej założonej na urządzenie B: (dyskietka źródłowa) i drugiej, założonej na urządzenie robocze A:.

6.16. Rozkaz DISKCOPY (Copy Diskettes Only).

Format:

[d:][ścieżka]DISKCOPY [d1:[d2:]][/1]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz DISKCOPY pozwala na kopiowanie zawartości jednej dyskietki na drugą, przy czym dyskietka wyjściowa może, jeśli to niezbędne, zostać zaformatowana przed wykonaniem na niej kopii.

Poszczególne parametry występujące w linii komendy rozkazu DISKCOPY mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym DISKCOPY, określa miejsce, w którym znajduje się program obsługi rozkazu DISKCOPY,

[d1:] określa stację dyskietkową, na której znajduje się dyskietka źródłowa,

[d2:] określa stację dyskietkową, na której zostanie wykonana kopia dyskietki źródłowej,

klucz /1 (nie występuje w wersji MS DOS 3.10) wymusza kopiowanie tylko pierwszej strony dyskietki źródłowej, bez względu na typ stacji dyskietkowej.

Uwagi dotyczące posługiwania się rozkazem DISKCOPY:

1. Rozkaz DISKCOPY wykonuje kopiowanie całej dyskietki źródłowej sektor po sektorze, a nie jak rozkazy COPY lub XCOPY zbiór po zbiorze. Oznacza to, że kopia dyskietki wykonana tym rozkazem jest identyczna, co do zawartości i położenia poszczególnych danych na nośniku, jak dyskietka źródłowa. Jeśli dyskietka źródłowa była uprzednio wielokrotnie wykorzystywana przy modyfikowaniu, kasowaniu, zapisywaniu etc. informacji, to na skutek przyjętego przez system operacyjny PC DOS sposobu wykonywania

zapisu, pewna jej część, zarezerwowana dla określonych zbiorów, pozostaje faktycznie niewykorzystana. W takim przypadku wskazane jest używanie rozkazów COPY i XCOPY, które w przeciwieństwie do rozkazu DISKCOPY, pozwalają eliminować niewykorzystane, ale zarezerwowane fragmenty nośnika dyskiety.

2. Rozkazu DISKCOPY można używać tylko do dyskietek. Próba wyspecyfikowania w linii komendy rozkazu DISKCOPY stacji dysków twardych lub dysku wirtualnego (patrz instrukcja DEVICE w rozdziale pt. 'KONFIGUROWANIE SYSTEMU OPERACYJNEGO'), kończy się komunikatem o wykryciu błędu.
3. W linii komendy rozkazu DISKCOPY można wyspecyfikować oba jednakowe parametry 'd1:' i 'd2:'. W takim przypadku program obsługi rozkazu DISKCOPY odpowiednimi komunikatami, wyświetlanymi na ekranie monitora, będzie sterował wymianą dyskietki źródłowej i wyjściowej na wskazanym urządzeniu. Użytkownik może również pominąć oba parametry 'd1:' i 'd2:' - wówczas kopiowanie wykonywane będzie na urządzeniu roboczym. Jeśli pominięty zostanie parametr 'd2:', to system operacyjny PC DOS domyślnie przyjmie zamiast niego urządzenie robocze.
4. Zakończenie kopiowania system operacyjny sygnalizuje następującym komunikatem:

Copy another (Y/N)?_

Jeśli użytkownik naciśnie klawisz z literą 'Y', to będzie mógł kopiować następną dyskietkę korzystając z tych samych stacji dyskietkowych co poprzednio, przy czym program obsługi rozkazu DISKCOPY będzie sterował wymianą dyskietek. Wciśnięcie klawisza z literą 'N' kończy wykonywanie rozkazu DISKCOPY.

5. Podczas wykonywania rozkazu DISKCOPY może wystąpić błąd odczytu zawartości dyskietki źródłowej. Powoduje to wyświetlenie komunikatu gdzie i taki błąd wystąpił, po czym kopiowanie jest kontynuowane. Należy mieć świadomość, że w takim przypadku wykonana kopia może być bezwartościowa.
6. Program obsługi rozkazu DISKCOPY automatycznie określa numer strony, ścieżki i sektora na ścieżce, który będzie następnie kopiowany:
 - jeśli dyskietka źródłowa została zaformatowana jednostronnie, to rozkaz DISKCOPY kopiuje tylko pierwszą stronę,

- jeśli stacja dyskietkowa i założona na nią dyskietka źródłowa umożliwiają dwustronny zapis, to rozkaz DISKCOPY wykona kopiowanie dwustronne, chyba że wyspecyfikowano klucz /1. W przypadku, gdy stacja z dyskietką wyjściową umożliwia tylko zapis jednostronny, to program obsługi rozkazu DISKCOPY wygeneruje odpowiedni komunikat o błędzie i zakończy kopiowanie,
 - w przypadku, gdy dyskietka źródłowa została zaformatowana z 9 sektorami na ścieżce, to jej kopia także zostanie wykonana w tym formacie.
7. Dyskietki dwustronne, zapisane w stacjach dyskietkowych umożliwiających dużą gęstość zapisu, mogą nie dać się odczytać w stacjach innego typu.
 8. Sprawdzenie poprawności wykonanej kopii można wykonać rozkazem DISKCOMP.
 9. Rozkaz DISKCOPY ignoruje przyporządkowania dokonane wcześniej rozkazami ASSIGN, JOIN i SUBST.

U W A G A :

Rozkaz DISKCOPY może generować następujący komunikat:

Insert disk with \COMMAND.COM in drive A
and strike any key when ready

Oznacza to żądanie założenia na urządzenie A: dyskietki z systemem operacyjnym i potwierdzenie wciśnięciem klawisza dowolnej litery.

Możliwości wykonywania rozkazu DISKCOPY
w zależności od zestawu sprzętowego:

1. Jeśli mikrokomputer posiada stacje dyskietkowe jednostronne, to rozkaz DISKCOPY pozwala kopiować tylko dyskietki jednostronne o pojemności 160/180kB.
2. Jeśli mikrokomputer posiada stacje dyskietkowe dwustronne, to rozkaz DISKCOPY pozwala na kopiowanie:
 - dyskietki jednostronnej o pojemności 160/180kB na dyskietkę jednostronną o pojemności 160/180kB,

- dyskietki dwustronnej o pojemności 720kB na dyskietkę dwustronną o pojemności 720kB (tylko w wersji PC DOS 3.20).

3. Jeśli komputer posiada stacje dyskietkowe dwustronne o dużej gęstości zapisu to rozkaz DISKCOPY pozwala na kopiowanie:

- dyskietki jednostronnej o pojemności 160/180kB na dyskietkę jednostronną o pojemności 160/180kB (*),
- dyskietki dwustronnej o pojemności 320/360kB na dyskietkę dwustronną o pojemności 320/360kB (*),
- dyskietki o dużej gęstości zapisu o pojemności 1.2MB na dyskietkę o dużej gęstości zapisu o pojemności 1.2MB (*).

(*) - oznacza, że odczyt wymienionych typów dyskietek w stacjach dyskietkowych umożliwiających dużą gęstość zapisu, nie zawsze wykonywany jest poprawnie.

Rozkazu DISKCOPY nie można używać do kopiowania dyskietek założonych na dwie stacje o różnych kombinacjach niż wymienione powyżej. W takim przypadku wykonywanie rozkazu DISKCOPY zawsze zostaje zakończone następującym komunikatem:

Drive types or
diskette types not compatible

Dla kopiowania zawartości dyskietki dwustronnej na dyskietkę dwustronną o dużej gęstości zapisu można użyć rozkazu COPY *.*.

Przykłady:

Rozkaz DISKCOPY o następującej postaci:

A>DISKCOPY A: B:

spowoduje skopiowanie zawartości dyskietki źródłowej założonej na urządzenie A: na dyskietkę wyjściową, założoną na urządzenie B:.

Rozkaz DISKCOPY o następującej postaci:

A>DISKCOPY

spowoduje skopiowanie zawartości dyskietki źródłowej na dyskietkę wyjściową, które będą zakładane kolejno na urządzenie A: (urządzenie robocze), przy czym wymiana obu nośników będzie sterowana odpowiednimi komunikatami wyświetlanymi na ekranie monitora.

6.17. Rozkaz ERASE.

Format:

ERASE [d:][ścieżka]nazwa[.rozszerzenie]

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz ERASE usuwa zawartość wyspecyfikowanego zbioru (zbiórów), jeśli nie posiada on ustawionego atrybutu RO (ang. Read Only – patrz opis rozkazu ATTRIB).

Poszczególne parametry występujące w linii komendy rozkazu ERASE mają następujące znaczenie:

[d:] określa urządzenie, na którym znajduje się nośnik ze zbiorem przeznaczonym do usunięcia. Jest to parametr opcjonalny i w razie jego pominięcia system operacyjny PC DOS domyślnie przyjmie urządzenie robocze,

[ścieżka] określa katalog ze zbiorem przeznaczonym do usunięcia. Jest to także parametr opcjonalny i w razie jego pominięcia system operacyjny PC DOS domyślnie przyjmie katalog roboczy,

nazwa[.rozszerzenie] określają nazwę i rozszerzenie zbioru przeznaczonych do usunięcia. Można tutaj używać znaków specjalnych '?' i '*'.

Uwagi dotyczące posługiwania się rozkazem ERASE:

1. Rozkaz ERASE pozwala na usunięcie zarówno pojedynczego zbioru jak i całej grupy zbiorów. Jeśli zamiast nazwy i rozszerzenia wyspecyfikowano *,*, sygnalizując zamiar usunięcia wszystkich zbiorów zapisanych w katalogu, to system operacyjny PC DOS zażąda dodatkowego potwierdzenia wyświetlając następujące pytanie:

Are you sure (Y/N)?

2. Rozkazu ERASE nie można używać do usuwania katalogów podrzędnych. Do tego celu służy rozkaz RMDIR (patrz opis).
3. Rozkazem ERASE nie można usunąć 'ukrytych' zbiorów systemowych IBMBIO.COM i IBMDOS.COM.
4. Należy zwrócić szczególną uwagę podczas wykonywania kasowania po zainicjowaniu rozkazów ASSIGN, JOIN i SUBST.

Kasowanie zbioru nie polega na skasowaniu jego zapisu z dysku lub dyskietki, lecz na odpowiednich zmianach w tzw. File Allocation Table, zwalniających miejsca zarezerwowane na ten zbiór. Zbiór taki może zostać odtworzony, o ile nie zapisano w tym samym miejscu innych informacji lub nie zaformatowano dysku lub dyskietki. Programy odtwarzające skasowane zbiory nie są opisane w tej instrukcji.

Przykłady:

Rozkaz ERASE o następującej postaci:

ERASE A:FIL.DAT

spowoduje usunięcie zbioru FIL.DAT zapisanego na dyskietce założonej na urządzenie A: w katalogu roboczym.

Rozkaz ERASE o następującej postaci:

ERASE C:*.BAS

spowoduje usunięcie wszystkich zbiorów o rozszerzeniu .BAS zapisanych na dysku C: w katalogu roboczym.

Rozkaz ERASE o następującej postaci:

ERASE C:\LEVEL1\LEVEL2

spowoduje usunięcie wszystkich zbiorów zapisanych na dysku C:, w katalogu podrzędnym LEVEL2.

6.18: Rozkaz EXE2BIN.

Format:

[d:][ścieżka]EXE2BIN [d1:][ścieżka1]nazwa1[.rozszerzenie1]
[d2:][ścieżka2][nazwa2[.rozszerzenie2]]

czyli:

[d:][ścieżka]EXE2BIN (zbior wejściowy) (zbior wyjściowy)

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz EXE2BIN pozwala wykonać konwersję zbioru binarnego o rozszerzeniu .EXE (zbioru będącego zbiorem wynikowym programu konsolidatora LINK) na zbiór o rozszerzeniu .COM lub .BIN.

Poszczególne parametry występujące w linii komendy rozkazu EXE2BIN mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym EXE2BIN, określa miejsce, w którym znajduje się program obsługi rozkazu EXE2BIN,

[d1:][ścieżka1]nazwa1[.rozszerzenie1] określa zbiór źródłowy. Składnia linii komendy rozkazu EXE2BIN dopuszcza możliwość pominięcia parametrów 'd1:', 'ścieżka1' i '.rozszerzenie1'. Wówczas:

- jeśli nie określono parametru 'd1:', to domyślnie zostanie przyjęte urządzenie robocze,
- jeśli nie określono parametru 'ścieżka1', to domyślnie zostanie przyjęty katalog roboczy,
- jeśli nie określono parametru '.rozszerzenie1', to

domyślnie zostanie przyjęte rozszerzenie .EXE.

[d2:][ścieżka2][nazwa2[.rozszerzenie2]] określa zbiór wyjściowy (wynikowy). Składnia linii komendy rozkazu EXE2BIN dopuszcza możliwość pominięcia parametrów 'd2:', 'ścieżka2', 'nazwa2' i '.rozszerzenie2'. Wówczas:

- jeśli pominięto parametr 'd2:', to domyślnie zostanie przyjęte urządzenie, na którym znajduje się zbiór źródłowy,
- jeśli pominięto parametr 'ścieżka2', to domyślnie zostanie przyjęty katalog roboczy urządzenia, na którym ma zostać zapisany zbiór wynikowy,
- jeśli pominięto parametr 'nazwa2', to domyślnie zostanie przyjęta nazwa zbioru źródłowego,
- jeśli pominięto parametr 'rozszerzenie2', to domyślnie zostanie przyjęte rozszerzenie .BIN.

Uwagi dotyczące posługiwania się rozkazem EXE2BIN:

1. Zbiór źródłowy może mieć inne rozszerzenie niż .EXE, ale koniecznie musi mieć format zbioru wynikowego, utworzonego przez program konsolidatora LINK. Dodatkowo jego długość obejmująca kod programu i dane nie może przekraczać 64kB. Ponadto w programie źródłowym nie może być zdefiniowany segment stosu STACK SEGMENT.
2. W zależności od określenia w zbiorze źródłowym wartości początkowych pary rejestrów: CS:IP, program obsługi rozkazu EXE2BIN może wykonywać dwa rodzaje konwersji zbioru źródłowego .EXE w zbiór wynikowy .BIN:
 - jeśli w programie źródłowym nie określono wartości początkowych pary rejestrów CS:IP (a zatem w zbiorze .EXE ich wartości zostają ustalone na 0000:0000H), to zostanie wykonana bezpośrednia konwersja binarna. Jeśli jest wymagane określenie, od jakiego adresu ma rozpoczynać się segment programu (np. gdy program zawiera instrukcje wymagające relokowania segmentu), to system operacyjny PC DOS zażąda podania wartości tego adresu. W przypadku tego typu konwersji zbiór wynikowy .BIN będzie użyteczny tylko wtedy, gdy zostanie załadowany do pamięci operacyjnej od adresu określonego bezpośrednio przez użytkownika (system operacyjny PC DOS nie jest zdolny samodzielnie wyko-

nać poprawnego załadowania tego zbioru do pamięci),

- jeśli w programie źródłowym wartości początkowe pary rejestrów: CS:IP ustalono na 0000:0100H, to system operacyjny traktuje ten program tak; jakby wystąpiła w nim dyrektywa asemblera ASMB6 o nazwie ORG 100H. Nie jest zatem konieczne określanie adresu, od którego rozpoczyna się segment kodu, a zbiór wynikowy .BIN można przekształcić w zbiór typu .COM przez zwykłą zmianę rozszerzenia z .BIN na .COM, wykonywaną rozkazem RENAME.

3. Aby w wyniku działania rozkazu EXE2BIN bezpośrednio uzyskać zbiór wynikowy typu .COM, w programie źródłowym musi być umieszczona dyrektywa ORG 100H oraz musi zostać określony adres, od którego rozpoczyna się segment kodu, tak jak pokazano poniżej:

```
ORG 100H
START:
.
.
.
END START
```

Przykład:

Rozkaz EXE2BIN o następującej postaci:

EXE2BIN A:ABC.EXE B:

spowoduje wykonanie konwersji zbioru źródłowego ABC.EXE na zbiór binarny ABC.BIN, który zostanie zapisany w katalogu roboczym na urządzeniu B:.

6.19. Rozkaz FDISK (nie występuje w wersji MS DOS 3.10).

Format:

[d:][ścieżka]FDISK

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz FDISK jest specjalnym programem, używanym do organizacji dysku twardego. Dysk twardy może zawierać od jednego do czterech systemów operacyjnych. Każdy z tych systemów musi być zapisany w oddzielnym obszarze dysku stałego, zwanym partycją. Każdy system operacyjny, inicjowany z dysku twardego, powinien zawierać program umożliwiający tworzenie i usuwanie partycji z dysku stałego.

Program FDISK zgłasza się wyświetleniem następujących możliwości opcji:

1. Create DOS partition

Tworzenie partycji DOSa o określonej wielkości. Tylko jedną partycję można przeznaczyć dla systemu operacyjnego PC DOS. Jeśli używany jest tylko system operacyjny PC DOS, to cały dysk powinien być przeznaczony tylko dla niego.

2. Change active partition

Zmiana aktywnej partycji w celu użycia innego systemu operacyjnego. Funkcja determinuje, która partycja będzie używana i, który system operacyjny będzie restartowany podczas inicjacji wykonywanej z dysku twardego.

3. Delete DOS partition

Usuwanie aktualnej partycji DOSa. Dane znajdujące się w tej partycji zostają usunięte. Funkcja jest używana do zmiany wielkości partycji lub, jeśli segment jest uszkodzony, do odtworzenia go z kopii zapasowej, utworzonej rozkazem BACKUP.

4. Display partition data

Wyświetlanie informacji o dysku twardym, takich jak: status aktywności, typ systemu operacyjnego przydzielonego dla każdej partycji, początek i koniec obszaru (w cylindrach) i wielkość każdej partycji.

5. Select next fixed disk drive

Wybieranie następnego dysku stałego do podziału. Wybór tej opcji i algorytmy postępowania opisano dokładnie w rozdziale pt. 'Przygotowanie dysku twardego do pracy z systemem operacyjnym PC DOS'.

Przykład:

Rozkaz FDISK o następującej postaci:

A:FDISK

inicjuje wykonanie programu FDISK znajdującego się na urządzeniu A:

6.20. Rozkaz FIND (FIND Filter).

Format:

```
[d:][ścieżka]FIND [/V][C][N]"string"  
[[di:][ścieżka][nazwa][rozszerzenie]...]
```

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz (filtr) FIND pozwala na przesłanie na standardowe urządzenie wyjściowe (domyślnie monitor konsoli) wszystkich tych linii z wyspecyfikowanych zbiorów, które zawierają (lub nie zawierają, jeśli użyto klucza /V) określony ciąg znaków.

Poszczególne parametry występujące w linii komendy rozkazu FIND mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym FIND, określa miejsce, w którym znajduje się program obsługi rozkazu FIND,

[[di:][ścieżka][nazwa][rozszerzenie]...] określa i-ty kolejny zbiór, który będzie przeglądany w poszukiwaniu linii zawierającej wyspecyfikowany ciąg znaków,

"string" określa poszukiwany ciąg znaków. Ciąg ten musi być wyspecyfikowany w 'pojedynczych' lub "podwójnych" apostrofach. Należy zaznaczyć, że program obsługi rozkazu FIND rozróżnia duże i małe litery,

klucz /V wymusza listowanie wszystkich linii nie zawierających wyspecyfikowanego ciągu znaków,

klucz /C powoduje wyświetlenie informacji ile razy pojawił się określony ciąg znaków w wyspecyfikowanych zbiorach. Jeśli rozkaz FIND zainicjowano z kluczami /C i /N, to klucz /N jest ignorowany,

klucz /N wymusza wyświetlenie przed linią jej numeru względem początku zbioru.

Uwagi dotyczące posługiwania się rozkazem FIND:

1. W specyfikacji linii komendy rozkazu FIND, dla określenia nazw i rozszerzeń poszczególnych zbiorów, nie wolno używać znaków specjalnych '?' i '*'.
2. Każdy zbiór jest przeszukiwany tylko do momentu napotkania pierwszego znaku Ctrl/Z (1AH), oznaczającego koniec zbioru tekstowego (nie musi wcale to być koniec zbioru, szczególnie w zbiorach binarnych).

Przykłady:

Rozkaz FIND o następującej postaci:

```
A>FIND "Fool's Parad" BOOK1.TXT BOOK2.TXT BOOK3.TXT
```

spowoduje wyświetlenie na ekranie monitora wszystkich linii ze zbiorów: BOOK1.TXT, BOOK2.TXT i BOOK3.TXT które zawierają ciąg znaków: "Fool's Parad".

Rozkaz FIND o następującej postaci:

```
A>DIR B: | FIND /V "DAT"
```

spowoduje wyświetlenie na ekranie monitora nazw wszystkich zbiorów z katalogu roboczego na urządzeniu B:, które nie zawierają ciągu znaków "DAT".

Rozkaz FIND o następującej postaci:

```
A>DIR : FIND "<DIR>"
```

spowoduje wylistowanie wszystkich katalogów zapisanych w katalogu roboczym na urządzeniu A:. Należy zwrócić uwagę, że dwa ostatnie przykłady realizują tzw. przetwarzanie potokowe (ang. piping).

Niech zbiór PROG.TXT składa się z następującego tekstu:

```
This is a  
beautiful  
day for a picnic.
```

Rozkaz FIND o następującej postaci:

```
A>FIND /V"f"PROG.TXT
```

spowoduje wylistowanie wszystkich linii nie zawierających ciągu znaków "f":

```
-----A:PROG.TXT  
This is a
```

Rozkaz FIND o następującej postaci:

```
A>FIND /N"f"PROG.TXT
```

spowoduje wylistowanie wszystkich linii ze zbioru PROG.TXT, zawierających ciąg znaków "f", wraz z ich numeracją względem początku zbioru:

```
-----A:PROG.TXT  
[2]beautiful  
[3]day for a picnic.
```

Rozkaz FIND o następującej postaci:

```
A>FIND /C" a"A:PROG.TXT
```

spowoduje wyświetlenie informacji, ile razy ciąg znaków " a" wystąpił w zbiorze PROG.TXT:

```
-----A:PROG.TXT:2
```

6.21. Rozkaz FORMAT.

Format:

[d:][ścieżka]FORMAT [d1:][S][I][L][B][V][B][4]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz FORMAT pozwala na przygotowanie nowego dysku lub dyskietki do pracy z systemem operacyjnym PC DOS. Polega to na odpowiednim zaformatowaniu nośnika; kontroli powierzchni dysku lub dyskietki dla ustalenia uszkodzeń mechanicznych ścieżek, oraz założeniu katalogu podstawowego (ang. root directory) i tablicy FAT, ustalającej położenie poszczególnych zbiorów (ang. File Allocatin Table).

U W A G Ą :

Formatowanie dysku lub dyskietki bezpowrotnie niszczy wszystkie dotychczas zapisane informacje, dlatego musi być wykonywane ze szczególną ostrożnością, szczególnie w odniesieniu do dysków twardych. W przypadku dysku twardego ulega skasowaniu zawartość całej, poprzednio utworzonej partycji DOSa wraz z katalogami i ich zawartością. Program obsługi rozkazu FORMAT nie ma wbudowanego mechanizmu kontroli, jakie informacje ulegają zatarciu podczas formatowania.

Poszczególne parametry występujące w linii komendy rozkazu FORMAT mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym FORMAT, określa miejsce, gdzie znajduje się program obsługi rozkazu FORMAT,

[d1:] określa urządzenie na którym znajduje się dysk lub dyskietka przeznaczona do zaformatowania. Parametr ten jest zdeteminowany konfiguracją sprzętowa i rodzajem formatowanych dyskietek (jednostronne, dwustronne, o dużej pojemności etc.),

klucz /S wymusza po zakończeniu formatowania kopiowanie zbiorów systemowych IBMBIO.COM, IBMDOS.COM i COMMAND.COM z dyskietki systemowej w urządzeniu roboczym na formatowany dysk. Jeśli dyskietki z systemem operacyjnym nie założono na urządzenie robocze, lub urządzenie robocze jest dyskiem twardym, wówczas system operacyjny żąda założenia tej dyskietki na stację A:

klucz /I (nie występuje w wersji MS DOS 3.10) wymusza jednostronne formatowanie dyskietek, bez względu na typ stacji dyskietkowej,

klucz /B (nie występuje w wersji MS DOS 3.10) formatuje dyskietkę z 8 sektorami na ścieżce (format wymagany w wersji PC DOS 1.10). Domyślnie dyskietki są formatowane z 9/15 sektorami na ścieżce,

klucz /V umożliwia wprowadzenie etykiety identyfikującej dysk lub dyskietkę,

klucz /B (nie występuje w wersji MS DOS 3.10) wymusza formatowanie z 8 sektorami na ścieżce i z zarezerwowaniem miejsca na 'ukryte' zbiory systemowe IBMBIO.COM i IBMDOS.COM. Zbiory te oraz interpreter komend nie zostają jednak przepisane na formatowaną dyskietkę. Zastosowanie klucza /B pozwala na wpisanie, przy pomocy rozkazu SYS, każdej wersji systemu operacyjnego PC DOS, podczas gdy formatowanie bez klucza /B pozwala zapisać tylko wersję 3.20.

klucz /4 umożliwia dwustronne formatowanie dyskietek o pojemności 320/360kB w stacjach dyskietkowych o dużej gęstości zapisu (1.2MB). Dyskietki zaformatowane w tych stacjach z kluczem /4 nie zawsze dają się poprawnie zapisywać i odczytywać na stacjach o mniejszej gęstości zapisu.

Poniższa tabela przedstawia dopuszczalne klucze rozkazu FORMAT w odniesieniu do poszczególnych typów urządzeń:

Typ dysku/dyskietki	Dopuszczalne klucze
160/180kB	/S, /V, /1, /8, /B, /4
320/360kB	/S, /V, /1, /8, /B, /4
720kB	/S, /V
1.2MB	/S, /V
dysk twardy	/S, /V

Uwagi dotyczące posługiwania się rozkazem FORMAT:

1. Każda nowa dyskietka lub dysk, przed pierwszym użyciem pod systemem operacyjnym PC DOS, musi zostać zaformatowana.
2. Każdy dysk twardy musi zostać powtórnie zaformatowany, jeśli przy pomocy rozkazu FDISK będą zmieniane rozmiary partycji DOSa.
3. Formatowanie bezpowrotnie niszczy wszelkie informacje dotychczas zapisane na formatowanym nośniku.
4. Podczas formatowania wszystkie uszkodzone mechanicznie ścieżki zostają automatycznie oznaczone, aby zapobiec zapisywaniu na nich jakichkolwiek informacji.
5. Zbiory systemowe IBMBIO.COM i IBMDOS.COM są zbiorami ukrytymi, a więc nie są zapisane w żadnym katalogu i dla użytkownika pozostają niewidoczne.
6. Podczas formatowania dysków lub dyskietek z kluczem /V, system operacyjny żąda podania etykiety. Etykieta może składać się z ciągu od 1 do 11 znaków, przy czym dozwolone jest używanie wszystkich tych znaków, które mogą tworzyć nazwę zbioru. Dysk lub dyskietkę można oznaczyć etykietą, także po wykonaniu formatowania, wykorzystując rozkaz LABEL.
7. Po zakończeniu formatowania program obsługi rozkazu FORMAT wyświetla raport zawierający informacje o:
 - miejscach oznaczonych jako miejsca uszkodzone mechanicznie,
 - ilości miejsca przeznaczonego na zbiory systemowe (o ile rozkaz FORMAT został zainicjowany z kluczem /S),
 - ilości miejsca przeznaczonym na zbiory użytkownika.

8. Rozkaz FORMAT determinuje, na które urządzenia może zostać założona dyskietka przeznaczona do zaformatowania. Dyskietki formatowane jednostronnie z 8 sektorami na ścieżce można przygotowywać w dowolnej stacji dyskietkowej. Natomiast formatowanie, a następnie wykorzystywanie dyskietek o zapisie dwustronnym, nie może odbywać się w stacjach umożliwiających tylko pracę jednostronną. Poniższa tabela pokazuje jakie dyskietki mogą być używane w poszczególnych typach stacji dyskietkowych:

Typ stacji dyskietkowej	Typ dyskietki
160/180kB 320/360kB	160/180kB z zapisem jednostronnym 320/360kB z zapisem jednostronnym lub dwustronnym
720kB 1.2MB	720kB z zapisem dwustronnym 160/180kB z zapisem jednostronnym 320/360kB z zapisem dwustronnym dyskietki o dużej gęstości zapisu

9. Dyski twarde są fabrycznie zaformatowane przez producenta. Inicjując wykonanie rozkazu FORMAT dla dysku twardego faktycznie nie wykonuje się formatowania, lecz tylko kontrolę partycji DOSa.
10. Inicjując wykonanie rozkazu FORMAT z kluczem /S należy pamiętać, że system operacyjny może kopiowanie zbiorów systemowych wykonywać partiami. Ma to miejsce wtedy, gdy w danej chwili rozmiar pamięci operacyjnej jest zbyt mały dla załadowania do niej zbiorów w całości, w celu ich późniejszego zapisu. W takim przypadku przedwczesne usunięcie dyskietki źródłowej ze stacji zawsze wywoła reakcję systemu operacyjnego PC DOS, żądającego jej powtórne założenia.
11. Rozkaz FORMAT nie może być inicjowany jednocześnie z kluczami /S i /B oraz /V i /B.
12. Jeśli wykonanie rozkazu FORMAT zainicjowano z kluczem /S to zbiory systemowe są kopiowane z urządzenia roboczego. Jeśli urządzeniem roboczym jest dysk twardy, nie zawierający zbiorów systemowych to system zażąda założenia na stację A: dyskietki z tymi zbiorami.
13. Rozkaz FORMAT ignoruje przyporządkowania ustawione rozkazami: ASSIGN, JOIN i SUBST.

14. Rozkaz FORMAT ustawia następujące wartości parametru ERRORLEVEL instrukcji warunkowej IF (przetwarzanie warunkowe):

- 0 - formatowanie zakończono pomyślnie,
- 1 - nie zdefiniowano znaczenia,
- 2 - nie zdefiniowano znaczenia,
- 3 - wykonywanie rozkazu FORMAT zostało przerwane przez użytkownika wciśnięciem kombinacji klawiszy 'Ctrl/Break',
- 4 - wykonywanie rozkazu FORMAT zostało przerwane jakimś błędem,
- 5 - wykonywanie formatowania dysku twardego zostało przerwane odpowiedzią 'N', udzieloną na zadane przez system pytanie.

Kompatybilność zapisu na dyskietkach formatowanych pod różnymi wersjami systemu operacyjnego PC DOS:

Aby uniknąć sytuacji spowodowanych niemożnością wykorzystywania dyskietek zaformatowanych w wersji 3.20 do pracy pod systemem w wersji 1.10 należy zwrócić uwagę, że formatowanie z kluczami /B lub /8 pozwala na pracę dyskietek z wersją 1.10. W tym przypadku należy jednak unikać formatowania z kluczem /V, gdyż wersja 1.10 nie etykietuje nośników. W przypadku zainicjowania rozkazu FORMAT z kluczem /S należy pamiętać, że w wersji 1.10 istniała możliwość kopiowania tylko tej samej (1.10) wersji systemu operacyjnego. Toteż inicjując wykonanie rozkazu FORMAT w wersji 3.20 z kombinacją klawiszy /B /V /S, należy być świadomym, że jest to niedozwolona kombinacja dla formatowania dyskietek przeznaczonych do pracy z wersją 1.10.

Przykład:

Dla zaformatowania, oznaczenia etykieta i skopiowania zbiorów systemowych na dyskietce założonej na urządzenie B:, zainicjowano wykonanie następującego rozkazu FORMAT (urządzeniem roboczym jest stacja A:):

A>FORMAT B:/S/V

System odpowiada następującym komunikatem:

Insert new diskette for drive B:
and strike ENTER when ready

Po założeniu dyskietki i wciśnięciu klawisza 'Enter' generuje następujący komunikat:

Head h Cylinder c

po zakończeniu formatowania:

Format complete
System transferred

Volume label (11 characters, ENTER for none)? MYDISK

xxxxxx bytes total disk space
xxxxxx bytes used by system
xxxxxx bytes available on disk

Format another (Y/N)? N

W powyższym przykładzie dyskietkę oznaczoną etykietą MYDISK. Ostatnie pytanie dotyczy możliwości zaformatowania kolejnej dyskietki. Podczas formatowania dysku twardego, zamiast ponaglenia dotyczącego założenia dyskietki, system wyświetla następujący komunikat:

Enter current Volume Label for Drive
(Press enter for none):

WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE x: WILL BE LOST

Proceed with Format (Y/N)?

Gdzie litera x oznacza urządzenie wyspecyfikowane w linii komendy rozkazu FORMAT.

Ze względu na rozmiar nośnika formatowanie dysku twardego trwa kilka minut i kończy się takim samym zapytaniem o etykietę, jak w przypadku dyskietki.

- 6.22. Rozkaz GRAFTABL (Load Graphics Table, nie występuje w wersji MS DOS 3.10).

Format:

[d:][ścieżka]GRAFTABL

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz GRAFTABL umożliwia załadowanie do pamięci operacyjnej tablicy zawierającej definicję dodatkowych znaków używanych przez adapter koloru/grafiki. Parametry 'd:' i 'ścieżka', przed słowem kluczowym GRAFTAB, określają miejsce, gdzie znajduje się program obsługi rozkazu GRAFTABL.

Rozkaz GRAFTABL pozwala wyświetlać na ekranie monitora znaki obcojęzyczne lub znaki rozszerzonego kodu ASCII, o ile użytkownik posługuje się graficznym trybem wyświetlania. Użycie tego rozkazu zwiększa obszar pamięci zajmowanej przez część rezydentną systemu operacyjnego PC DOS.

Uwagi dotyczące posługiwania się rozkazem GRAFTABL:

1. Prawidłowość załadowania do pamięci operacyjnej wektora przerwań oraz tablicy zawierającej definicje dodatkowych znaków, rozkaz GRAFTABL sygnalizuje następującym komunikatem:

GRAPHIC CHARACTERS LOADED

2. Rozkaz GRAFTABL powinien zostać wykonany tylko raz, po restarcie systemu operacyjnego. Jeśli użytkownik próbuje zainicjować go powtórnie, to na ekranie monitora pojawia się następujący komunikat:

GRAPHICS CHARACTERS ALREADY LOADED

Przykład:

Rozkaz GRAFTABL o następującej postaci:

A>GRAFTABL

spowoduje załadowanie do pamięci operacyjnej znaków rozszerzonego kodu ASCII. Znaki te są dostępne jedynie w graficznym trybie pracy monitora.

- 6.23. Rozkaz GRAPHICS (Screen Print, nie występuje w wersji MS DOS 3.10).

Format:

[d:][ścieżka]GRAPHICS [printer type][R][B]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz GRAPHICS umożliwia wydruk zawartości ekranu graficznego na drukarce mogącej pracować w trybie graficznym, o ile mikrokomputer IBM PC został wyposażony w adapter koloru/grafiki.

Poszczególne parametry występujące w linii komendy rozkazu GRAPHICS mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym GRAPHICS, określa miejsce, gdzie znajduje się program obsługi rozkazu GRAPHICS,

[printer type] określa typ drukarki. Parametr ten może przybierać jedną z następujących wartości:

COLOR1 - oznacza IBM PC Color Printer (drukarkę umożliwiającą kolorowy wydruk, ale z założoną czarną taśmą barwiącą),

- COLOR4 - oznacza IBM PC Color Printer z założoną taśmą barwiącą typu RGB, umożliwiającą wydruk 4 kolorów: czerwonego, zielonego, niebieskiego i czarnego (ang. Red, Green, Blue, Black - RGB),
- COLOR8 - oznacza IBM PC Color Printer z założoną taśmą barwiącą typu CMY, umożliwiającą wydruk 4 kolorów: niebiesko-zielonego, karmazynowego, żółtego i czarnego (ang. Cyan, Magenta, Yellow, Black - CMY),
- COMPACT - oznacza IBM PC Compact Printer,
- GRAPHICS - oznacza IBM PC Graphics Printer.

Jeśli parametr 'printer type' nie zostanie określony, to system operacyjny PC DOS domyślnie przyjmie wartość GRAPHICS.

klucz /R wymusza wydruk koloru czarnego - jako czarnego i białego - jako białego (tak jak to jest wyświetlane na ekranie monitora). Jeśli linia komendy rozkazu GRAPHICS nie zawiera klucza /R, to system operacyjny PC DOS domyślnie będzie drukował kolor czarny jako biały, a biały - jako czarny (tak jak w 'zwykłym' wydruku),

klucz /B wymusza wydruk koloru tła. Parametr ten jest używany tylko dla drukarek typu COLOR4 i COLOR8 (patrz parametr 'printer type' i jeśli nie został wyspecyfikowany, to kolor tła nie będzie drukowany).

Uwagi dotyczące posługiwania się rozkazem GRAPHICS:

1. Zainicjowanie rozkazu GRAPHICS powoduje zwiększenie obszaru pamięci zajmowanego przez część rezydentną systemu operacyjnego PC DOS.
2. Aby wydrukować aktualną zawartość ekranu należy wcisnąć kombinację klawiszy 'Shift/PrtSc':
 - jeśli ekran monitora znajduje się w znakowym trybie pracy, to wydruk trwa do 30 sekund,
 - jeśli ekran monitora znajduje się w graficznym trybie pracy, to wydruk jego zawartości może trwać dłużej niż 3 minuty, a postać samego wydruku zależy od rodzaju monitora i typu drukarki:

- dla monitora kolorowego o rozdzielczości 320*200 punktów i drukarki typu GRAPHICS lub COLOR1 wydruk zawartości ekranu trwa do chwili uzyskania czterech odcieni szarości,
- dla monitora kolorowego o rozdzielczości 640*200 punktów wydruk jest odwrócony o 90 stopni zgodnie z ruchem wskazówek zegara w stosunku do obrazu na ekranie.

3. Aby wywołać funkcje drukowania zawartości ekranu z programu napisanego w języku assemblera ASM86, należy użyć następującego ciągu instrukcji:

```
PUSH BP
INT 5
POP BP
```

Przykład:

Rozkaz GRAPHICS o następującej postaci:

```
A>GRAPHICS COLORB/B/R
```

spowoduje, że po wciśnięciu kombinacji klawiszy 'Shift/PrtSc', na drukarce typu 'COLORB' z założoną taśmą barwiącą CMY zostanie wydrukowana zawartość ekranu monitora z zachowaniem barwy białej i czarnej, tak jak na ekranie monitora (kolor biały jako biały, a czarny jako czarny).

6.24. Rozkaz JOIN.

Format:

```
[d:][ścieżka]JOIN
```

lub:

```
[d:][ścieżka]JOIN d1: d2:\directory
```

lub:

[d:][ścieżka]JOIN d1:/D

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz JOIN pozwala na utworzenie pojedynczej struktury katalogowej z dwóch struktur, istniejących na dwóch różnych urządzeniach, określonych parametrami 'd1:' i 'd2:'. Nowa struktura katalogowa powstanie na urządzeniu d2: w ten sposób, że katalog podstawowy z urządzenia d1: stanie się jednym z katalogów podrzędnych, zapisanych bezpośrednio w katalogu podstawowym na urządzeniu d2:, pod nazwą określona parametrem 'directory'. Program obsługi rozkazu JOIN zakłada jednocześnie, że:

- katalog o nazwie 'directory' dotychczas nie istniał w katalogu podstawowym na urządzeniu d2:. W takim przypadku, po zainicjowaniu wykonania rozkazu JOIN, zostanie tam utworzony,
- katalog o nazwie 'directory' istnieje już w katalogu podstawowym na urządzeniu d2:. Jeśli jest on pustym katalogiem, tzn. jeśli zawiera tylko dwa zbiory specjalne:

. <DIR>

.. <DIR>

to 'przejmie' funkcję katalogu podstawowego z urządzenia d1:. W wypadku, gdy katalog ten nie jest pustym katalogiem, to na ekranie monitora zostanie wyświetlony następujący komunikat:

Directory not Empty

co kończy wykonywanie rozkazu JOIN bez utworzenia nowej struktury.

Poszczególne parametry występujące w linii komendy rozkazu JOIN mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym JOIN, określa miejsce, w którym znajduje się program obsługi rozkazu JOIN,

d1: określa urządzenie, z którego struktura katalogowa zostanie dopisana do struktury katalogowej na urządzeniu

d2: ,

d2: określa urządzenie, na którym będzie modyfikowana struktura katalogowa,

\directory określa nazwę katalogu na urządzeniu d2:, który 'przejmie' funkcje katalogu podstawowego z urządzenia d1:. Katalog ten będzie (lub już jest) zapisany bezpośrednio w katalogu podstawowym, na urządzeniu d2:.

klucz /D powoduje anulowanie pojedynczej struktury katalogowej, utworzonej z dwóch struktur katalogowych: z urządzenia d1: i d2:. Struktury katalogowe na obu tych urządzeniach wracają do swojej pierwotnej postaci.

Uwagi dotyczące posługiwania się rozkazem JOIN:

1. Struktury katalogowej z urządzenia d1: nie można dopisywać do struktury na urządzeniu d2: w innym katalogu niż katalog podstawowy. Innymi słowy: katalog podrzędny, określony parametrem 'directory', który przejmie funkcje katalogu podstawowego z urządzenia d1: musi być zapisany bezpośrednio w katalogu podstawowym, na urządzeniu d2:. W przypadku próby ominięcia tej zasady, wykonywanie rozkazu JOIN zostanie zakończone z wyświetleniem następującego komunikatu:

Invalid parameter

2. Po zainicjowaniu wykonania rozkazu JOIN nie jest możliwe odwoływanie się do urządzenia d1:, aż do chwili wykonania kolejnego rozkazu JOIN, anulującego skutki poprzedniego:

[d:][ścieżka]JOIN d1:/D

Każde takie odwołanie, przed wykonaniem powyższego rozkazu, kończy się następującym komunikatem:

Invalid drive specification

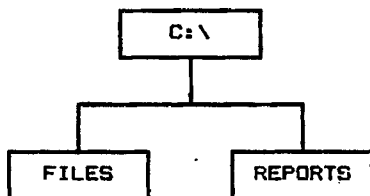
3. Rozkaz JOIN, zainicjowany bez żadnych parametrów, wyświetla informacje o aktualnych zmianach, utworzonych poprzednimi rozkazami JOIN.
4. Wykonanie rozkazu JOIN nie zależy od tego, którym katalogiem był katalog roboczy na urządzeniu d1:.
5. Należy unikać stosowania rozkazu JOIN w połączeniu z rozkazami ASSIGN i SUBST, gdyż w razie nieuwagi użyt-

kownika może wywołać to trudne do przewidzenia skutki.

6. Rozkazy: BACKUP, RESTORE, FORMAT, DISKCOPY i DISKCOMP ignorują efekt wywołany wykonaniem rozkazu JOIN.

Przykład:

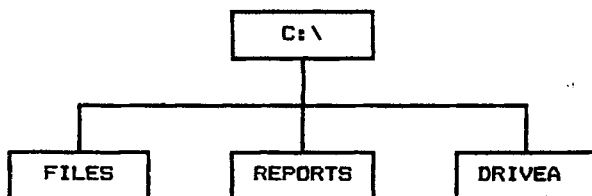
Niech urządzeniem roboczym będzie pierwsza stacja dyskieta A:. Niech katalogiem roboczym na urządzeniu C: będzie katalog podstawowy. Niech ponadto struktura katalogowa na urządzeniu C: będzie miała następującą postać:



Zainicjowanie rozkazu JOIN o następującej postaci:

B>JOIN A: C:\DRIVEA

zmodyfikuje w następujący sposób strukturę katalogową na urządzeniu C:



Katalog podstawowy z urządzenia A: stał się katalogiem podrzędnym, zapisanym w katalogu podstawowym na urządzeniu C: pod nazwą DRIVEA.

Wykonanie rozkazu DIR C: spowoduje wyświetlenie następującej informacji:

Volume in drive C: is FIXEDC
Directory for C:\

FILES	<DIR>			
REPORTS	<DIR>	8-21-85	2:10p	
DRIVEA	<DIR>	9-23-85	5:55p	
			9-30-85	12:03a
3 File(s)		1048956 bytes free		

Wywołanie rozkazu DIR o następującej postaci:

DIR A:

spowoduje wyświetlenie następującego komunikatu:

Invalid drive specification

Aby wyświetlić zawartość katalogu podstawowego z urządzenia A należy zatem użyć następującego rozkazu DIR:

DIR C:\DRIVEA

Wywołanie rozkazu JOIN o następującej postaci:

B>JOIN

spowoduje wyświetlenie następującej informacji:

A: => C:\DRIVEA

Wywołanie rozkazu JOIN o następującej postaci:

B>JOIN A: /D

spowoduje przywrócenie struktur katalogowych sprzed wykonania rozkazu:

B>JOIN A: C:\DRIVEA

6.25. Rozkaz KEYBxx (Load Keyboard, nie występuje w wersji MS DOS 3.10).

Format:

[d:][ścieżka]KEYBxx

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz KEYBxx pozwala na załadowanie do pamięci operacyjnej specjalnego programu obsługi klawiatury o układzie obowiązującym w innym kraju niż w USA. Program ten przejmie funkcje obsługi klawiatury od standardowego, zapisanego w ROM-BIOS programu obsługi klawiatury USA.

Poszczególne parametry występujące w linii komendy rozkazu KEYBxx mają następujące znaczenie:

[d:] [ścieżka], przed słowem kluczowym KEYBxx, określa miejsce, gdzie znajduje się program obsługi rozkazu KEYBxx,

xx, stanowiący część nazwy rozkazu KEYBxx, określa dwuliterowy kod kraju, którego układ klawiatury ma obowiązywać po zainicjowaniu rozkazu KEYBxx. Poniższa tabela definiuje pełny zestaw rozkazów, w zależności od przyjętego kraju:

Rozkaz	Układ klawiatury
KEYBUK	Wielka Brytania
KEYBGR	RFN
KEYBFR	Francja
KEYBIT	Włochy
KEYBSP	Hiszpania

Uwagi dotyczące posługiwania się rozkazem KEYBxx:

1. Po restarcie systemu operacyjnego PC DOS, do pamięci operacyjnej można załadować tylko jeden program obsługi klawiatury o innym układzie niż w USA. Program ten pozostaje w pamięci do chwili wyłączenia zasilania lub ponownego restartu. Jeśli użytkownik wykona powtórnie rozkaz KEYBxx dla innego niż uprzednio układu klawiatury, to obowiązującym stanie się nowy układ, który będzie obowiązywał do chwili wywołania kolejnego rozkazu KEYBxx.
2. System operacyjny pozwala na wymiennność zainicjowanego układu klawiatury z układem obowiązującym w USA. Mając

załadowany do pamięci operacyjnej program obsługi dowolnego układu klawiatury można:

kombinacją klawiszy 'Ctrl/Alt/F1' wybrać format klawiatury USA,

kombinacją klawiszy 'Ctrl/Alt/F2' przywrócić format wybrany ostatnim rozkazem KEYBxx.

UWAŻA:

Programy obsługi poszczególnych układów klawiatur z wersji PC DOS 3.20 nie są kompatybilne z programami obsługi klawiatur z wersji PC DOS 1.10, stąd te ostatnie nie mogą być wykorzystywane w wersji 3.20.

Przykład:

Rozkaz KEYBxx o następującej postaci:

A:\KEYBGR

ładuje do pamięci operacyjnej program obsługi układu klawiatury obowiązujący w RFN. Użycie kombinacji klawiszy 'Ctrl/Alt/F1' powoduje przejście do standardowego programu obsługi (układ klawiatury obowiązujący w USA). Powrót do wczytanego programu obsługi klawiatury (obowiązującego w RFN) nastąpi po wciśnięciu kombinacji klawiszy 'Ctrl/Alt/F2'. Znak '\' informuje system operacyjny PC DOS, że program KEYBGR znajduje się w katalogu podstawowym (systemowym) na urządzeniu A:.

6.26. Rozkaz LABEL (Volume Label, nie występuje w wersji .IS DOS 3.10).

Format:

[d:][ścieżka]LABEL [d1:][etykieta]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz LABEL pozwala na tworzenie, zmianę lub usunięcie etykiety dyskowej. Etykieta dyskowa jest nazwą identyfikującą dysk twardy lub dyskietkę.

Poszczególne parametry występujące w linii komendy rozkazu LABEL mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym LABEL, określa miejsce, gdzie znajduje się program obsługi rozkazu LABEL,

[d2:] określa urządzenie, na którym znajduje się etykietowany dysk lub dyskietka. Jest to parametr opcjonalny, którego pominięcie spowoduje domyślne przyjęcie urządzenia roboczego,

[etykieta] określa nazwę identyfikującą dysk lub dyskietkę. Parametr 'etykieta' może zawierać od 1 do 11 znaków, przy czym dozwolone jest używanie wszystkich tych znaków, które mogą tworzyć nazwę zbioru.

Uwagi dotyczące posługiwania się rozkazem LABEL:

1. Jeśli w specyfikacji linii komendy rozkazu LABEL pominięto parametr 'etykieta', to system operacyjny PC DOS zgłasza się następującym komunikatem:

Volume in drive X is xxxxxxxxxxxx
Volume label (11 characters, ENTER for none)?

po którym można wpisać nazwę etykiety i potwierdzić ją wciśnięciem klawisza 'Enter'. Pozwala to jednocześnie na zmianę nazwy dotychczasowej etykiety.

2. Jeśli mimo wyświetlonego przynaglenia użytkownik wcisnął tylko klawisz 'Enter', a dysk lub dyskietka były już kiedyś zaetykietowane, to system operacyjny zgłasza kolejne pytanie:

Delete current volume label (Y/N)?

Wciśnięcie klawisza 'Y', a następnie 'Enter' oznacza, że dysk ma pozostać bez etykiety.

3. Jeśli w specyfikacji parametru 'etykieta' podano więcej niż 11 znaków, wówczas dysk lub dyskietka zostanie oznaczona pierwszymi 11 znakami.

Przykłady:

Rozkaz LABEL o następującej postaci:

A>LABEL A:ADRESS

spowoduje oznaczenie dyskietki, znajdującej się na urządzeniu A:, etykietą ADDRESS:. Program obsługi rozkazu LABEL znajduje się na tej samej dyskietce.

Aby zmienić etykietę dyskietki z powyższego przykładu należy zainicjować wykonanie następującego rozkazu LABEL:

A>LABEL A:

System odpowiada:

Volume in drive A is ADDRESS

Volume label (11 characters, ENTER for none)?

Jeśli użytkownik wpisze nową nazwę np. PROGRAMS to oznaczy nową etykietą tę dyskietkę. Wciśnięcie tylko klawisza 'Enter' spowoduje wyświetlenie następnego komunikatu:

Delete current volume label (Y/N)?

Wciśnięcie klawisza 'Y', a następnie 'Enter' oznacza usunięcie starej etykiety (dyskietka będzie bez etykiety). Wciśnięcie klawisza 'N', a następnie 'Enter' oznacza pozostawienie starej etykiety dyskietki na urządzeniu A:.

6.27. Rozkaz MKDIR (Make Directory).

Format:

MKDIR [d:]ścieżka

lub:

MD [d:]ścieżka

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz MKDIR służy do tworzenia katalogów podrzędnych w katalogu podstawowym lub dowolnym innym katalogu podrzędnym. Nazwa utworzonego katalogu pojawi się w katalogu wyższego poziomu wraz z dopiskiem <DIR>, w miejscu, w którym podawana jest wielkość zbioru. Nazwa katalogu może składać się od 1 do 8 znaków, za wyjątkiem:

. " / \ [] : ! < > + ; ,

ani żadnego ze znaków ASCII o kodzie mniejszym niż 20H.

W każdym katalogu, za wyjątkiem katalogu podstawowego, można utworzyć dowolną ilość katalogów podrzędnych, limitowana tylko pojemnością dysku lub dyskietki. W przypadku katalogu podstawowego, ilość ta zależy od zaformatowania i rodzaju dysku lub dyskietki (patrz podrozdział pt. 'Katalogi w systemie operacyjnym PC DOS').

U W A G A :

Należy zachować szczególną ostrożność przy tworzeniu katalogów podrzędnych, wykorzystując efekty zainicjowania rozkazów ASSIGN, JOIN i SUBST.

Poszczególne parametry występujące w linii komendy rozkazu MKDIR mają następujące znaczenie:

[d:] określa urządzenie, na którym ma być utworzony katalog podrzędny. Jeśli użytkownik pominął nazwę urządzenia, to domyślnie zostanie przyjęte urządzenie robocze.

ścieżka określa ścieżkę z katalogu podstawowego lub aktualnego katalogu roboczego do tworzonego katalogu.

Ostatnim ogniwem tej ścieżki jest nazwa tworzonego katalogu podrzędnego. Nazwy poszczególnych katalogów muszą być odseparowane od siebie znakami '\'. Długość wyspecyfikowanej ścieżki z katalogu podstawowego do tworzonego katalogu podrzędnego, włącznie z tym katalogiem i znakami '\', nie może zawierać więcej niż 63 znaki.

Przykłady:

Rozkaz MKDIR o następującej postaci:

```
A>MKDIR \LEVEL1
```

spowoduje utworzenie w katalogu podstawowym, na roboczym urządzeniu A:, katalogu podrzędnego o nazwie LEVEL1.

Rozkaz MKDIR o następującej postaci:

```
A>MKDIR \LEVEL1\LEVEL2
```

spowoduje utworzenie w katalogu LEVEL1, na roboczym urządzeniu A:, katalogu podrzędnego o nazwie LEVEL2. Należy zwrócić uwagę, że wyspecyfikowano ścieżkę od katalogu podstawowego. Jeśli katalogiem roboczym byłby katalog LEVEL1, to powyższy rozkaz można uprościć do następującej postaci:

```
A>MKDIR LEVEL2
```

6.28. Rozkaz MODE (nie występuje w wersji MS DOS 3.10).

Format:

opcja 1: [d:][ścieżka]MODE LPT#[:][n][,][m][,][P]]
opcja 2: [d:][ścieżka]MODE n
lub: [d:][ścieżka]MODE [n],m[,T]
opcja 3: [d:][ścieżka]MODE COMn[:][baud
[,parity[,batabits[,stopbits[,P]]]]
opcja 4: [d:][ścieżka]MODE LPT#:=COMn

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz `MODE` kontroluje tryb operacji na drukarkę, przełącza adaptory ekranowe, określa tryb operacji dla adaptera grafiki kolorowej, określa opcje dla adaptera komunikacyjnego oraz pozwala na przestanie danych, kierowanych zamiast na drukarkę, na asynchroniczny układ transmisji szeregowej.

Parametry `'d:'` i `'ścieżka'`, przed słowem kluczowym `MODE`, określają miejsce, gdzie znajduje się program obsługi rozkazu `MODE`. Pominięcie lub podanie niedozwolonych parametrów `'n'` lub `'m'` (zostaną omówione w dalszej części) powoduje, że ich wartości nie ulegną zmianie.

Opcja 1: - ustawienie parametrów drukarki:

```
MODE LPT#[[:][n]][,m][,P]
```

gdzie:

- # określa numer drukarki. Parametr # może być równy: 1, 2, lub 3,
- n określa maksymalną ilość znaków w linii. Parametr n może zmieniać się od 80 do 132, wartość domyślna wynosi 80,
- m określa ilość linii na cal. Parametr m może być równy 6 lub 8, wartość domyślna wynosi 8,
- P powoduje ciągłe ponawianie próby pisania przy błędzie typu time-out. Ponawianie można przerwać wciskając kombinację klawiszy `'Ctrl/Break'`.

Przykład:

Rozkaz `MODE` o następującej postaci:

```
MODE LPT1:132,8
```

ustawia następujące parametry drukarki LPT1: maksymalna liczba znaków w linii wynosi 132, ilość linii na cal wynosi 8.

Opcja 2: - ustawienie parametrów adaptera ekranowego i adaptera koloru/grafiki:

MODE n lub MODE [n],m[,T]

gdzie:

n może przybierać jedną z poniższych wartości:

- 40 ustawia szerokość ekranu na 40 znaków w linii dla adaptera koloru/grafiki,
- 80 ustawia szerokość ekranu na 80 znaków w linii dla adaptera koloru/grafiki,
- BW40 ustawia w trybie wyświetlania czarno-białym szerokość ekranu na 40 znaków w linii,
- BW80 ustawia w trybie wyświetlania czarno-białym szerokość ekranu na 80 znaków w linii,
- CO40 ustawia w kolorowym trybie pracy szerokość ekranu na 40 znaków w linii,
- CO80 ustawia w kolorowym trybie pracy szerokość ekranu na 80 znaków w linii,
- MONO ustawia monochromatyczny tryb pracy i szerokość ekranu na 80 znaków w linii.

m może przybierać jedną z poniższych dwóch wartości:

- R umożliwia przesuwanie wydruku w prawo,
- L umożliwia przesuwanie wydruku w lewo,

T oznacza żądanie testu znakowego dla ustawienia wyświetlania.

Wyjaśnienie dotyczące parametrów 'm' i 'T':

W zależności od ustawienia trybu pracy (40 lub 80 kolumnowy) i parametru 'm', użytkownik ma możliwość przesuwania na ekranie monitora w prawo lub w lewo wzorcowej linii znakowej, o 1 znak w trybie 40 kolumnowym, lub o 2 znaki w trybie 80 kolumnowym. Sterowaniem przesuwania zajmuje się system operacyjny, żądając potwierdzenia 'Y' lub zaprzeczenia 'N' poprawności ustawienia wzorcowej linii. Odpowiadając 'Y' użytkownik kończy wykonywanie testu (a zarazem rozkazu MODE), natomiast odpowiedź 'N' powoduje kolejne przesunięcie linii znakowej we wskazanym kierunku i kolejne pytanie skierowane do użytkownika. Procedura ta jest powtarzana, aż do właściwego ustawienia linii wzorcowej na ekranie monitora. Zakończenie wykonywania rozkazu MODE oznacza ustalenie trybu pracy określonego parametrem 'n'. Np. rozkaz o postaci:

MODE 80,R,T

ustala 80-cio kolumnowy tryb pracy i żąda wyświetlenia wzorcowej linii znakowej z przesuwaniem jej w prawo.

Opcja 3: - ustawienie parametrów pracy asynchronicznego układu transmisji:

MODE COMn[:]baud[,parity[,databits[,stopbits[,P]]]]

gdzie:

n określa numer asynchronicznego układu transmisji, n=1,2

baud określa szybkość przesyłania danych w bodach, parametr baud może być równy: 110, 150, 300, 600, 1200, 2400, 4800, 9600 (tylko dwie pierwsze cyfry są znaczące),

parity określa parzystość:
N (none - brak kontroli parzystości),
O (odd - kontrola nieparzystości),
E (even - kontrola parzystości, opcja domyślna).

databits określa długość słowa, databits=7,8 bitów, standardowo jest przyjmowane słowo 7 bitowe,

stopbits określa liczbę bitów stopu, stopbits=1,2 standardowo są przyjmowane 2 bity stopu przy szybkości transmisji 110 bodów, lub 1 bit stopu przy innej szybkości transmisji,

P wskazuje, że asynchroniczny układ transmisji szeregowy będzie używany doysterowywania drukarki, a więc przy błędzie typu time-out spowoduje ciągłe powstawianie próby pisania (można to przerwać wciskając kombinację klawiszy 'Ctrl/Break')

Powyższe parametry określają protokół transmisji szeregowy przez złącze COMn. Określając ten protokół można pominąć wszystkie parametry, za wyjątkiem parametru 'baud', wstawiając w ich miejsce przecinki (,). System przyjmuje wówczas ich wartości domyślne.

Przykład:

Aby ustawić asynchroniczny układ transmisji nr 1, służący doysterowania drukarki, na transmisję o szybkości 1200 bodów, kontrolę nieparzystości, 8 bitów danych i 1 bit stopu (za wyjątkiem parametrów 'baud' i 'P' są to wartości domyślne) należy wykonać jeden z równoważnych rozkazów MODE:

```
MODE COM1:12,n,8,1,p
MODE COM1:12,,,,,p
```

Opcja 4: - wysyłanie danych, zamiast na LPT#, do asynchronicznego układu transmisji szeregowej COMn:

```
MODE LPT#[:] = COMn
```

gdzie:

określa numer drukarki #=1,2

n określa numer asynchronicznego układu transmisji n=1,2,

U W A G A :

Przed podłączeniem drukarki do złącza szeregowego COMn należy, przy pomocy opcji 3 rozkazu MODE, ustawić parametry transmisji tego złącza.

Przykład:

Aby skierować dane wysyłane na LPT1: do drukarki, podłączonej do złącza szeregowego COM2, należy użyć następującego rozkazu MODE:

```
MODE LPT1:=COM2:
```

6.29. Rozkaz MORE (More Filter).

Format:

[d:][ścieżka]MORE

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz (filtr) MORE pozwala na czytanie danych ze standardowego urządzenia wejściowego i wyświetlanie ich na ekranie monitora w ten sposób, że po zapełnieniu ekranu wyświetlanie zostaje zawieszona i pojawia się następujący komunikat:

--MORE--

Przyciśnięcie dowolnego klawisza powoduje wyświetlenie dalszej części zbioru, aż do kolejnego zapełnienia ekranu. System operacyjny PC DOS kontynuuje tę procedurę aż do wyświetlenia całego zbioru, lub przerwania wykonywania rozkazu MORE wciśnięciem kombinacji klawiszy 'Ctrl/Break'. Parametry 'd:' i 'ścieżka', przed słowem kluczowym MORE, określają miejsce, w którym znajduje się program obsługi rozkazu MORE.

Rozkaz MORE w zasadzie jest używany w dwóch przypadkach:

1. Jako samodzielny rozkaz. Format linii komendy ma następująca postać:

[d:][ścieżka]MORE [<d1:][ścieżka]nazwa
[.rozszerzenie]

gdzie [d1:][ścieżka]nazwa[.rozszerzenie] określają zbiór, który zostanie wyświetlony na ekranie monitora. Znak '<' wymusi odczytywanie wyspecyfikowanego zbioru z dysku d1: (jest to tzw. wtórna definicja standardowego urządzenia wejściowego - zamiast klawiatury zbiór z dysku).

2. Jako filtr realizujący z innym rozkazem systemu operacyjnego PC DOS tzw. przetwarzanie potokowe. Format linii komendy ma wówczas następująca postać:

rozkaz : [d:]ścieżka\MORE

Przykłady:

Rozkaz MORE o następującej postaci:

A\MORE <TEST.ASM

spowoduje wyświetlenie strona po stronie zbioru TEST.ASM. Zarówno program obsługi rozkazu MORE jak i zbiór TEST.ASM znajdują się w katalogu roboczym, na urządzeniu roboczym.

Rozkaz MORE o następującej postaci:

C>A:\USER TREE C:/F ! B:MORE

wraz z rozkazem TREE, spowoduje tzw. przetwarzanie potokowe. Rozkaz TREE z kluczem /F wyświetla strukturę katalogów z urządzenia C: wraz z nazwami wszystkich zapisanych w nim zbiorów. Filtr MORE pozwala wstrzymywać wyświetlanie po zapełnieniu ekranu. Program obsługi rozkazu TREE znajduje się na urządzeniu A: w katalogu \USER. Program obsługi rozkazu MORE znajduje się w katalogu roboczym, na urządzeniu B:.

6.30. Rozkaz PATH (Set Search Directory).

Format:

PATH [[d1:]ścieżka1[;[d2:]ścieżka2]...]]

lub:

PATH ;

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz PATH określa jakie katalogi i na jakich urządzeniach będą przeglądane przez system operacyjny PC DOS w poszukiwaniu programu, rozkazu zewnętrznego, instrukcji lub zbioru rozkazów przetwarzania wsadowego, które zostały zainicjowane z katalogu roboczego bez określenia miejsca ich zapisu.

Linia komendy rozkazu PATH może zawierać całą listę wyspecyfikowanych urządzeń i katalogów, oddzielonych od siebie średnikami (;). Rozkaz PATH nie zmienia katalogu roboczego. Jeśli w linii komendy nie wyspecyfikowano żadnych parametrów, to zostają wyświetlone wszystkie, zainicjowane tym rozkazem, pary: 'urządzenie' 'ścieżka-do-zbioru'. Wywołanie rozkazu PATH tylko ze średnikiem po słowie kluczowym PATH, anuluje wszystkie poprzednio podjęte ustalenia, co oznacza, że ewentualne poszukiwania będą prowadzone tylko w katalogu roboczym (jest to stan domyślny, obowiązujący w systemie operacyjnym od chwili wykonania jego restartu do wykonania pierwszego rozkazu PATH z parametrami).

Uwagi dotyczące posługiwania się rozkazem PATH:

1. Rozkaz PATH pozwala na poszukiwanie tylko zainicjowanych do wykonania zbiorów (programów) o rozszerzeniach .COM, .EXE lub .BAT.
2. Jeśli jedna lub więcej, z wyspecyfikowanych w linii komendy rozkazu PATH, ścieżek-do-zbioru nie istnieje na wskazanym urządzeniu, to podczas wykonywania poszukiwań zostanie zignorowana.

Przykład:

Program o nazwie MYPROG.COM zapisano w podkatalogu \MYDIR, znajdującym się w katalogu podstawowym na urządzeniu B:. Niech urządzeniem roboczym będzie urządzenie A, katalogiem roboczym katalog LEVEL3, przy założeniu następującej ścieżki z katalogu podstawowego do katalogu roboczego: \LEVEL2\LEVEL3.

Zainicjowanie wykonania programu MYPROG.COM można wykonać:

- albo poprzez zmianę urządzenia roboczego na B: i katalogu roboczego na MYDIR, a następnie wywołując program MYPROG,
- albo bez zmiany katalogu i urządzenia roboczego, wykonując rozkaz PATH o następującej postaci:

PATH A:\LEVEL2\LEVEL3;B\MYDIR

a następnie także wywołując program MYPROG.

Jeśli system operacyjny nie odzyska wywołanego programu ani w katalogu roboczym, ani w żadnym innym katalogu określonym rozkazem PATH, to na ekranie monitora zostanie wyświetlony następujący komunikat:

Bad command or file name

Jeśli po zainicjowaniu rozkazu PATH, o postaci jak powyżej, wywołano go powtórnie, ale bez parametrów, to na ekranie monitora pojawi się następująca informacja:

PATH=A:\LEVEL2\LEVEL3;B:\MYDIR

6.31. Rozkaz PRINT.

Format:

[d:][ścieżka]PRINT [/D:device][/B:buffsiz][/U:busytick]
[/M:maxtick][/S:timeslice][/Q:quesiz]
[[di:][ścieżka]nazwa[.rozszerzenie]...]
[/C][/T][/P]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz PRINT dołącza wyspecyfikowany zbiór lub zbiory do kolejki wyjściowej na drukarkę. Zbiory te są następnie kolejno drukowane. W czasie drukowania mikrokomputer może być wykorzystywany do innych prac, chociaż spowalnia to zarówno wydruk jak i wykonywaną czynność.

Poszczególne parametry występujące w linii komendy rozkazu PRINT mają następujące znaczenie:

[d:]ścieżka, przed słowem kluczowym PRINT, określa miejsce, w którym znajduje się program obsługi rozkazu PRINT,

klucz /D:device - określa urządzenie, na które zostaną przesłane zbiory do wydrukowania. Za parametr 'device' można podstawiać urządzenie PRN, LPT1, LPT2, LPT3, COM1, COM2 i AUX, domyślnie system przyjmuje PRN. Klucz /D z parametrem 'device', jeśli został wyspecyfikowany w linii komendy, to musi wystąpić jako pierwszy spośród wszystkich kluczy,

klucz /B:buffsiz - określa wielkość wewnętrznego bufora, wyrażoną w bajtach. Domyślnie wartość parametru 'buffsiz' wynosi 512 bajtów. Zwiększenie wewnętrznego bufora polepsza efektywność wykonywania wydruków, ale zmniejsza rozmiar pamięci operacyjnej przeznaczonej dla użytkownika,

klucz /U:busytick - określa czas oczekiwania przez program obsługujący rozkaz PRINT na potwierdzenie gotowości drukarki. Czas ten jest wyrażony w taktach zegara, a wartość domyślna parametru 'busytick' wynosi 1 takt,

klucz /M:maxtick - określa maksymalny czas wydruku znaków, wyrażony w taktach zegara. Wartość parametru 'maxtick' może zmieniać się od 1 do 255, domyślnie wynosi 2 takty,

klucz /S:timeslice - określa czas pobierania porcji danych, wyrażony w taktach zegara. Wartość parametru 'timeslice' może zmieniać się od 1 do 255, domyślnie wynosi 8 taktów,

klucz /Q:quesiz - określa maksymalną ilość zbiorów, które mogą jednocześnie znaleźć się w kolejce do drukarki. Wartość parametru 'quesiz' może zmieniać się od 1 do 32, domyślnie wynosi 10 zbiorów,

klucz /T - powoduje przerwanie drukowania i usunięcie

wszystkich zbiorów z kolejki do drukarki. Jeśli rozkaz PRINT z kluczem /T został zainicjowany w trakcie drukowania zbioru, to wydruk zostaje przerwany z wydrukowaniem odpowiedniego komunikatu, a następnie drukarka przewija papier do początku nowej strony i uruchamia sygnał dźwiękowy,

klucz /C - pozwala wybiórczo usuwać zbiory oczekujące na wydruk. Z kolejki do drukarki zostaną usunięte następujące zbiory:

- zbiór (lub grupa zbiorów, jeśli dla ich określenia użyto znaków specjalnych '*' lub '?'), którego nazwę wyspecyfikowano w linii komendy rozkazu PRINT bezpośrednio przed kluczem /C,
- wszystkie zbiory wyspecyfikowane w linii komendy rozkazu PRINT, od klucza /C do końca linii, lub od klucza /C do zbioru poprzedzającego klucz /P, z wyłączeniem tego zbioru.

klucz /P pozwala na dodawanie zbiorów do kolejki oczekującej na wydruk (jest to opcja standardowa). Do kolejki zostaną dodane następujące zbiory:

- zbiór (lub grupa zbiorów, jeśli dla ich określenia użyto znaków specjalnych '*' i '?'), którego nazwę wyspecyfikowano w linii komendy rozkazu PRINT bezpośrednio przed kluczem /P,
- wszystkie zbiory wyspecyfikowane w linii komendy rozkazu PRINT, od klucza /P do końca linii, lub od klucza /P do zbioru poprzedzającego klucz /C, z wyłączeniem tego zbioru.

Klucze /D, /B, /Q, /S, /U i /M mogą zostać użyte tylko przy pierwszym od chwili restartu systemu operacyjnego, wywołaniu rozkazu PRINT. Jeśli użytkownik próbuje uczynić to po raz wtóry, to system operacyjny PC DOS ignoruje ten rozkaz wyświetlając następujący komunikat:

Invalid Parameters

Uwagi dotyczące posługiwania się rozkazem PRINT:

1. W specyfikacjach nazw zbiorów, występujących w linii komendy rozkazu PRINT, mogą wystąpić znaki specjalne '?' i '?'.
i '?'.
2. Po dołączeniu zbioru do kolejki zbiorów oczekujących na wydruk, użytkownik może zmieniać swój katalog roboczy, bez wpływu na wykonanie drukowania.
3. Do momentu zakończenia operacji wydruku nie wolno usuwać dyskietek, na których znajdują się zbiory przeznaczone do drukowania.
4. Jeśli rozkaz PRINT został wywołany bez żadnych parametrów, to wyświetli nazwy wszystkich zbiorów oczekujących w kolejce do drukarki.
5. Jeśli podczas pierwszego od chwili restaru systemu operacyjnego wywołania rozkazu PRINT, nie wyspecyfikowano klucza /D z parametrem 'device', to na ekranie monitora wyświetlony zostanie następujący komunikat:

Name of list device [PRN:]

Należy wówczas określić nazwę urządzenia drukującego: LPT1, LPT2, LPT3, PRN, COM1, COM2 lub AUX. Wartością domyślną jest PRN i decydując się na ten parametr wystarczy wcisnąć klawisz 'Enter', bez specyfikowania oznaczenia tego urządzenia.

6. Każdy drukowany zbiór system operacyjny pobiera z kolejki zbiorów oczekujących na wydruk. Po wydrukowaniu danego zbioru drukarka automatycznie przesuwą papier do początku nowej strony.
7. Jeśli podczas wykonywania wydruku wystąpi błąd w odczycie z dysku drukowanego zbioru, to:
 - wydruk tego zbioru zostanie przerwany z dodrukowaniem informacji o błędzie i przerwaniu wydruku,
 - drukarka przesunie papier do początku nowej strony,
 - rozpocznie się drukowanie następnego zbioru z kolejki zbiorów oczekujących na wydruk.
8. Jeśli rozkaz PRINT został użyty z kluczami /T lub /C dla usunięcia zbioru (zbiorów) z kolejki do drukarki, to:

- system operacyjny uruchomi sygnał dźwiękowy drukarki,
- w przypadku klucza /T zostanie dodrukowany następujący komunikat:

All files canceled by operator

- w przypadku klucza /C zostanie wydrukowana nazwa usuniętego zbioru oraz następujący komunikat:

File canceled by operator

- drukarka przesunie papier do początku nowej strony,
- rozpocznie się drukowanie następnych zbiorów z kolejki zbiorów oczekujących na wydruk (jeśli użyto klucza /C).

9. Podczas wykonywania wydruków zbiorów z kolejki do drukarki użytkownik nie może wykonywać żadnych operacji związanych z pracującym urządzeniem drukującym, np. drukowania aktualnej zawartości ekranu monitora, kierowanie danych zamiast na ekran monitora na drukarkę etc.

Przykłady:

Niech w poniższym przykładzie następujący rozkaz PRINT zostanie wywołany po raz pierwszy od chwili wykonania ostatniego restartu systemu operacyjnego:

PRINT A:TEMPL.LST

Po jego zainicjowaniu pojawia się następujący komunikat:

Name of list device [PRN]:

w którym system operacyjny PC DOS żąda określenia urządzenia drukującego.

Rozkaz PRINT o następującej postaci:

PRINT /T

usuwa wszystkie zbiory z kolejki zbiorów oczekujących na wydruk. Należy zwrócić uwagę, że wszelkie inne informacje podawane w linii komendy rozkazu PRINT, w której wyspecyfikowano klucz /T zostają zignorowane.

Rozkaz PRINT o następującej postaci:

PRINT TEMP.* /C

usuwa wszystkie zbiory o nazwie TEMP i dowolnym rozszerzeniu z kolejki zbiorów oczekujących na wydruk.

Rozkaz PRINT o następującej postaci:

PRINT A:TEMP1.LST /C A:TEMP2.LST A:TEMP3.LST

usuwa trzy zbiory: TEMP1.LST, TEMP2.LST i TEMP3.LST z kolejki zbiorów oczekujących na wydruk.

Rozkaz PRINT o następującej postaci:

PRINT TEMP1.LST /C TEMP2.LST /P TEMP3.LST

usuwa zbiór TEMP1.LST i dodaje zbiory TEMP2.LST i TEMP3.LST do kolejki zbiorów oczekujących na wydruk.

Rozkaz PRINT o następującej postaci:

PRINT TEMP1.LST TEMP2.LST TEMP3.LST /C

dodaje zbiory TEMP1.LST i TEMP2.LST oraz usuwa zbiór TEMP3.LST z kolejki zbiorów oczekujących na wydruk.

6.32. Rozkaz PROMPT (Set System Prompt).

Format:

PROMPT [prompt-text]

Typ:

Rozkaz wewnętrzny.

Opis:

System operacyjny PC DOS domyślnie zgłasza się wyświetleniem nazwy urządzenia roboczego i znaku >. Rozkaz PROMPT pozwala wprowadzać nową formę zgłoszenia systemu. Zgłoszenie to będzie polegało na wyświetleniu tekstu określonego parametrem 'prompt-text'. Jeśli w tym tekście występują znaki \$ (znak dolara), to rozkaz PROMPT dodatkowo wyświetli pewne informacje lub znaki, których formę określa znak występujący bezpośrednio za znakiem dolara (patrz przykłady). Wprowadzenie tego rozkazu bez opcjonalnego parametru 'prompt-text' przywraca normalne zgłoszenie systemu operacyjnego PC DOS.

Znaki poprzedzone znakiem \$ (znak dolara) pozwalają na wyświetlenie:

- t - aktualnego czasu systemowego,
- d - aktualnej daty systemowej,
- p - katalogu roboczego urządzenia roboczego,
- v - numeru wersji załadowanego systemu DOS,
- n - nazwy urządzenia roboczego,
- g - znaku > ,
- l - znaku < ,
- b - znaku ! ,
- q - znaku równości (nie występuje w wersji MS DOS 3.10),
- h - usunięcie ostatniego znaku (nie występuje w wersji MS DOS 3.10),
- e - znaku Esc,
- _ - powrót karetki/rozpoczęcie nowej linii.

Każdy inny znak spoza powyższej listy rozkaz PROMPT ignoruje.

Przykłady:

PROMPT \$n\$g - system operacyjny zgłasza się literą identyfikującą urządzenie robocze i znakiem >.

PROMPT HELLO - system operacyjny zgłasza się komunikatem: HELLO

PROMPT \$p\$g - system operacyjny zgłasza się nazwą katalogu roboczego i znakiem >.

PROMPT Time=\$t\$_Date=\$dg - system operacyjny zgłasza się wyświetlając:
 Time=(aktualny czas)
 Date=(aktualna data) i znak >.

6.33. Rozkaz RECOVER.

Format:

Opcja 1: [d:][ścieżka]RECOVER [d1:][ścieżka1]nazwa[.rozszerzenie]
 lub:

Opcja 2: [d:][ścieżka]RECOVER d1:

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz RECOVER pozwala odzyskać zbiór (zbiory), którego pewna część została zapisana w miejscu, gdzie nośnik magnetyczny uległ uszkodzeniu. Odzyskiwanie zbioru polega na analizie jego zapisu, zarejestrowaniu miejsc, gdzie nośnik jest uszkodzony i dokonaniu zmian w tzw. FAT (ang. File Allocation Table), czyli tablicy, w której m.in. rejestruje się rozmieszczenie poszczególnych zbiorów w różnych sektorach na dysku lub dyskietce. Odzyskany zbiór zwykle nie odpowiada dokładnie zbiorowi przed uszkodzeniem, gdyż podczas odzyskiwania zostaje pominięta informacja zapisana w uszkodzonych miejscach nośnika. Rozkazu RECOVER można używać do odzyskiwania pojedynczego zbioru (opcja 1) lub całej informacji zapisanej na dysku lub dyskietce (opcja 2). W obu przypadkach opcjonalne parametry 'd:' i 'ścieżka', przed słowem kluczowym RECOVER, określają miejsce, w którym zapisano program obsługi rozkazu RECOVER.

Opcja 1: - Odzyskiwanie pojedynczego zbioru.

Wyboru opcji 1 dokonuje się poprzez wyspecyfikowanie w linii komendy rozkazu RECOVER parametru 'nazwa'. Wraz z parametrami: 'dl:', 'ścieżka', '.rozszerzenie' określa on zbiór, który ma zostać odzyskany. Jeśli w linii komendy pominięto parametry 'dl:' i 'ścieżka', to domyślnie zostaną przyjęte odpowiednio urządzenie robocze i katalog roboczy. Odzyskany zbiór może mieć różną długość od zbioru 'oryginalnego', ponieważ:

- znak końca zbioru mógł znajdować się w obszarze uszkodzonego nośnika magnetycznego i nie został odzyskany, a do zbioru wynikowego dopisano informacje zawartą poza faktycznym końcem zbioru,
- w czasie odzyskiwania opuszczono te fragmenty zbioru, które były zapisane w miejscu, w którym nośnik magnetyczny uległ uszkodzeniu.

Opcja 2: - Odzyskiwanie całej informacji zapisanej na dysku (dyskiecie).

Wybór opcji 2 zostaje dokonany poprzez wyspecyfikowanie w linii komendy rozkazu RECOVER tylko parametru 'dl:', określającego urządzenie, z załadowym nośnikiem magnetycznym zawierającym zbiory do odzyskiwania. W tym przypadku odzyskiwanie zbiorów polega na:

- kolejnym czytaniu informacji zawartej we wszystkich kolejnych sektorach dysku lub dyskietki,
- eliminowaniu informacji z tych sektorów, które zostały uszkodzone,
- grupowaniu informacji zapisanej w ciągu kolejnych, nieuszkodzonych sektorów w pojedyncze zbiory, zapisywane następnie w katalogu podstawowym pod nazwami:

FILEnnnn.REC

gdzie nnnn jest kolejną liczbą dziesiętną, od 0001.

W tym przypadku opisane postępowanie może być utrudnione przez ograniczoną ilość zbiorów, które można zapisać w katalogu podstawowym. Rozwiązaniem jest więc wykonywanie odzyskiwania etapami:

- odzyskanie takiej ilości informacji, ile zbiorów FILEnnnn.REC można wpisać do katalogu podstawowego,

- wykonanie weryfikacji odzyskanej informacji pod kątem jej dalszej użyteczności, co potrzebne skopiować na inny nośnik, co zbędne skasować,
- powtarzać powyższe czynności, aż do odzyskania wszystkich niezbędnych informacji.

Opcja 2 rozkazu RECOVER jest przydatna szczególnie wtedy, gdy uszkodzeniu uległ ten fragment nośnika, na którym zapisano katalog. Niestety, rozkaz RECOVER nie posiada możliwości oceny dalszej przydatności poszczególnych informacji, dlatego w obu jego wariantach odzyskane zbiory muszą być poddane szczegółowej edycji, wykonywanej przez użytkownika. W zasadzie rozkaz RECOVER nadaje się przede wszystkim do odzyskiwania zbiorów tekstowych i programów źródłowych, napisanych w językach symbolicznych, które łatwo można modyfikować. W obu wariantach zastosowania, rozkaz RECOVER oznacza na dysku lub dyskietce uszkodzone sektory, co zapobiega wpisywaniu do nich jakiegokolwiek informacji.

Przykłady:

Rozkaz RECOVER o następującej postaci:

A>RECOVER A:MYPROG.TXT

spowoduje odzyskanie (odtworzenie) zbioru MYPROG.TXT, zapisanego w katalogu, roboczym na dyskietce założonej na pierwszą stację dysków miękkich A:.

Rozkaz RECOVER o następującej postaci:

A>RECOVER A:

spowoduje odzyskanie (odtworzenie) informacji zapisanej w nieuszkodzonych sektorach, na dyskietce założonej na pierwszą stację dysków miękkich A:. Odtworzenie będzie polegało na wyeliminowaniu informacji zapisanej w uszkodzonych sektorach, pogrupowaniu danych z kolejnych nieuszkodzonych sektorów w pojedyncze zbiory FILEnnnn.REC, które zostaną następnie zapisane w katalogu podstawowym.

6.34. Rozkaz RENAME (lub REN).

Format:

RENAME [d:][ścieżka]nazwa[.rozszerzenie] nazwa[.rozszerzenie]

lub:

REN [d:][ścieżka]nazwa[.rozszerzenie] nazwa[.rozszerzenie]

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz RENAME zmienia nazwę pierwszego wyspecyfikowanego zbioru (stara nazwa) na drugą nazwę (nowa nazwa) na tym samym urządzeniu, określonym parametrem 'd:'. Dopuszczalna jest zmiana nazw lub rozszerzeń całych grup zbiorów (przy użyciu znaków specjalnych '*' i '?').

Przykłady:

Rozkaz RENAME o następującej postaci:

RENAME A:ABC.DAT ABC.BAK

spowoduje zmianę nazwy zbioru ABC.DAT na ABC.BAK na urządzeniu A:.

Rozkaz RENAME o następującej postaci:

REN TEST.BAK ABC.DAT

spowoduje zmianę nazwy zbioru TEST.BAK na ABC.DAT na urządzeniu roboczym.

6.35. Rozkaz REPLACE (tylko w wersji PC DOS 3.20).

Format:

[d:][ścieżka]REPLACE [d1:][ścieżka1]nazwa[.rozszerzenie]
[d2:][ścieżka2][A][P][R][S][W]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz REPLACE pozwala na selekcyjne zastępowanie zbiorów, zapisanych w katalogu przeznaczenia, zbiorami o tych samych nazwach zapisanych w katalogu źródłowym i na dopisywanie w katalogu przeznaczenia zbiorów istniejących tylko w katalogu źródłowym. Katalogi źródłowy i przeznaczenia mogą znajdować się na różnych urządzeniach.

Poszczególne parametry występujące w linii komendy rozkazu REPLACE mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym REPLACE, określa miejsce, w którym znajduje się program obsługi rozkazu REPLACE,

[d1:][ścieżka1]nazwa[.rozszerzenie] określa położenie, nazwę i rozszerzenie zbioru (zbiorów) źródłowego. W specyfikacji źródła można używać znaków specjalnych '*' i '?',

[d2:][ścieżka2] określa urządzenie i katalog przeznaczenia. Wartością domyślną tego parametru jest zawsze urządzenie i katalog roboczy,

klucz /A wymusza kopiowanie tylko tych zbiorów z katalogu źródłowego, które nie istnieją w katalogu przeznaczenia. Klucz /A nie może jednocześnie występować z kluczem /S,

klucz /P wymusza konwersacyjny tryb pracy rozkazu REPLACE. Oznacza to, że przed podjęciem dopisywania lub zastępowania każdego zbioru w katalogu przeznaczenia, użytkownik musi zaakceptować działanie rozkazu REPLACE w odniesieniu do tego zbioru,

klucz /R wymusza zastępowanie w katalogu przeznaczenia tylko tych zbiorów, które mają ustawiony atrybut RO (Read Only - tylko odczyt),

klucz /S wymusza zastępowanie w katalogu przeznaczenia tylko tych zbiorów, które jednocześnie istnieją w obu katalogach. Klucz /S nie może jednocześnie występować z kluczem /A,

klucz /W powoduje zawieszenie wykonywania rozkazu REPLACE związane z wymianą dyskietki. Komunikat, wyświetlany na ekranie monitora, zależy wówczas od tego, czy wyspecyfikowano dodatkowo klucz /A:

- jeśli tak, to wyświetlony zostaje komunikat:

Press any key to replacing file(s)

- jeśli nie, to wyświetlony zostaje komunikat:

Press any key to begin adding file(s)

W zależności od wyników operacji wykonanych rozkazem REPLACE, system operacyjny PC DOS ustawia parametr ERRORLEVEL instrukcji warunkowej IF, wykonywanej w zbiorach rozkazów przetwarzania wsadowego, na następujące wartości:

- 2 - nie odnaleziono zbioru,
- 3 - nie odnaleziono ścieżki,
- 5 - zablokowany dostęp do zbioru (można próbować wykonać rozkaz REPLACE z kluczem /R),
- 8 - za mała pamięć operacyjna,
- 11 - błędna specyfikacja linii komendy rozkazu REPLACE,
- 15 - określono błędne urządzenie,
- 22 - nieznana komenda (wersja systemu operacyjnego nie zawiera wywołanego rozkazu).

Przykład:

Rozkaz REPLACE o następującej postaci:

```
REPLACE C:\EDITORS\FILE1.TEX C:\ /S/P
```

spowoduje zastąpienie zbioru o nazwie ZBIOR1.TEX, który wpisano uprzednio do katalogu podstawowego, zbiorem o tej samej nazwie, zapisanym w katalogu podrzędnym EDITORS. Operacja będzie wykonywana na urządzeniu C:. Jeśli zbiór FILE1.TEX nie istnieje w katalogu podstawowym, to nie zostanie do niego dopisany. Klucz /P wymusza konwersacyjny tryb wykonywania rozkazu REPLACE.

6.36. Rozkaz RESTORE.

Format:

```
[d:][ścieżka]RESTORE d1: [d2:][ścieżka2]nazwa[.rozszerzenie]
[/S][/P]
```

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz RESTORE odtwarza zbiory z dysków, na których, przy pomocy rozkazu BACKUP, zostały utworzone ich kopie zapasowe.

Poszczególne parametry występujące w linii komendy rozkazu RESTORE mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym RESTORE, określa miejsce, w którym znajduje się program obsługi rozkazu RESTORE,

[d1:] określa, na jakim urządzeniu znajdują się kopie zapasowe,

[d2:][ścieżka2]nazwa[.rozszerzenie] określają gdzie oraz który zbiór (zbiory) mają zostać odtworzone na podstawie kopii zapasowej,

klucz /S powoduje odtwarzanie wszystkich zbiorów zapisanych w katalogach podrzędnych w stosunku do wyspecyfikowanego katalogu (odtworzenie obejmuje tylko katalogi podrzędne, bez wyspecyfikowanego katalogu),

klucz /P wymusza konwersacyjny tryb pracy rozkazu RESTORE. System operacyjny informuje które zbiory zostały zmienione przez ostatni rozkaz BACKUP oraz które z nich posiadają atrybut RO (Read Only - tylko odczyt), pozostawiając użytkownikowi decyzję: odtwarzać czy nie odtwarzać.

W systemie operacyjnym MS DOS wersja 3.10 występują dodatkowo następujące klucze:

klucz /B:mm-dd-yy (lub: /D:dd-mm-yy lub: /D:yy-mm-dd, zależnie od formatu daty ustawionego instrukcją COUNTRY lub rozkazem SELECT) wymusza odtwarzanie tylko tych zbiorów, które zostały utworzone lub zmodyfikowane do wyspecyfikowanej daty,

klucz /A:mm-dd-yy (lub: /D:dd-mm-yy lub: /D:yy-mm-dd, zależnie od formatu daty ustawionego instrukcją COUNTRY lub rozkazem SELECT) wymusza odtwarzanie tylko tych zbiorów, które zostały utworzone lub zmodyfikowane od wyspecyfikowanej daty,

klucz /E:hh:mm:ss wymusza odtwarzanie tylko tych zbiorów, które zostały utworzone lub zmodyfikowane do wyspecyfikowanego czasu,

klucz /M wymusza odtwarzanie tylko tych zbiorów, które modyfikowano od chwili wykonania ostatniej kopii zapasowej tego zbioru (ang. backup),

klucz /N wymusza odtwarzanie tylko tych zbiorów, które utworzono później niż odpowiadające im zbiory, znajdujące się na urządzeniu d2:.

Uwagi dotyczące posługiwania się rozkazem RESTORE:

1. Kopie zapasowe zbiorów odtwarzanych muszą być wykonane rozkazem BACKUP.
2. Jeśli nie podano parametru 'ścieżka2', to odtwarzane będą zbiory z katalogu roboczego (koniecznie trzeba podać nazwy tych zbiorów). Zbiory odtwarzane muszą znajdować się w tym samym katalogu co ich kopie zapasowe.
3. Dla określenia nazw i rozszerzeń zbiorów odtwarzanych można używać znaków specjalnych '*' i '?'.
.
4. Jeśli na dyskietce znajdującej się w urządzeniu d1: brak

jest kopii zapasowej zbioru, który ma zostać odtworzony, to system operacyjny odpowiednimi komunikatami będzie sterował wymianą dyskielek. Wymiana taka będzie miała miejsce także wtedy, gdy dany zbiór, przy wykonywaniu kopii zapasowej, został przez rozkaz BACKUP podzielony i zapisany na dwóch (lub więcej) różnych dyskielekach.

5. W przypadku, gdy dane zbiory są jednocześnie wykorzystywane przez inne zadania (ang. sharing files - patrz rozkaz SHARE), to odtwarzaniu można poddawać tylko te zbiory, do których możliwy jest dostęp (nie są używane przez żadne zadanie).
6. W przypadku odtwarzania zbiorów systemowych należy używać opcji z kluczem /P dla uniknięcia odtwarzania zbiorów pochodzących ze starych wersji systemu operacyjnego.

W zależności od wyniku operacji odtwarzania system operacyjny ustawia parametr ERRORLEVEL instrukcji warunkowej IF (przetwarzanie wsadowe) na następujące wartości:

- 0 - poprawne zakończenie operacji odtwarzania,
- 1 - nie odnaleziono zbioru do odtworzenia,
- 2 - niektóre zbiory nie mogą być odtworzone wskutek zaistniałego konfliktu podziału (ang. sharing, nie występuje w wersji MS DOS 3.10),
- 3 - operacja odtwarzania przerwana przez użytkownika,
- 4 - operacja odtwarzania przerwana wskutek błędu.

Przykłady:

Rozkaz RESTORE o następującej postaci:

```
A>RESTORE A: C:\LEVEL *.TXT/P
```

spowoduje, że przed odtworzeniem kolejnego zbioru z rozszerzeniem .TXT znajdującego się w katalogu LEVEL na urządzeniu A:, system operacyjny zgłosi się, umożliwiając użytkownikowi podjęcie decyzji odnośnie odtwarzania kolejnych zbiorów,

Rozkaz RESTORE o następującej postaci:

A>RESTORE A: C:\VS

spowoduje odtworzenie zbiorów we wszystkich katalogach z urządzenia A: na urządzenie C:.

6.37. Rozkaz RMDIR (Remove Directory).

Format:

RMDIR [d:]ścieżka

lub:

RD [d:]ścieżka

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz RMDIR pozwala usunąć wyspecyfikowany katalog podrzędny. Możliwe jest także usunięcie katalogu podrzędnego z katalogu o kilka poziomów wyższego. Wówczas usunięty zostanie ostatni katalog w wyspecyfikowanej ścieżce-do-zbioru. Usuwane katalogi nie mogą zawierać żadnych zbiorów ani innych katalogów za wyjątkiem .<DIR> i ..<DIR>. Rozkazem RMDIR nie można usunąć katalogu podstawowego (ang. root directory) i katalogu roboczego. Podczas posługiwania się rozkazem RMDIR należy zachować szczególną ostrożność, jeśli przedtem zainicjowano wykonanie rozkazów: ASSIGN lub JOIN. Rozkaz RMDIR nie pozwala także na usuwanie katalogów poprzez symboliczne odwoływanie się do nich, zainicjowane rozkazem SUBST.

Przykłady:

Rozkaz RMDIR o następującej postaci:

A>RMDIR C:LEVEL1

spowoduje usunięcie katalogu podrzędnego LEVEL1 z katalogu roboczego na urządzeniu C:.

Rozkaz RMDIR o następującej postaci:

A>RD \LEVEL1\LEVEL2

spowoduje usunięcie katalogu podrzędnego LEVEL2 z katalogu LEVEL1 na urządzeniu roboczym A:.

6.38. Rozkaz SELECT (nie występuje w wersji MS DOS 3.10).

Format:

[d:][ścieżka]SELECT [[A:|B:] d1:[ścieżka]] xxx yy

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz SELECT pozwala na tworzenie kopii dyskietek z systemem operacyjnym PC DOS, zawierających nowy, wybrany przez użytkownika format wprowadzania daty i czasu oraz nowy układ klawiatury. Kopie systemu operacyjnego można wykonywać tylko na nowych dyskietkach lub takich, z których informacja może ulec skasowaniu, gdyż zainicjowanie wykonania tego rozkazu wymazuje dotychczas zapisane informacje.

Poszczególne parametry występujące w linii komendy rozkazu SELECT mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym SELECT, określa miejsce, w którym znajduje się program obsługi rozkazu SELECT,

[A:|B:] określa stację dyskietkowa, na którą założono dyskietkę z systemem operacyjnym PC DOS, przeznaczonym do skopiowania. Wolno tutaj podawać tylko urządzenie A:

lub urządzenie B:. W razie pominięcia tego parametru domyślnie zostanie przyjęte urządzenie A:.

dl:[ścieżka] określa stację dyskietską i katalog, do którego zostaną skopiowane programy obsługi poszczególnych rozkazów systemu operacyjnego PC DOS. W razie pominięcia parametru 'dl:', domyślnie zostanie przyjęta stacja B:. Parametr 'dl:' nie może określać urządzenia z założoną dyskietską zawierającą system operacyjny przeznaczony do skopiowania (nie można kopiować dyskietek 'na siebie'),

xxx określa kod kraju, który identyfikuje nowy format daty i czasu, oraz separator. Kod identyfikujący kraj można określić na podstawie poniższej tabelki,

yy określa kod układu klawiatury. Parametr 'yy' można określić na podstawie poniższej tabelki.

Parametry 'xxx' i 'yy' rozkazu SELECT:

KRAJ	KOD KRAJU (xxx)	KOD KLAWIATURY (yy)	KRAJ	KOD KRAJU (xxx)	KOD KLAWIATURY (yy)
Australia	061	(*)	kraje Środ-	786	(*)
Belgia	032	(*)	kowego Wsch.		
Kanada	002	(*)	Holandia	031	(*)
(franc.)			Norwegia	047	(*)
Dania	045	(*)	Portugalia	351	(*)
Finlandia	358	(*)	Hiszpania	034	SP
Francja	033	FR	Szwecja	046	(*)
RFN	049	GR	Szwajcaria	041	(*)
Włochy	039	IT	W. Brytania	044	UK
Izrael	972	(*)	USA	001	US

(*) - program obsługi tego układu klawiatury może znajdować się poza dyskietską systemową.

Rozkaz SELECT współpracuje z rozkazami FORMAT i XCOPY (inicjuje wykonanie tych rozkazów). W przypadku wykonywania kopii systemu na dysku twardym, wskazane jest formatowanie z kluczem /V, co umożliwia wprowadzenie etykiety dysku. W przypadku odwoływania się do katalogów, w specyfikacji rozkazu XCOPY, należy zawsze podawać pełną ścieżkę-do-zbioru, a nie używać odwołań . lub ..

Sesja pracy z programem obsługi rozkazu SELECT:

1. Właściwe założenie dyskietek na obie stacje (w przypadku kopiowania na dysk twardy - założenie tylko dyskietki z systemem operacyjnym).
2. Zainicjowanie rozkazu SELECT z właściwymi parametrami. Na ekranie monitora pojawia się następujący komunikat:

SELECT is used to install DOS the first
time. SELECT erases everything on the
specified target and then installs DOS.
Do you want to continue (Y/N)?

Odpowiadając 'Y' użytkownik zezwala na rozpoczęcie wykonywania rozkazu SELECT, a więc i na zaformatowanie dyskietki wyjściowej (co niszczy całą zapisaną na niej informację).

3. Dyskietka systemowa może nie zawierać właściwego programu KEYyy.COM obsługującego klawiaturę o zadanym układzie (pod warunkiem, że nie jest to układ klawiatury obowiązujący w USA). W takim przypadku system wyświetla następujący komunikat:

Is KEYByy.COM on another
diskette (Y/N)?

- odpowiedź 'N' powoduje wyświetlenie kolejnego komunikatu:

Keyboard routine not found

i zakończenie wykonywania rozkazu SELECT.

- odpowiedź 'Y' zawiesza dalsze wykonywanie rozkazu SELECT z następującym komunikatem:

Insert KEYByy.COM diskette in drive d1:
And strike any key when ready

Po założeniu właściwej dyskietki i wciśnięciu klawisza dowolnej litery, rozkaz SELECT kopiuje zbiór KEYByy.COM, a następnie:

- tworzy na dyskietce wyjściowej zbiór CONFIG.SYS zawierający instrukcję:

COUNTRY=xxx

- tworzy na dyskietce wyjściowej zbiór AUTOEXEC.BAT o następującej postaci:

```
PATH\;[\path;]  
KEYByy  
ECHO OFF  
CLS  
DATE  
TIME  
VER
```

Dla dysku twardego rozkaz PATH zawiera oznaczenia identyfikujące to urządzenie. Jeśli parametr kodu układu klawiatury określa układ obowiązujący w USA, to instrukcja KEYByy nie jest umieszczana w zbiorze AUTOEXEC.BAT.

Przykład:

Rozkaz SELECT o następującej postaci:

```
C>A:SELECT A: B:/USER 034 FR
```

spowoduje skopiowanie systemu operacyjnego z dyskietki założonej na stację A: na dyskietkę założoną na stację B:, przy czym zbiory systemowe na dyskietce w stacji B: zostaną zapisane w katalogu USER, a w skopiowanym systemie operacyjnym będzie obowiązywał hiszpański format daty i czasu oraz francuski układ klawiatury. Program obsługi rozkazu SELECT znajduje się na urządzeniu A:, w katalogu roboczym.

6.39. Rozkaz SET (Set Environment).

Format:

```
SET [name=[parameter]]
```

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz SET pozwala kształtować tzw. otoczenie programisty, a więc określa ciąg znakowy opisujący stan obowiązujący w czasie pracy systemu operacyjnego PC DOS (termin 'stan' nie jest tutaj najzręczniejszym określeniem). Mogą nim być: obecność w zestawie sprzętowym koprocessora arytmetycznego 8087, aktywność standardowego interpretera rozkazów COMMAND.COM, sposób zgłaszania się systemu operacyjnego, dostęp do zbiorów bibliotecznych z różnych katalogów etc. Wykonanie rozkazu SET polega na wstawieniu, zmodyfikowaniu lub usunięciu ze specjalnego, 127-bajtowego obszaru pamięci operacyjnej, zarezerwowanej dla opisu otoczenia programisty, ciągu znakowego określającego dany stan. Argument 'name' określa nazwę (rodzaj) stanu, argument 'parameter' jego opis, przy czym oba argumenty muszą być odseparowane od siebie znakiem równości (=). Np. ciąg znaków:

```
COMSPEC=C:\COMMAND.COM
```

gdzie argument COMSPEC oznacza 'aktywny interpreter rozkazów' a argument C:\COMMAND.COM określa zbiór nadający się do bezpośredniego uruchamiania, system operacyjny PC DOS interpretuje jako: 'aktywny interpreter rozkazów nosi nazwę COMMAND.COM i znajduje się w katalogu podstawowym na urządzeniu C:'.

Wspomniany ciąg znakowy można wyspecyfikować zarówno małymi jak i dużymi literami, ale przed wpisaniem do pamięci wszystkie małe litery zostaną automatycznie zamienione na duże.

Rozkaz SET bez żadnych argumentów spowoduje wyświetlenie całej tablicy zawierającej opis otoczenia programisty. Przykładowy taki opis przedstawiono poniżej:

```
COMSPEC=C:\COMMAND.COM
PATH=C:\;C:\UTILITIE;C:\DOS;C\MOKEY
PROMPT=$P$G
```

Jeśli w linii komendy rozkazu SET wyspecyfikowano tylko argument 'name' ze znakiem równości, ale bez argumentu 'parameter':

```
SET name=
```

to określony argumentem 'name' ciąg znaków zostanie usunięty z obszaru pamięci zawierającej opis otoczenia programisty.

Opis otoczenia programisty jest dostępny dla wszystkich rozkazów systemu operacyjnego PC DOS oraz wszystkich programów użytkowych. Do tego opisu są automatycznie dołączane parametry, z którymi zainicjowano rozkazy PROMPT i PATH, toteż w ich przypadku otoczenie programisty nie musi być kształtowane dodatkowymi rozkazami SET. Ponadto podczas restartu systemu operacyjnego PC DOS do tego opisu zostaje automatycznie dołączony ciąg znakowy:

COMSPEC=parameter

gdzie 'parameter' określa nazwę i miejsce zapisania interpretera rozkazów (zwykle jest to zbiór COMMAND.COM, zapisany w katalogu podstawowym na urządzeniu A: lub C:). Otoczenie programisty kształtują także rozkazy MODE, PRINT i GRAPHIC, pozostające w pamięci operacyjnej do czasu ponownego restartu systemu operacyjnego PC DOS. W przypadku tego typu rozkazów, system operacyjny rozszerza rozmiar tablicy, zawierającej opis otoczenia systemu poza początkowe 127 bajtów, z tym, że jest to rozszerzenie 'jedenrazowe', a każda następna próba wpisania kolejnego ciągu znaków do tej tablicy kończy się komunikatem:

Out of environment space

Przykłady:

Rozkaz SET o następującej postaci:

A>SET PGMS=\LEVEL1

wpisuje ciąg PGMS=\LEVEL1 do tablicy opisującej otoczenie programisty. Pozwala to odwoływać się w programie użytkowym do katalogu LEVEL1 poprzez parametr PGMS.

Rozkaz SET o następującej postaci:

A>SET PGMS=

usuwa z tablicy opisującej otoczenie programisty ciąg z. akowy, wyspecyfikowany w poprzednim przykładzie.

6.40. Rozkaz SHARE.

Format:

[d:][ścieżka]SHARE [/F:filesize][/L:locks]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz SHARE umożliwia jednoczesne wykorzystywanie pojedynczego zbioru (zbiorów) przez różne zadania (programy, rozkazy) pracujące w systemie operacyjnym PC DOS, a także zapobiega wymianie dyskiek, jeśli którekolwiek z zadań pracujących w systemie nie zakończyło wykonywania swoich operacji zapisu/odczytu na tej dyskiecie.

Poszczególne parametry występujące w linii komendy rozkazu SHARE mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym SHARE, określa miejsce, w którym znajduje się program obsługi rozkazu SHARE,

klucz /F:filesize pozwala na zarezerwowanie bloku pamięci operacyjnej, potrzebnego do rejestrowania informacji niezbędnych do obsługi jednocześnie wykorzystywanych zbiorów. Rozmiar tego bloku określa parametr 'filesize', którego wartość domyślna wynosi 2048 bajtów. Każdy otwarty (czyli aktualnie wykorzystywany) zbiór wymaga zarezerwowania bloku pamięci, o długości równej ilości znaków określającej nazwę tego zbioru plus 11 dodatkowych bajtów.

klucz /L:lock określa liczbę zadań mogących jednocześnie wykorzystywać pojedynczy zbiór. Wartość domyślna parametru 'lock' wynosi 20.

Rozkaz SHARE można inicjować tylko raz od chwili ostatniego restartu systemu operacyjnego, o ile nie został zainicjowany podczas tego restartu. Próba wykonania kolejnej inicjacji kończy się wyświetleniem następującego komunikatu:

Share Already Installed

Program obsługi rozkazu SHARE posiada możliwość rejestracji otwierania stacji dyskietkowych. Jeśli w danej stacji wymieniono dyskietkę przed zakończeniem wykonywania zadania korzystającego z niej, to system operacyjny PC DOS, z zainicjowanym rozkazem SHARE, zawsze, przed wykonaniem kolejnej operacji z tą dyskietką, sprawdza jej etykietę. W przypadku, gdy nie jest to właściwa dyskietka, zostanie zawieszona wykonywanie na niej wszelkich czynności zapisu/odczytu i system zażąda jej wymiany.

Zainicjowanie rozkazu SHARE powoduje sprawdzenie tablic z blokami kontrolnymi zbiorów (FCBs). Jeśli w rozkazie CONFIG.SYS zainicjowano instrukcję FCBS=4,0 (domyślne wartości parametrów instrukcji FCBS), to tablice te zostaną zmodyfikowane tak, jak po zainicjowaniu instrukcji FCBS=16,8 (patrz opis instrukcji FCBS w rozdziale pt. "KONFIGUROWANIE SYSTEMU OPERACYJNEGO PC DOS").

6.41. Rozkaz SORT (Sort Filter).

Format:

[d:][ścieżka]SORT[/R][/+n]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz (filtr) SORT pozwala na odczytywanie danych ze standardowego urządzenia wejściowego, porządkowania ich według wzrastających lub malejących (domyślnie według wzrastających) kodów ASCII, a następnie zapisywania na standardowym urządzeniu wyjściowym.

Poszczególne parametry występujące w linii komendy rozkazu SORT mają następujące znaczenie:

[d:]ścieżka, przed słowem kluczowym SORT, określa miejsce, gdzie znajduje się program obsługi rozkazu SORT,

klucz /R powoduje porządkowanie danych według malejących kodów ASCII (tzn. w kolejności od Z do A),

klucz /+n wymusza porządkowanie od n-tej kolumny. Parametr n może zmieniać się w granicach od 1 do 63, wartość domyślna wynosi 1.

Uwagi dotyczące posługiwania się rozkazem SORT:

1. Nazwa zbioru wyjściowego musi być różna od nazwy zbioru źródłowego.
2. Podczas wykonywania porządkowania duże litery A-Z odpowiadają małym literom a-z i na odwrót.
3. Porządkowanie znaków o kodach ASCII większych niż 127 odbywa się na nie podstawie wartości tych kodów, lecz na podstawie odpowiadających im wartościom według zamieszczonej tabeli:

Wartość ASCII	Waga	Wartość ASCII	Waga	Wartość ASCII	Waga
128	67	145	65	162	79
129	85	146	65	163	85
130	69	147	79	164	78
131	65	148	79	165	78
132	65	149	79	166	166
133	65	150	85	167	167
134	65	151	85	168	63
135	67	152	89	169	169
136	69	153	79	170	170
137	69	154	85	171	171
138	69	155	36	172	172
139	73	156	36	173	33
140	73	157	36	174	34
141	73	158	36	175	34
142	65	159	36	225	83
143	65	160	65		
144	69	161	73		

Przykłady:

Rozkaz SORT o następującej postaci:

A>SORT /R ABC.DAT

spowoduje, że zbiór o nazwie ABC.DAT będzie odczytywany, porządkowany według malejących kodów ASCII i wyświetlany na ekranie monitora.

Rozkaz SORT o następującej postaci:

A>SORT /R <UNSORT.TXT >SORT.TXT

spowoduje, że zbiór UNSORT.TXT będzie odczytywany, porządkowany według malejących kodów ASCII, a następnie zapisywany na urządzeniu roboczym pod nazwą SORT.TXT.

Rozkaz SORT o następującej postaci:

A>DIR : SORT /+14

spowoduje, że wykonywany jest rozkaz DIR listowania zawartości katalogu roboczego, ale nazwy poszczególnych zbiorów nie są wyświetlane w kolejności ich zapisu w katalogu roboczym, lecz według rozmiarów tych zbiorów (kolumna 14 zawiera rozmiary poszczególnych zbiorów). Pokazany przykład realizuje tzw. przetwarzanie potokowe (ang. piping).

6.42. Rozkaz SUBST (Substitute).

Format:

[d:] [ścieżka] SUBST d1: d2: ścieżka2

lub:

[d:] [ścieżka] SUBST d1: /D

lub:

[d:] [ścieżka] SUBST

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz SUBST przyporządkowuje nazwie urządzenia i ścieżce do-zbioru symbol literowy, odpowiadający oznaczeniu innego urządzenia. Pozwala to na odwoływanie się w innych rozkazach systemu operacyjnego PC DOS do tego urządzenia i katalogu poprzez zdefiniowany symbol.

Poszczególne parametry występujące w linii komendy rozkazu SUBST mają następujące znaczenie:

[d:] [ścieżka], przed słowem kluczowym SUBST, określa miejsce, w którym znajduje się program obsługi rozkazu SUBST,

d1: określa symbol literowy urządzenia, definiowany rozkazem SUBST,

d2:ścieżka2 określa urządzenie i ścieżkę-do-zbioru, której zostanie przyporządkowany symbol d1:,

klucz /D unieważnia wykonaną rozkazem SUBST operację przyporządkowania.

Parametr 'd1:' może być literą z zakresu zdefiniowanego instrukcją LASTDRIVE (patrz rozdział dotyczący konfigurowania systemu operacyjnego). Rozkaz SUBST, wykonany bez parametrów, wyświetla aktualny stan przyporządkowania.

Uwagi dotyczące posługiwania się rozkazem SUBST:

1. Nazwy obu urządzeń określonych parametrami 'd1:', 'd2:' muszą być różne, a ponadto parametr 'd1:' musi być różny od nazwy aktualnego urządzenia roboczego.
2. Określając ścieżkę-do-zbioru (parametr 'ścieżka2') należy zawsze rozpoczynać od katalogu podstawowego.
3. Szczególną uwagę należy zachować w czasie odwoływania się, poprzez symbol zdefiniowany rozkazem SUBST, w następujących rozkazach: CHDIR, MKDIR, RMDIR i PATH.
4. Rozkazy: ASSIGN, BACKUP, DISKCOMP, DISKCOPY, FDISK, FORMAT, JOIN, LABEL i RESTORE ignorują odwoływanie się poprzez symbol zdefiniowany rozkazem SUBST.
5. Jeśli wyspecyfikowano niedozwolony symbol urządzenia 'd1:' (np. spoza zakresu określonego instrukcją LASTDRIVE, oznaczający urządzenie robocze etc.), to na ekranie monitora wyświetlony zostanie następujący komunikat:

Invalid Parameter

Ten sam komunikat pojawi się w chwili wywoływania rozkazu SUBST z kluczem /D bez określenia parametru 'd1:'.

6. Jeśli w linii komendy rozkazu SUBST zostanie wyspecyfikowana, nie istniejąca na urządzeniu d2:, ścieżka-do-zbioru, wówczas na ekranie monitora zostanie wyświetlony następujący komunikat:

Invalid Path .

Przykład:

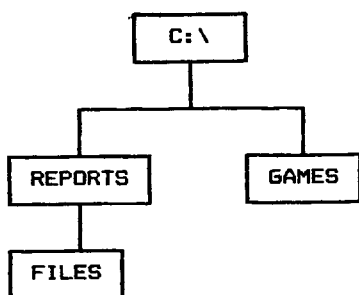
Niech:

- urządzeniem roboczym jest urządzenie B:,
- katalogiem roboczym na urządzeniu C: jest:

`C:\REPORTS\FILES`

- restart systemu operacyjnego wykonano ze zbiorem `CONFIG.SYS`, zawierającym instrukcję `LASTDRIVE=H`

Ponadto niech drzewo katalogowe na urządzeniu C: będzie miało następującą postać:



Jeśli wykonano następujący rozkaz `SUBST`:

`B>SUBST G: C:\REPORTS\FILES`

to aby np. wylistować zawartość katalogu `FILES` można użyć dwóch równoważnych rozkazów:

`B>DIR C:\REPORTS\FILES`

lub:

`B>DIR G:`

Dla wyświetlenia aktualnego stanu przyporządkowania należy wykonać następujący rozkaz `SUBST`:

B>SUBST

Wówczas komunikat będzie miał następującą postać:

G:=> C:\REPORTS\FILES

Dla usunięcia tego przyporządkowania musi zostać użyty rozkaz SUBST o postaci:

B>SUBST G:/D

6.43. Rozkaz SYS (System).

Format:

[d:][ścieżka]SYS d1:

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz SYS pozwala na przepisanie dwóch ukrytych zbiorów systemowych (ang. hidden files): IBMDOS.COM i IBMBIO.COM z dysku systemowego na dysk lub dyskietkę określoną parametrem 'd1:'. Parametry: 'd:' i 'ścieżka', przed słowem kluczowym SYS, określają miejsce, w którym znajduje się program obsługi rozkazu SYS.

Dla wykonania rozkazu SYS musi być spełniony jeden z poniższych warunków:

- albo katalog podstawowy na dysku d1: jest całkowicie pusty,
- albo dysk d1: został właśnie zaformatowany rozkazem FORMAT z kluczem /S lub /B (a więc katalog systemowy jest także pusty).

Spełnienie jednego z powyższych zadań jest warunkowane tym, że zbiory IBMBIO.COM i IBMDOS.COM mogą być kopiowane tylko w określone miejsca w pamięci masowej (dysk, dyskietka), co umożliwia ich odzyskanie podczas restartu systemu operacyjnego przez program BOOTSTRAP (patrz rozdział pt. 'STRUKTURA SYSTEMU OPERACYJNEGO PC DOS'). Rozkaz SYS nie kopiuje całego systemu operacyjnego. Zbiór COMMAND.COM musi zostać 'indywidualnie' przepisany rozkazem COPY do katalogu podstawowego na dyskietkę (dysk) na urządzeniu d1:.

Przykład:

Rozkaz SYS o następującej postaci:

```
C>SYS B
```

spowoduje przepisanie zbiorów systemowych na dyskietkę założoną na drugą stację dysków miękkich B:.

6.44. Rozkaz TIME.

Format:

```
TIME [hh:mm[:ss[.xx]]]
```

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz TIME pozwala na wprowadzenie lub zmianę czasu systemowego. Czas może zostać zmieniony poprzez wywołanie rozkazu TIME zarówno z klawiatury jak i ze zbioru rozkazów przetwarzania wsadowego.

Poszczególne parametry występujące w linii komendy rozkazu TIME mają następujące znaczenie:

hh określa godzinę. Godzina może być jedno lub dwucyfrową liczbą z zakresu od 0 do 23,

mm określa minutę. Minuta może być jedno lub dwucyfrową liczbą z zakresu od 0 do 59,

ss określa sekundę. Sekunda może być jedno lub dwucyfrową liczbą z zakresu od 0 do 59,

xx określa setną część sekundy. Setną częścią sekundy może być jedno lub dwucyfrowa liczba z zakresu od 0 do 99.

Uwagi dotyczące posługiwania się rozkazem TIME:

1. Jeśli rozkaz TIME zostanie wywołany bez parametrów, to na ekranie monitora wyświetlony zostanie następujący komunikat:

```
Current time is hh:mm:ss.xx  
Enter new time: _
```

po którym należy wyspecyfikować nowy czas, lub od razu wciśnąć klawisz 'Enter', sygnalizując rezygnację z wykonania zmiany czasu.

2. Separatorami pomiędzy godzinami, minutami i sekundami mogą być: dwukropek (:) lub kropka (.). Separatorami pomiędzy sekundami a setnymi częściami sekund mogą być: kropka (.) lub przecinek (,).
3. Jeśli czas został wyspecyfikowany poprawnie pod względem formalnym, to system operacyjny PC DOS zaakceptuje go, potwierdzając to jego wyświetleniem. W przeciwnym wypadku na ekranie monitora pojawi się następujący komunikat:

```
Invalid time  
Enter new time: _
```

4. Pominięcie w specyfikacji linii komendy rozkazu TIME sekund i setnych części sekund spowoduje, że system operacyjny domyślnie przyjmie za ich wartości zera.
5. Dla zmiany formatu wprowadzania czasu można użyć instrukcji COUNTRY, znajdującej się w zbiorze konfiguracyjnym system operacyjny CONFIG.SYS.
6. W mikrokomputerze IBM PC/AT zmiana czasu nie zawsze jest konieczna, nawet po przerwie spowodowanej wyłączeniem zasilania, gdyż w zestawie sprzętowym przewidziano baterijne zasilanie zegara.

- 6.45. Rozkaz TREE (Display Directory, nie występuje w wersji MS DOS 3.10).

Format:

[d:][ścieżka]TREE [d1:][/F]

Typ:

Rozkaz zewnętrzny.

Opis:

Rozkaz TREE wyświetla strukturę katalogów wyspecyfikowanego urządzenia i opcjonalnie listuje wszystkie zbiory zapisane w katalogu podstawowym i katalogach podrzędnych.

Poszczególne parametry występujące w linii komendy rozkazu TREE mają następujące znaczenie:

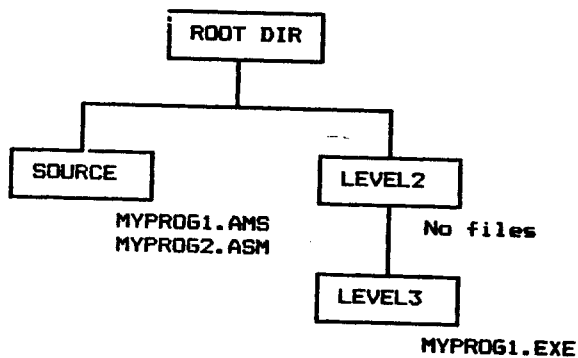
[d:][ścieżka], przed słowem kluczowym TREE, określa miejsce, gdzie znajduje się program obsługi rozkazu TREE,

d1: określa urządzenie, którego struktura katalogów zostanie wyświetlona,

klucz /F wymusza listowanie nazw zbiorów zapisanych w katalogu podstawowym i wszystkich katalogach podrzędnych.

Przykład:

Niech dyskietka na urządzeniu B: o etykiecie MYDISK posiada następujące drzewo katalogowe:



Aby zarejestrować tę strukturę pod nazwą Tree.LST należy wykonać następujący rozkaz TREE:

```
C>TREE B:/F >Tree.LST
```

przy założeniu, że program obsługi rozkazu TREE znajduje się na urządzeniu roboczym. Dla powyższego drzewa katalogowego rozkaz TREE utworzy następujący raport (zbiór Tree.LST):

DIRECTORY PATH LISTING FOR VOLUME MYDISK

Files: None

Path:\SOURCE

Subdirectories: None

Files: MYPROG1.AMS
MYPROG2.ASM

Subdirectories: LEVEL3

Path: \LEVEL2\LEVEL3

Subdirectories: None

Files: MYPROG1.EXE

Format:

TYPE [d:][ścieżka]nazwa[.rozszerzenie]

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz TYPE wyświetla na standardowym urządzeniu wyjściowym zawartość wyspecyfikowanego zbioru. Zbiory tekstowe, generowane przez większość formatorów i edytorów tekstowych, są wyświetlane w takiej postaci, w jakiej zostały zapisane, pomijając znaki sterujące (tabulacja itp.). Zbiory generowane przez kompilatory i assembly mogą nie nadawać się do wyświetlenia za pomocą rozkazu TYPE. Zawartość zbioru wyprowadzanego na monitor można na bieżąco drukować, korzystając z klawiszy funkcyjnych 'Ctrl', 'PrtSc'. Rozkaz TYPE nie dopuszcza stosowania znaków specjalnych '?' i '2'.

Przykład:

Rozkaz TYPE o następującej postaci:

TYPE C:\LEVEL1\ABC.DAT >PRN

spowoduje przesłanie do kolejki na drukarkę zawartości zbioru ABC.DAT, znajdującego się w katalogu LEVEL1 na urządzeniu C:.

6.47. Rozkaz VER (Version).

Format:

VER

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz VER wyświetla aktualnie załadowaną do pamięci wersję systemu operacyjnego PC DOS.

Przykład:

Rozkaz VER o następującej postaci:

A>VER

spowoduje wyświetlenie na ekranie monitora następującego komunikatu (przy założeniu, że w pamięci operacyjnej znajduje się wersja 3.20 systemu operacyjnego PC DOS):

IBM Personal Computer DOS Version 3.20.

6.48. Rozkaz VERIFY.

Format:

VERIFY [ON|OFF]

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz VERIFY umożliwia kontrolę poprawności zapisu na dyskach. Jeśli ustawiono VERIFY ON to system operacyjny po zapisaniu danych na dysk sprawdza wykonany zapis pod kątem jego bezbłędneho odczytu (sprawdzenie nie polega więc na porównywaniu z danymi źródłowymi, lecz na zachowaniu zasad zapisu na dysku). Stan VERIFY ON obowiązuje aż do wykonania rozkazu VERIFY OFF. Rozkaz VERIFY, bez podania argumentów, wyświetla aktualny stan ustawiony rozkazem VERIFY. Domyślnie system operacyjny ustawia VERIFY OFF. Wykonanie dodatkowej weryfikacji każdego zapisu powoduje, że system operacyjny pracuje wolniej, ale z mniejszym ryzykiem popełnienia błędu.

Przykład:

Ustawienie weryfikacji powoduje wykonanie następującego rozkazu VERIFY:

A>VERIFY ON

Aby wyświetlić aktualny stan weryfikacji należy wykonać następujący rozkaz:

A>VERIFY

Jego rezultatem jest komunikat:

VERIFY is on

6.49. Rozkaz VOL (Volume).

Format:

VOL [d:]

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz VOL wyświetla etykiety dysku znajdującego się na wyspecyfikowanym urządzeniu. Jeśli pominięto parametr d:, wówczas domyślnie zostanie wyświetlona etykieta dysku roboczego.

1. Postać komunikatu, jeśli dysk posiada etykietę:

Volume in drive d: is etykieta

2. Postać komunikatu, jeśli dysk nie posiada etykiety:

Volume in drive d: has no label.

6.50. Rozkaz XCOPY (tylko w wersji PC DOS 3.20).

Format:

[d:][ścieżka]XCOPY [d1:][ścieżka1][nazwa1[.rozszerzenie1]]
[d2:][ścieżka2][nazwa2[.rozszerzenie2]]
[/A][/D][/E][/M][/P][/S][/V][/W]

lub:

[d:][ścieżka]XCOPY [d1:][ścieżka1] [nazwa1[.rozszerzenie1]]
[d2:][ścieżka2][nazwa2[.rozszerzenie2]]
[/A][/D][/E][/M][/P][/S][/V][/W]

lub:

```
[d:][ścieżka]XCOPY d1:[ścieżka1][nazwa1[.rozszerzenie1]]
[d2:][ścieżka2][nazwa2[.rozszerzenie2]]
[/A][/D][/E][/M][/P][/S][/V][/W]
```

Typ:

Rozkaz zewnętrzny

Opis:

Rozkaz XCOPY pozwala na selekcyjne kopiowanie zbiorów i katalogów, które mogą zawierać także katalogi podrzędne.

Poszczególne parametry występujące w linii komendy rozkazu XCOPY mają następujące znaczenie:

[d:][ścieżka], przed słowem kluczowym XCOPY, określa miejsce, w którym zapisano program obsługi rozkazu XCOPY,

[d1:][ścieżka1][nazwa1[.rozszerzenie1]] określa to, co ma zostać skopiowane. Kopiować można pojedyncze zbiory, grupy zbiorów, całe katalogi oraz całą zawartość dysku lub dyskietki. Jeśli nie wyspecyfikowano parametrów 'nazwa1' i 'rozszerzenie1', to system operacyjny domyślnie przyjmie *.*,

[d2:][ścieżka2][nazwa2[.rozszerzenie2]] określa urządzenie, katalog oraz ewentualnie nazwę i rozszerzenie dla wykonywanej kopii. Kopiując katalogi i poszczególne zbiory można zmieniać ich nazwy.

Sposób wykonania rozkazu XCOPY mogą dodatkowo wymuszać następujące klucze:

klucz /A wymusza kopiowanie tylko tych zbiorów, które mają ustawiony tzw. bit archiwalny (atrybut +A, patrz rozkaz ATTRIB), przy czym w odróżnieniu od kopii wykonanej z kluczem /M, przepisany zbiór zachowuje wszystkie atrybuty zbioru źródłowego,

klucz /D wymusza kopiowanie tylko tych zbiorów, których data powstania (założenia, zapisania) jest taka sama lub późniejsza niż data wyspecyfikowana za kluczem /D. Format daty jest taki, jaki ustawiono instrukcją COUNTRY lub rozkazem SELECT:

/D:mm-dd-yy
lub: /D:mm-dd-yy
lub: /D:mm-dd-yy

klucz /E przy kopiowaniu zawartości całego katalogu spowoduje przepisanie także zawartych w nim pustych katalogów podrzędnych. Wykonanie rozkazu XCOPY bez klucza /E kopiuje tylko te katalogi podrzędne, które nie są puste (zawierają zbiory lub inne katalogi podrzędne),

klucz /M wymusza kopiowanie tylko tych zbiorów, które mają ustawiony tzw. bit archiwalny (atrybut +A, patrz rozkaz ATTRIB), przy czym w odróżnieniu od kopii wykonanej z kluczem /A, przepisany zbiór ma zgaszony (atrybut -A) bit archiwalny. Pozwala to na wykorzystywanie rozkazu XCOPY do tworzenia kopii zapasowych (ang. backup). Ustawiony bit archiwalny (atrybut +A) danego zbioru może wskazywać, że zbiór ten został utworzony lub zmodyfikowany od czasu ostatniego wykonania rozkazu XCOPY /M lub BACKUP /M,

klucz /P wymusza konwersacyjny tryb wykonywania kopii każdego zbioru. Przed przepisaniem każdego zbioru lub katalogu system operacyjny wyświetla na ekranie monitora nazwę zbioru i miejsce jego zapisu, żądając jednocześnie potwierdzenia 'Y' lub odstąpienia 'N' od wykonywania jego kopii,

klucz /S pozwala podczas przepisywania całych katalogów, zmieniać strukturę drzewa katalogowego w wykonywanej kopii. Wykonywanie kopiowania z kluczem /S przedstawiono w trzech przykładach zawartych na końcu tego rozdziału,

klucz /V wymusza wykonywanie przez system operacyjny PC DOS kontroli poprawności zapisu wykonanej kopii (nie jest to porównywanie kopii z wersją źródłową lecz sprawdzenie zachowania zasad zapisu na dysku lub dyskietce). Opcja /V spowalnia wykonanie rozkazu XCOPY, ale zmniejsza ryzyko błędnego zapisu,

klucz /W wymusza zawieszenie poszukiwania na dyskietce źródłowej zbiorów, które mają zostać skopiowane, na czas załadowania tej dyskietki. Wyświetlony zostaje przy tym następujący komunikat:

Press any key to begin copying file(s)

Wznowienie kopiowania nastąpi po wciśnięciu klawisza dowolnej litery.

Uwagi dotyczące posługiwania się rozkazem XCOPY:

1. Jeśli w linii komendy rozkazu XCOPY pominięto parametr 'ścieżka1', to kopiowanie rozpocznie się od katalogu roboczego.
2. Jeśli na dyskietce lub dysku wyjściowym nie istnieje katalog opisany parametrem 'ścieżka2', to przed wykonaniem kopiowania zostanie on tam utworzony.
3. Kopiowanie wykonywane rozkazem XCOPY pozwala także na zmianę nazw zbiorów (analogicznie jak w przypadku rozkazu COPY) i nazw katalogów.
4. Rozkazem XCOPY nie można przesyłać zbiorów na urządzenia: CON, COM1, COM2, LPT1 i LPT2.
5. Specyfikacja nazwy urządzenia, ścieżki, nazwy i rozszerzenia zbioru nie może zawierać więcej niż 63 znaki.
6. Jeśli podczas wykonywania rozkazu XCOPY na dyskietce źródłowej zabraknie miejsca, to wyświetlony zostanie następujący komunikat:

'Disk full'

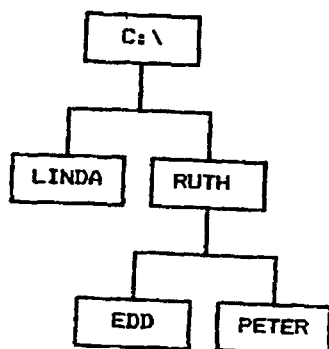
✓ i kopiowanie zostanie zakończone.

7. Rozkaz XCOPY nie pozwala na wykonywanie kopii 'ukrytych' zbiorów systemowych: IBMBIO.COM i IBMDOS.COM.

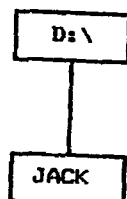
Przykłady:

Założmy, że przed wykonaniem rozkazu XCOPY drzewa katalogowe na dysku C: (dysk źródłowy) i dysku D: (dysk przeznaczenia) mają następującą postać:

Dysk źródłowy:



Dysk przeznaczenia:

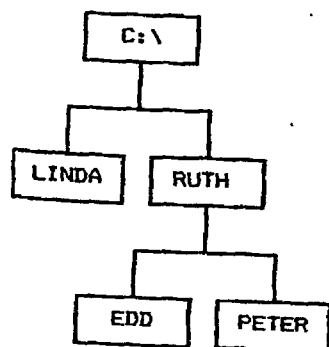


1. Po wykonaniu następującego rozkazu XCOPY:

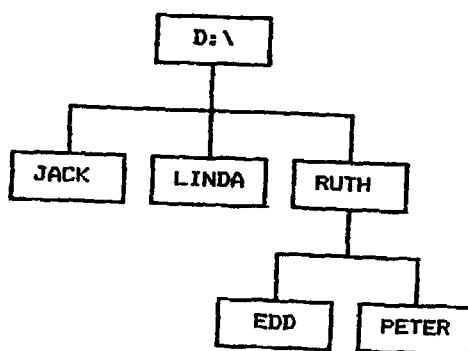
`XCOPY C:\ D:\ /S`

drzewo katalogowe na dysku przeznaczenia będzie miało postać:

Dysk źródłowy:



Dysk przeznaczenia:



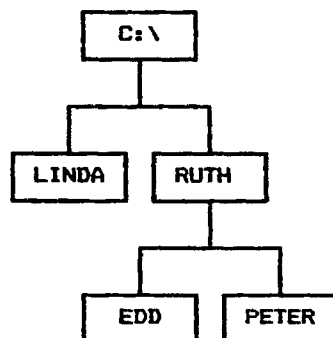
W powyższym przykładzie wykonywanie rozkazu XCOPY rozpoczyna się na dysku źródłowym C: i na dyskietce wyjściowej D:, od katalogu podstawowego.

2. Po wykonaniu następującego rozkazu XCOPY:

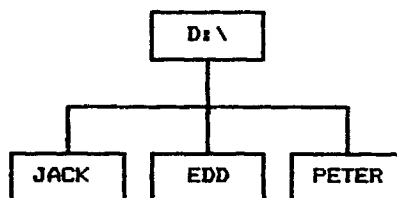
```
XCOPY C:\RUTH D:\ /S
```

drzewo katalogowe na dysku przeznaczenia będzie miało postać:

Dysk źródłowy:



Dysk przeznaczenia:



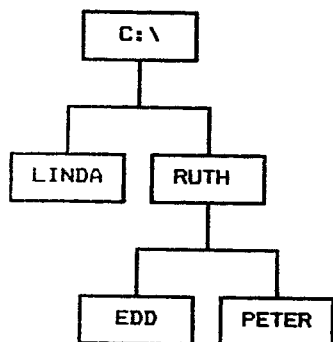
W powyższym przykładzie w wykonanej kopii nastąpi zmiana struktury drzewa katalogowego: katalogi podrzędne EDD i PETER będą zapisane bezpośrednio w katalogu podstawowym (pozostała zawartość katalogu RUTH zostanie także wpisana bezpośrednio do katalogu podstawowego).

3. Po wykonaniu następującego rozkazu XCOPY:

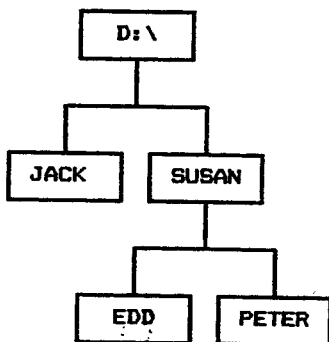
```
XCOPY C:\RUTH D:\SUSAN /S
```

drzewo katalogowe na dysku przeznaczenia będzie miało postać:

Dysk źródłowy:



Dysk przeznaczenia:



W powyższym przykładzie w wykonanej kopii nie nastąpi zmiana struktury drzewa katalogowego. Zamiast katalogu RUTH utworzony zostanie katalog SUSAN, do którego zostanie przepisana zawartość katalogu RUTH.

7. ZBIÓR ROZKAZÓW PRZETWARZANIA WSADOWEGO.

Zbiór rozkazów przetwarzania wsadowego (Batch File Commands) jest ciągiem rozkazów, akceptowanych przez system operacyjny PC DOS, które są odczytywane z tego zbioru, a następnie wykonywane zgodnie z kolejnością wynikającą z ich położenia, lub ustaloną specjalnymi rozkazami sterującymi.

Format:

[d:] [ścieżka] nazwa [.BAT] [parametry]

Typ:

Wykonywany jak rozkaz wewnętrzny.

Opis:

Zbiór rozkazów przetwarzania wsadowego może zawierać jeden lub więcej rozkazów wykonywanych przez system operacyjny PC DOS. Zbiór taki można utworzyć przy pomocy dowolnego edytora, bądź bezpośrednio z klawiatury konsoli, wykorzystując rozkaz COPY:

COPY CON nazwa.BAT 'Enter'

. . . ciąg kolejnych rozkazów, każdy zakończony
. . . wciśnięciem klawisza 'Enter'

klawisz 'F6', a następnie 'Enter'

W razie pomyłki można przerwać wpisywanie, bez zachowania dotychczas wpisanego fragmentu, wciskając kombinację klawiszy 'CTRL/Break'.

Tworząc zbiór rozkazów do przetwarzania wsadowego należy pamiętać, że:

1. Nie należy oznaczać go nazwą identyczną z nazwą któregośkolwiek z rozkazów wewnętrznych.
2. Przy inicjacji wykonania zbioru rozkazów przetwarzania wsadowego wystarczy podanie jego nazwy, bez rozszerzenia .BAT.
3. Zbiór rozkazów przetwarzania wsadowego, oprócz tych, które wymieniono w rozdziale pt. 'ROZKAZY SYSTEMU OPERACYJNEGO PC DOS', może dodatkowo zawierać następujące

rozkazy: ECHO, FOR, GOTO, IF, SHIFT, PAUSE i REM. Zostaną one szczegółowo opisane w dalszej części tego rozdziału

4. Usunięcie dyskietki zawierającej aktualnie wykonywany zbiór rozkazów powoduje wyświetlenie komunikatu o konieczności jej ponownego założenia.
5. Inicjację wykonania zbioru rozkazów można wykonać z 'wnętrza' innego zbioru rozkazów, pod warunkiem, że jest to ostatni rozkaz w zbiorze wywołującym. System operacyjny nie zwraca sterowania do zbioru wywołującego po zakończeniu wykonywania zbioru wywołanego.
6. System operacyjny zachowuje katalog, z którego zainicjowano wykonanie zbioru rozkazów, nawet wówczas, gdy zbiór ten zawiera rozkaz zmiany katalogu.
7. Przerwanie wykonywania zbioru rozkazów można wykonać wciskając kombinację klawiszy 'Ctrl/Break'. Spowoduje to zawieszenie wykonywania z jednoczesnym zapytaniem systemu operacyjnego:

Terminate batch job (Y/N)?

Wciśnięcie klawisza 'Y' spowoduje wznowienie wykonywania pozostałych rozkazów, natomiast wciśnięcie klawisza 'N' ostatecznie przerwie wykonywanie zbioru rozkazów.

8. Szybkość wykonywania zbioru rozkazów może ulec zwiększeniu, jeśli zostanie on umieszczony w tzw. dysku wirtualnym (jest to fragment pamięci operacyjnej, patrz rozdział pt. 'KONFIGUROWANIE SYSTEMU OPERACYJNEGO').

Zbiór rozkazów przetwarzania wsadowego .BAT może zawierać również parametry formalne (ang. dummy parameters), które podczas wykonywania tego zbioru zostaną zastąpione przez określone wartości. Np. jeśli zbiór rozkazów o nazwie EXAMPLE.BAT zawiera następujące rozkazy:

```
COPY %1.mac %2.mac  
TYPE %2.prn  
TYPE %0.bat
```

to tymi parametrami formalnymi są: %0, %1, %2 i zostaną one zastąpione wartościami podanymi przez użytkownika podczas przetwarzania tego zbioru. Należy zaznaczyć, że za parametr formalny %0 będzie zawsze podstawiona nazwa urządzenia z zapisanym zbiorem rozkazów (o ile nie jest to urządzenie robocze) lub nazwa tego zbioru rozkazów (dokładniej zostanie to wyjaśnione na przykładzie poniżej).

U W A G A :

1. Zbiór rozkazów przetwarzania wsadowego może zawierać do 10 parametrów formalnych (od %0 do %9). Ich liczbę można zwiększyć używając rozkazu SHIFT.
2. Używając w zbiorze wsadowym znaku '%' należy zapisać go jako '%%', np. dla określenia zbioru ABC%.EXE należy użyć w zbiorze wsadowym zapisu ABC%%.EXE.

Aby wykonać zbiór rozkazów o nazwie EXAMPLE.BAT, zamieszczony na poprzedniej stronie, podstawiając za parametry formalne ich wartości aktualne, należy kolejno, po podaniu nazwy tego zbioru, wymienić aktualne wartości parametrów:

A>EXAMPLE A:prog1 B:prog2

co spowoduje podstawienie za parametr formalny %0 wartości EXAMPLE, za %1 - wartości A:prog1, a za %2 - wartości B:prog2. Identyczny efekt byłby osiągnięty, jeśli zbiór rozkazów EXAMPLE.BAT miałby postać:

A>COPY A:prog1.mac B:prog2.mac

A>TYPE B:prog2.prn

A>TYPE EXAMPLE.bat

Zbiór rozkazów przetwarzania wsadowego AUTOEXEC.BAT:

Podczas każdego restartu systemu operacyjnego PC DOS zostaje automatycznie wykonany zbiór rozkazów przetwarzania wsadowego o nazwie AUTOEXEC.BAT. Znajduje się on zawsze na dysku z systemem operacyjnym, w katalogu podstawowym. Zbiór AUTOEXEC.BAT pozwala na wykonywanie szeregu rozkazów, realizujących czynności wstępne, podejmowane przed rozpoczęciem właściwej pracy. Czynnościami tymi mogą być: ustawienie aktualnej daty i czasu (rozkazy DATE i TIME), wyświetlanie komunikatów, czyszczenie ekranu monit., a itp.

7.1. Rozkaz ECHO.

Format:

ECHO [ON|OFF:message]

Typ:

Rozkaz wewnętrzny

Opis:

Rozkaz ECHO umożliwia lub blokuje wyświetlanie tekstów rozkazów, odczytywanych ze zbioru wsadowego. Standardowo po włączeniu zasilania komputera ustawione jest ECHO ON. Rozkaz ECHO bez podania argumentów powoduje wyświetlenie aktualnego stanu ON lub OFF. Komunikat zapisany w linii komendy rozkazu ECHO (parametr 'message') zawsze zostanie wyświetlony, niezależnie od ustawionego stanu wyświetlania.

Przykład:

Niech zbiór rozkazów zawiera następujące rozkazy:

```
ECHO OFF
REM **** command display is off
DIR a:/w
ECHO ON
DIR a:/w
```

Po zainicjowaniu wykonania tego zbioru na ekranie monitora pojawiają się następujące komunikaty:

A>ECHO OFF

Volume on drive A has no ID
Directory of A:\

file1.ext file2.ext

2 file(s) xxxxx bytes free

A>ECHO ON

A>DIR a:/w

Volum in drive A has no ID
Directory of A:\

file1.ext file2.ext

2 file(s) xxxxx bytes free

W powyższym przykładzie wyświetlone zostanie ECHO OFF, natomiast rozkazy: REM i DIR a:/w nie będą wyświetlone (wyświetlony będzie efekt działania tych rozkazów). Ustawienie stanu wyświetlania na ON spowoduje wyświetlenie rozkazów: ECHO ON i DIR a:/w.

7.2. Rozkaz FOR.

Format:

FOR %%variable IN (set) DO command

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz FOR umożliwia wykonanie tego samego polecenia dla pewnego zestawu argumentów.

Poszczególne parametry występujące w linii komendy rozkazu FOR mają następujące znaczenie:

'%variable' - oznacza zmienną, pod którą będą podstawiane poszczególne elementy zbioru 'set',

'set' - oznacza zbiór argumentów, dla których będą kolejno wykonywane polecenia 'command'. W zbiorze tym mogą wystąpić znaki specjalne '*' i '?',

'command' - oznacza symbolicznie rozkaz systemu operacyjnego PC DOS.

U W A G A :

1. Zapis %variable wymagany jest tylko przy przetwarzaniu wsadowym. Jeśli rozkaz FOR będzie wykonywany jak zwykły rozkaz systemu operacyjnego PC DOS, to należy pisać %variable, np.

A>FOR %h IN (file1) DO dir %h

2. Rozkaz FOR nie może być zagnieżdżony, tj. przy jego pomocy nie wolno wywoływać innego rozkazu FOR.

Przykład:

Rozkaz FOR o następującej postaci:

FOR %h IN (prog1.ASM prog2.ASM prog3.ASM) DO dir %h

spowoduje taki sam efekt, jak wykonanie trzech rozkazów:

dir prog1.ASM

dir prog2.ASM

dir prog3.ASM

7.3. Rozkaz GOTO.

Format:

GOTO etykieta

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz GOTO przekazuje sterowanie do linii następującej po linii z etykietą. Jeśli w zbiorze rozkazów nie zdefiniowano tej etykiety, to jego wykonywanie zostanie przerwane z komunikatem:

Label not found

Etykieta może być dowolny ciąg liter poprzedzony dwukropkiem, z których znaczenie ma pierwszych osiem znaków. Etykieta nie może zawierać znaku kropki '.'. Podczas wykonywania zbioru rozkazów, linie z etykietami nie są wyświetlane (a więc wstawienie etykiet do których nie odwołuje się żaden rozkaz można traktować jako komentarz).

Przykład:

```
:PRZYKŁAD  
REM TO JEST PRZYKŁAD  
GOTO PRZYKŁAD
```

Powyższy ciąg rozkazów realizuje nieskończoną sekwencję wyświetlania komentarza: TO JEST PRZYKŁAD.

7.4. Rozkaz IF.

Format:

IF [NOT] condition command

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz zdefiniowany argumentem 'command' zostanie wykonany jeśli wartość logiczna warunku 'condition' jest TRUE. Warunek ten może mieć jedną z trzech następujących postaci:

1. ERRORLEVEL number

Warunek ten ma wartość logiczną TRUE, jeśli ostatnio wykonany program miał kod zakończenia (ang. exit code) większy lub równy podanej liczbie 'number'. Liczba ta musi być zapisana w postaci dziesiętnej.

U W A G A :

Z rozkazów systemu operacyjnego PC DOS tylko BACKUP, FORMAT i RESTORE ustalają warunki jakie można badać rozkazem IF ERRORLEVEL.

2. ciag1==ciag2

Warunek ten ma wartość logiczną TRUE, jeśli oba ciągi znaków są identyczne (rozróżnialne są małe i duże litery).

3. EXIST [d:]nazwa[.rozszerzenie]

Warunek ten ma wartość logiczną TRUE, jeśli odnaleziony zostanie na urządzeniu d: wyspecyfikowany zbiór (dopuszczalne jest użycie znaków specjalnych: '?' i '*').

Przykład 1:

```
MYPROG1
IF ERRORLEVEL 1 ECHO MYPROG1 FAILED
```

Powyższe dwie komendy tworzą zbiór rozkazów. Wykonanie programu MYPROG1 ustala wartość jego kodu zakończenia (ERRORLEVEL) w zależności od tego, czy wykonał się poprawnie (ERRORLEVEL=0), czy niepoprawnie (ERRORLEVEL=1). W drugim przypadku na ekranie monitora pojawi się komunikat będący efektem działania rozkazów IF i ECHO:

```
MYPROG1 FAILED
```

Przykład 2:

```
IF %1==xxxx ECHO HASŁO JEST POPRAWNE
```

W tym przypadku komunikat:

```
HASŁO JEST POPRAWNE
```

zostanie wyświetlony na ekranie monitora, jeśli wartością parametru %1 jest ciąg znakowy 'xxxx'. O ile warunek nie zostanie spełniony, wykonany zostanie następny rozkaz zbioru wsadowego.

Przykład 3:

```
IF NOT EXIST Zbior1.ASM ECHO Zbior1.ASM nie istnieje na dysku
```

Jeśli na urządzeniu roboczym nie istnieje zbiór o nazwie Zbior1.ASM, to na ekranie monitora pojawi się następujący komunikat:

```
Zbior1.ASM nie istnieje na dysku
```

7.5. Rozkaz PAUSE.

Format:

```
PAUSE [remark]
```

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz PAUSE wstrzymuje wykonanie programu i powoduje wyświetlenie komunikatu określonego parametrem 'remark' oraz ponaglenia:

Strike any key when ready ...

Wznowienie wykonywania zbioru rozkazow nastapi po wciśnięciu dowolnego klawisza. Wciśnięcie kombinacji klawiszy 'Ctrl/Break', a następnie klawisza 'Y' kończy wykonywanie całego zbioru rozkazów. Komunikat 'remark' może być dowolnym ciągiem o długości do 121 znaków.

Przykład:

Rozkaz PAUSE o następującej postaci:

PAUSE Wymien dyskietke

spowoduje, że system operacyjny zawiesza wykonywanie zbioru wsadowego z komunikatem 'Wymien dyskietke', co pozwala na wymianę dyskietki. Kontynuacja jest możliwa po wciśnięciu klawisza dowolnej litery.

7.6. Rozkaz REM (Remark).

Format:

REM [remark]

Typ:

Rozkaz wewnętrzny.

Opis:

Rozkaz REM powoduje wyświetlenie komunikatu określonego parametrem 'remark', o ile stan wyświetlania, ustawiony rozkazem ECHO jest ON (w przeciwnym wypadku komunikat nie jest wyświetlany). Komunikat 'remark' może być dowolnym ciągiem o długości do 123 znaków. Rozkazu REM można używać bez wyspecyfikowania komunikatu, dla oddzielania pustymi liniami, dla poprawy czytelności, poszczególnych poleceń w zbiorze rozkazów.

7.7. Rozkaz SHIFT.

Format:

SHIFT

Typ:

Rozkaz wewnętrzny...

Opis:

Podczas przetwarzania wsadowego możliwe jest wykorzystanie do 10 parametrów formalnych. Rozkaz SHIFT pozwala zwiększyć tę liczbę poprzez 'przesunięcie' argumentów parametrów o jedną pozycję w lewo: parametr %0 będzie zastąpiony parametrem %1, parametr %1 parametrem %2 etc.

Przykład:

Niech zbiór rozkazów MYFILE.BAT zawiera następujące polecenia:

```
ECHO %0 %1 %2 %3  
SHIFT  
ECHO %0 %1 %2 %3  
SHIFT  
ECHO %0 %1 %2 %3  
SHIFT  
ECHO %0 %1 %2 %3  
SHIFT
```

Zainicjowanie wykonania tego zbioru z następującymi parametrami:

```
MYFILE PROG1 PROG2 PROG3
```

spowoduje wyświetlenie następujących napisów:

A>ECHO MYFILE PROG1 PROG2 PROG3
MYFILE PROG1 PROG2 PROG3

A>SHIFT

A>ECHO PROG1 PROG2 PROG3
PROG1 PROG2 PROG3

A>SHIFT

A>ECHO PROG2 PROG3
PROG2 PROG3

A>SHIFT

A>ECHO PROG3
PROG3
A>SHIFT
A>ECHO

A>

8. EDYTOR LINIOWY - EDLIN.

8.1. Wprowadzenie.

Edytor liniowy EDLIN służy do tworzenia, poprawiania i przeglądania zbiorów tekstowych. Podczas sesji pracy z edytorem EDLIN zbiór tekstowy zostaje przez niego umownie podzielony na tzw. linie (fragmenty tekstu zakończone znakiem sterującym powrotu karetki CR), mogące zawierać do 253 znaków każda. Podczas edycji na początku każdej linii zostaje wyświetlony jej numer (nie należący jednak do poprawianego zbioru). Bez względu na to, czy w trakcie dotychczasowej pracy z edytorem skasowano, czy wstawiono dodatkowe linie, są one zawsze na bieżąco numerowane w stosunku do pierwszej linii poprawianego tekstu, znajdującego się aktualnie w pamięci operacyjnej.

8.2. Wywołanie edytora liniowego EDLIN.

Edytor liniowy EDLIN jest wywoływany następującą komendą:

```
[d:]ścieżkaEDLIN[d:]ścieżka[nazwa[.rozszerzenie]][/B]
```

gdzie:

- [d:]ścieżka przed słowem kluczowym EDLIN określa miejsce, w którym znajduje się program EDLIN,
- pozostałe parametry określają 'współrzędne' zbioru, który będzie poddany edycji,
- klucz /B implikuje sposób traktowania przez edytor kodu sterującego 'Ctrl/Z', oznaczającego koniec zbioru. Kod ten może wystąpić wielokrotnie w poprawianym zbiorze i wywołanie edytora EDLIN bez użycia klucza /B spowoduje, że edycja zbioru będzie ograniczona do fragmentu od jego początku do pierwszego wystąpienia znaku sterującego 'Ctrl/Z'.

Jeśli parametry dotyczące zbioru określają istniejący już zbiór, zostanie on ściągnięty do pamięci operacyjnej w taki sposób, że zapełni co najwyżej 75% jej pojemności. Jeśli zbiór zmieścił się w całości, to na ekranie monitora pojawia się komunikat:

End of input file

*_

Jeśli zbiór jest zbyt duży, tj. zająłby więcej niż 75% pojemności pamięci operacyjnej, to edytor zgłasza się na ekranie monitora znakiem '*', co oznacza, że edycji można poddać tę część zbioru, która zmieściła się w pamięci. Aby wykonać ewentualne poprawki w pozostałej części należy przepisać poprawiony fragment tekstu do pamięci masowej i wpisać do pamięci operacyjnej dalszą część zbioru. Jeśli wyszczególniony w rozkazie wywołania edytora zbiór jeszcze nie istnieje (będzie dopiero utworzony), wówczas na ekranie monitora pojawia się następujący komunikat:

New file

*_

co oznacza, że edytor oczekuje na wpisanie komendy I (insert) a następnie tekstu, który utworzy zbiór wyjściowy.

W odróżnieniu od edytora ekranowego, edytor liniowy operuje tzw. wskaźnikiem oraz numerami linii zamiast kursorem. Linie są wyświetlane łącznie z ich numerami. Każdy numer linii jest zakończony dwukropkiem. Jeśli wskaźnik jest ustawiony w danej linii, po dwukropku zostaje wyświetlony znak *, w przeciwnym wypadku znak spacji. Ani numer linii, ani żaden ze znaków: dwukropka, spacji lub gwiazdki nie należy do poprawianego zbioru.

8.3. Ogólne informacje dotyczące komend edytora EDLIN.

Parametry podawane podczas wywoływania komend edytora EDLIN:

nr - oznacza, że trzeba podać numer linii. Istnieją wówczas trzy możliwości:

- podanie liczby dziesiętnej z przedziału od 1 do 65529 (tylko z tego zakresu można podawać numery linii). Jeżeli zachodzi konieczność wyspecyfikowania kilku numerów linii, to muszą być oddzielone od siebie przecinkiem lub spacją.

- podanie znaku '#', który oznacza pierwszą linię po ostatniej linii poprawianego tekstu (tego znaku można użyć w przypadku dopisywania do końca zbioru dodatkowego tekstu).
- podanie znaku kropki '.', oznacza linię w której aktualnie znajduje się wskaźnik.

n - oznacza ilość linii,

'ciąg' - oznacza ciąg znaków, reprezentujący fragment tekstu, który będzie poszukiwany w poprawianym zbiorze, usuwany z niego lub zastępowany innym ciągiem.

[] - oznacza, że składnik komendy, ujęty w ten nawias jest opcjonalna częścią tej komendy.

Ogólne informacje dotyczące komend edytora EDLIN:

1. Wszystkie komendy edytora EDLIN, za wyjątkiem komendy EDIT LINE, są wywoływane poprzez podanie pojedynczej litery.
2. Wszystkie komendy, za wyjątkiem komend: END EDIT oraz QUIT EDIT, wymagają określenia pewnych parametrów. Jeśli parametry te zostaną pominięte, to edytor przyjmie ich wartości domyślne.
3. Komendy oraz parametr 'ciąg' można podawać zarówno małymi jak i dużymi literami.
4. Wykonanie komendy jest inicjowane wciśnięciem klawisza 'Enter', natomiast zatrzymanie wykonania - kombinacją klawiszy 'Ctrl/Break'.
5. Jeśli dana komenda powoduje wyświetlenie na ekranie monitora dużej ilości zbędnych dla użytkownika informacji, to można zawiesić wyświetlanie wciskając kombinację klawiszy 'Ctrl/NumLock'. Powtórne wyświetlanie można zainicjować wciśnięciem klawisza dowolnej litery.
6. Dozwolone jest wykorzystywanie kluczy funkcyjnych, akceptowanych przez system operacyjny PC DOS.
7. Edytor zgłasza gotowość do przyjęcia komendy znakiem gwiazdki '*'.
8. Możliwe jest odwoływanie się do linii, w kierunku końca lub początku tekstu, w stosunku do linii ze wskaźnikiem,

np. komenda:

-10,+10L

wyświetli 10 linii przed, 10 linii po i linię ze wskaźnikiem.

9. Możliwe jest wyspecyfikowanie kilku komend w jednej linii, np. komenda:

15;-5,+5L

ustawi wskaźnik w 15 linii i wyświetli tę linię wraz z 5 liniami poprzedzającymi i 5 liniami bezpośrednio za tą linią.

10. Aby wpisać do tekstu kod znaku sterującego, należy wcisnąć kombinację klawiszy 'Ctrl/V' i wprowadzić zadany kod używając klawiszy 'Ctrl' i odpowiedniego znaku literowego (duże litery) odpowiadającego żądanej kombinacji (patrz opis klawiatury).

8.4. Komenda A (APPEND LINES).

Zastosowanie komendy APPEND LINES jest uzasadnione w przypadku, gdy zbiór nie mieści się w 75% pamięci operacyjnej. Wówczas część już poprawionego tekstu należy przepisać do pamięci masowej (komenda WRITE LINES), a następnie komendą APPEND LINES o postaci:

[n]A

spowodować dopisanie kolejnych n linii z tej części zbioru, której nie poddano jeszcze edycji. Niewyspecyfikowanie parametru n spowoduje, że zbiór w pamięci operacyjnej będzie uzupełniany do chwili załadowania go w całości, albo wypełnienia 75% pojemności pamięci, przy czym przekroczenie tego zakresu spowoduje automatycznie przerwanie wczytywania. Jeśli zbiór w całości 'wszedł' do pamięci operacyjnej, to na ekranie pojawi się następujący komunikat:

End of input file

8.5. Komenda C (COPY LINES).

Postać komendy:

[nr1],[nr2],nr1,n]C

Opis:

Komenda COPY LINES kopiuje n razy wszystkie linie od numeru nr1 do numeru nr2 przed linię o numerze nr. Możliwe są następujące przypadki szczególne:

- jeśli pominięto jeden lub oba parametry: nr1, nr2, oznacza to, że kopiowanie ma dotyczyć linii ze wskaźnikiem,
- jeśli pominięto parametr n, to edytor przyjmuje ze n=1,
- jeśli wyspecyfikowano parametry nr1, nr2 tak, że nr1<nr2, to edytor zasygnalizuje błąd. Niedozwolone jest także użycie znaku '+' lub '-' w specyfikacji parametru n.

Po wykonaniu tej komendy wskaźnik zostaje ustawiony w pierwszej skopiowanej linii.

8.6. Komenda D (DELETE LINES).

Postać komendy:

[nr1][,nr2]D

Opis:

Komenda DELETE LINES powoduje skasowanie tekstu od linii o numerze nr1 do linii o numerze nr2 włącznie. Możliwe są następujące przypadki szczególne:

- jeśli pominięto pierwszy parametr, tzn. komenda będzie miała postać:

,nr2D

to kasowanie rozpocznie się od linii ze wskaźnikiem. Przecinek wskazuje tutaj pominięcie tego parametru i musi wystąpić,

- jeśli pominięto drugi parametr, tzn. komenda będzie miała postać:

nr1D lub nr1,D

to zostanie usunięta tylko linia o numerze nr1,

- jeśli pominięto oba parametry, tzn. komenda będzie miała postać:

D

to edytor usunie ze zbioru tylko linię ze wskaźnikiem.

Po wykonaniu kasowania wskaźnik zostanie ustawiony w pierwszej, w kierunku końca zbioru, nieskasowanej linii (nawet wówczas, jeśli zakres kasowania objął ostatnią linię poprawianego tekstu).

Przykład:

Niech dany będzie następujący zbiór tekstowy:

```
1: This is a sample file used to demonstrate
2: line deletion and dynamic
3: line number generation
```

.

.

.

.

```
20: See what happens to the lines
```

```
21: and line numbers when line are
```

```
22: *deleted
```

```
23: See how easy this is to use.
```

Znak '*', wyświetlony na początku linii 22, wskazuje, że jest to linia ze wskaźnikiem. Aby usunąć linie od linii 3 do linii 20 należy użyć komendy DELETE LINES o postaci:

3,20 D

W rezultacie poprawiany zbiór będzie miał postać:

- 1: This is a sample file used to demonstrate
- 2: line deletion and dynamic
- 3: *and line numbers when line are
- 4: deleted
- 5: See how easy this is to use.

Należy zwrócić uwagę, że po usunięciu linii 3-20, linie 21-23 zostały przeniebrowane a wskaźnik ustawił się w linii 3. Aby usunąć linię ze wskaźnikiem i linię bezpośrednio za nią, należy użyć komendy DELETE LINES o postaci:

,4 D

W rezultacie poprawiany zbiór będzie miał następującą postać:

- 1: This is a sample file used to demonstrate
- 2: line deletion and dynamic
- 3: *See how easy this is to use.

Po usunięciu ze zbioru linii 3 i 4 numer linii 5 został zmieniony na 3, a wskaźnik nadal znajduje się w linii 3 (tyle że tekst w tej linii jest inny). Aby usunąć ze zbioru pojedynczą linię np. o numerze 2 należy użyć komendy DELETE LINES o postaci:

2 D

W rezultacie poprawiany zbiór będzie miał następującą postać:

- 1: This is a sample file used to demonstrate
- 2: *See how easy this is to use.

Linia ze wskaźnikiem będzie teraz linią o numerze 2. Aby usunąć tę linię wystarczy teraz komenda DELETE LINES bez parametrów:

D

W rezultacie poprawiany tekst będzie miał następującą postać:

- 1: *This is a sample file used to demonstrate

8.7. Komenda EDIT LINE.

Postać komendy:

[nr]

Opis:

Komenda EDIT LINE pozwala na wykonywanie poprawek wewnątrz linii o podanym numerze nr (linię ze wskaźnikiem można 'zgłosić' do poprawki znakiem kropki '.'). Wciśnięcie klawisza 'Enter', po wyspecyfikowaniu numeru linii, spowoduje wyświetlenie jej na ekranie monitora. Dla wykonania poprawek można wykorzystywać wszystkie klucze funkcyjne, akceptowane przez system operacyjny PC DOS (patrz opis klawiatury). Wciśnięcie klawisza 'Enter' sygnalizuje zakończenie edycji linii i powoduje przepisanie jej w aktualnej postaci do zbioru. Jeżeli w poprawianej linii poczyniono już jakieś zmiany, które okazały się zbędne można je anulować wciskając klawisz 'Esc' lub 'Ctrl/Break' zamiast klawisza 'Enter' (takie samo działanie spowoduje wciśnięcie klawisza 'Enter', ale tylko wówczas jeśli kursor znajduje się na początku linii, przy innym położeniu kursora zostanie skasowana zawartość linii na prawo od niego).

Przykład:

Niech dana będzie linia o numerze np. 6 o następującej postaci:

6: This is a sample unedited line.

Należy, przy pomocy komendy EDIT LINE zamienić słowo 'unedited' słowem 'edited'. Algorytm postępowania jest następujący:

1. Wywołanie komendy EDIT LINE w odniesieniu do linii 6. Na ekranie monitora pojawi się następujący tekst:

86

6: This is a sample unedited line.

6: _

Oznacza to, że edytor oczekuje na dalszą reakcję użytkownika

2. Wciśnięcie klawisza funkcyjnego 'F2', a następnie litery 'u'. Edytor odtworzy całą linię 6 aż do pierwszej napotkanej litery 'u':

#6

6: This is a sample unedited line.

6: This is a sample _

3. Usunięcie dwóch kolejnych liter 'un' poprzez dwukrotne wciśnięcie klawisza 'Del', a następnie wciśnięcie klawisza funkcyjnego 'F3'. W rezultacie linia o numerze 6 uzyska postać:

#6

6: This is a sample unedited line.

6: This is a sample edited line._

W tym momencie możliwe jest dopisanie dodatkowego tekstu na końcu poprawionej linii. Zakończenie poprawek w linii 6 użytkownik sygnalizuje wciśnięciem klawisza 'Enter'. Poprawiona linia zostanie wpisana do pamięci operacyjnej.

8.8. Komenda E (END EDIT).

Postać komendy:

E

Opis:

Komenda END EDIT kończy pracę z edytorem i przepisuje poprawiony zbiór do pamięci masowej nadając mu nazwę i rozszerzenie zbioru źródłowego. Zbiór źródłowy (jeśli istniał przed wywołaniem edytora) zachowuje swoją nazwę, ale oznaczony zostaje rozszerzeniem .BAK.

Kończąc pracę z edytorem należy pamiętać, że:

- jeśli w pamięci masowej jest zbyt mało miejsca dla zapisania zbioru wynikowego, wówczas edytor przepisze tylko część tego zbioru pod nazwą zbioru źródłowego, ale z rozszerzeniem .\$\$\$ (\$ to znak dolara). Zbiór źródłowy zachowa swoją nazwę i rozszerzenie,
- jeśli na końcu zbioru źródłowego brak jest znaków sterujących: powrotu karetki i nowej linii, to edytor dopisuje te znaki wraz ze znakiem 'Ctrl/Z', sygnalizującym koniec zbioru tekstowego.

8.9. Komenda I (INSERT LINES).

Postać komendy:

[nr]I

Opis:

Komenda INSERT LINES umożliwia dopisywanie porcji tekstu bezpośrednio przed linią o podanym numerze nr. Możliwe są następujące przypadki szczególne:

- jeśli pominięto numer linii lub zamiast niego wpisano znak kropki '.', to dopisywanie będzie możliwe przed linią ze wskaźnikiem,
- jeśli zamiast numeru linii wpisano znak '#' lub wyspecyfikowano numer linii większy od liczby wszystkich linii w zbiorze, to tekst zostanie dopisany po ostatniej linii.

Pracując w reżimie dopisywania tekstu, edytor EDLIN wyświetla na ekranie monitora numer aktualnie wpisywanej linii, która jest jednocześnie linią ze wskaźnikiem. Aby wyjść z reżimu dopisywania należy wcisnąć kombinację klawiszy 'Ctrl/Break'.

Przykład:

Niech dany będzie fragment zbioru o postaci:

```
1: This is a sample file used to demonstrate
2: #line deletion
3: and dynamic line number generation.
```

Aby wstawić dodatkowy tekst przed linię 3, należy użyć komendy INSERT LINES o postaci:

```
3I
```

Edytor wyświetli następujące ponaglenie:

```
*3I
3: #_
```

i oczekuje na wpisanie dodatkowego tekstu. Użytkownik może kolejno wpisywać:

```
3I 'Enter'
3: #First new line of text 'Enter'
4: #Second new line of text 'Enter'
5: # 'Ctrl/Break'
```

Jeśli po wyświetleniu numeru 5 linii zostanie wciśnięta kombinacja klawiszy 'Ctrl/Break', wówczas wstawianie będzie zakończone, a tekst znajdujący się w pamięci operacyjnej będzie miał postać:

```
1: This is a sample file used to demonstrate
2: #line deletion
3: First new line of text
4: Second new line of text
5: and dynamic line number generation.
```

8.10. Komenda L (LIST LINES).

Postać komendy:

```
[nr1][,nr2]L
```

Opis:

Komenda LIST LINES umożliwia listowanie zawartości zbioru tekstowego, od linii o numerze nr1 do linii o numerze nr2 włącznie, przy czym położenie wskaźnika pozostaje niezmienione. Możliwe są następujące przypadki szczególne:

- jeśli pominięto pierwszy parametr nr1, tzn. komenda będzie miała postać:

,nr2L

to edytor rozpocznie wyświetlanie od 11 linii przed linią ze wskaźnikiem i zakończy na linii o numerze nr2 (przecinek sygnalizuje pominięcie pierwszego parametru i musi wystąpić).

- jeśli pominięto drugi parametr nr2, tzn. komenda będzie miała postać:

nr1L lub nr1,L

to edytor wyświetli 23 linie rozpoczynając od linii o podanym numerze nr1.

- jeśli pominięto oba parametry, tzn. komenda będzie miała postać:

L

to edytor wyświetli także 23 linie: 11 linii przed i 11 linii po linii ze wskaźnikiem oraz linię ze wskaźnikiem.

Przykład:

Niech dany będzie następujący zbiór:

1: This is a sample file used to demonstrate
 2: line deletion and dynamic
 3: line number generation

15:*This is the current line (note the asterisk)

25: See what happens to the lines
 26: and line numbers when lines are
 27: deleted

Aby wyświetlić linie od numeru 3 do numeru 25 należy użyć komendy LIST LINES o postaci:

3,25 L

Na ekranie monitora wyświetlony zostanie następujący fragment tekstu:

3: line number generation

15:*This is the current line (note the asterisk)

25: See what happens to the lines

Aby wylistować 11 linii przed, 11 linii po i linię ze wskaźnikiem należy użyć komendy LIST LINES o postaci:

L

Na ekranie monitora wyświetlony zostanie następujący fragment tekstu:

4: Fourth line of text
 5: Fifth line of text

15:*This is the current line (note the asterisk)

25: See what happens to the lines
 26: and line numbers when lines are

Należy zwrócić uwagę, że położenie wskaźnika nie uległo zmianie w żadnym z przedstawionych przykładów. Znajduje się on zawsze w linii 15.

B.11. Komenda M (MOVE LINES).

Postać komendy:

[nr1],[nr2],nrM

Opis:

Komenda MOVE LINES pozwala na przeniesienie bloku tekstu, ograniczonego liniami o numerach nr1, nr2 włącznie, przed linię o numerze nr. Pominięcie któregoś z opcjonalnych parametrów: nr1, nr2 (albo obu naraz) edytor potraktuje jako wyspecyfikowanie zamiast niego numeru linii ze wskaźnikiem. Po wykonaniu przeniesienia wskaźnik zostaje ustawiony w pierwszej linii przeniesionego bloku.

B.12. Komenda P (PAGE).

Postać komendy:

[nr1][,nr2]P

Opis:

Komenda PAGE umożliwia listowanie zawartości zbioru tekstowego od linii o numerze nr1 do linii o numerze nr2 włącznie, przy czym, w przeciwieństwie do komendy LIST LINES, po zakończeniu listowania wskaźnik zostanie ustawiony w ostatniej wyświetlonej linii. Możliwe są następujące przypadki szczególne:

- jeśli pominięto pierwszy parametr nr1, tzn. komenda będzie miała postać:

,nr2P

to edytor rozpocznie wyświetlanie od następnej linii za linią ze wskaźnikiem i zakończy na linii o numerze nr2,

jeśli pominięto drugi parametr nr2, tzn. komenda będzie miała postać:

nr1P lub nr1,P

to edytor wyświetli 23 linie, rozpoczynając od linii o podanym numerze nr1.

- jeśli pominięto oba parametry, tzn. komenda będzie miała postać:

P

to edytor wyświetli także 23 linie, rozpoczynając od linii ze wskaźnikiem.

8.13. Komenda Q (QUIT EDIT).

Postać komendy:

Q

Opis:

Komenda QUIT EDIT pozwala na zakończenie sesji z edytorem liniowym EDLIN z zachowaniem zbioru źródłowego (pod tą samą nazwą i rozszerzeniem), ale bez zachowywania zbioru z poprawkami. Przed podjęciem działania edytor 'upewnia się' co do dalszych czynności wyświetlając na ekranie monitora komunikat:

Abort edit (Y/N)?_

Użytkownik musi wówczas potwierdzić lub zaprzeczyć zamiarowi zakończenia edycji bez zachowania zbioru wyjściowego.

8.14. Komenda R (REPLACE TEXT).

Postać komendy:

```
[nr1][,nr2][?]R[ciag1][<F6>ciag2]
```

Opis:

Komenda REPLACE TEXT pozwala na zastąpienie ciągu znaków 'ciag1' ciągiem znaków 'ciag2', od linii o numerze nr1 do linii o numerze nr2 włącznie. Separatorem pomiędzy obydwojema ciągami jest tutaj wciśnięcie klawisza 'F6' lub kombinacji klawiszy 'Ctrl/Z'. Wspecyfikowanie opcjonalnego klucza '?' spowoduje, że przed każdą zmianą edytor zgłosi się pytaniem:

O.K.?

i oczekuje na potwierdzenie 'Y' lub zaprzeczenie (dowolny znak) (oba zakończone wciśnięciem klawisza 'Enter') zamiaru modyfikacji w danym, konkretnym przypadku. Bez względu na obecność klucza '?' edytor przed wykonaniem modyfikacji zawsze wyświetla linię zawierającą odszukany ciąg znaków 'ciag'. Możliwe są następujące przypadki szczególne:

- pominięcie 'ciag2' w specyfikacji linii komendy powoduje skasowanie wszystkich 'ciagow2' wewnątrz wyszczególnionego zakresu,
- pominięcie 'ciag1' w specyfikacji linii komendy powoduje, że poszukiwany będzie do ewentualnej modyfikacji 'ciag1', określony podczas wykonywania ostatniej komendy REPLACE TEXT lub SEARCH TEXT;
- pominięcie obu parametrów 'ciag1' i 'ciag2' w specyfikacji linii komendy spowoduje, że za te parametry będą podstawione odpowiednie parametry wyszczególnione w ostatnich komendach: REPLACE TEXT i SEARCH TEXT,
- pominięcie parametru nr1 spowoduje, że poszukiwania i modyfikacje będą prowadzone od linii ze wskaźnikiem,
- pominięcie parametru nr2 spowoduje, że poszukiwania i modyfikacje będą prowadzone aż do ostatniej linii fragmentu tekstu, który zmieścił się w pamięci operacyjnej (lub do końca zbioru - jeśli został wpisany w ca-

kości).

Przykład:

Niech dany będzie następujący zbiór tekstowy:

```
1: This is a sample file
2: used to demonstrate
3: the Replace and Search Text commands
4: This includes the
5: optional parameter ?
6: and required string
7: #parameter.
```

Aby zastąpić ciąg znaków 'and' ciągiem 'or' pomiędzy liniami o numerach 1 i 7 włącznie, należy użyć komendy REPLACE TEXT o postaci:

```
1,7 Rand'F6'or'Enter'
```

gdzie: 'F6' i 'Enter' oznaczają wciśnięcie odpowiednich klawiszy. Edytor wykona odpowiednie modyfikacje: dwukrotnie w linii 3 i jednokrotnie w linii 7, sygnalizując wykonywane zmiany wyświetlaniem odpowiednich linii w zmodyfikowanej postaci:

```
3: the Replace or Search commands.
3: the Replace or Search commors.
6: or required string.
```

Linia o numerze 6 staje się linią ze wskaźnikiem. W powyższym przykładzie wszelkie modyfikacje mogą być wykonywane w trybie pytanie/odpowiedź. W tym celu należy użyć komendy REPLACE TEXT o postaci:

```
1,7 ? Rand'F6'or'Enter'
```

Przed wykonaniem kolejnej modyfikacji edytor 'upewni się' o konieczności jej wykonania:

```
*1,7 ? Rand'F6'or'Enter'
3: the Replace or Search Text commands
O.K.? Y
3: the Replace or Search Text commors
O.K.? N
6: or required string
O.K.? Y
*
```

A zatem zgodnie z żądaniem użytkownika edytor wykona pojedyncze modyfikacje wewnątrz linii o numerach 3 i 6.

8.15. Komenda S (SEARCH TEXT).

Postać komendy:

[nr1][,nr2][?]S[ciąg]

Opis:

Komenda SEARCH TEXT pozwala na poszukiwanie linii zawierającej wyspecyfikowany ciąg znaków 'ciąg', od linii o numerze nr1 do linii o numerze nr2. Odszukana linia zostaje wyświetlona na ekranie monitora z jednoczesnym ustawieniem w niej wskaźnika. Wyspecyfikowanie opcjonalnego klucza '?' spowoduje, że edytor, po wyświetleniu odszukanej linii zgłosi się pytaniem:

O.K.?

i oczekuje na potwierdzenie (Y) lub zaprzeczenie (dowolny znak, oba zakończone wciśnięciem klawisza 'Enter'), czy odszukano właściwą linię. W tym drugim przypadku poszukiwania będą kontynuowane. Edytor ustawia wskaźnik w odszukanej linii, jeżeli uzyska potwierdzenie 'Y'. Jeśli wyspecyfikowany ciąg nie istnieje w podanym zakresie tekstu, to na ekranie monitora zostanie wyświetlony następujący komunikat:

Not found

a położenie wskaźnika nie ulegnie zmianie. Możliwe są następujące przypadki szczególne:

- pominięcie parametru nr1 spowoduje, że poszukiwania będą kontynuowane od linii ze wskaźnikiem,
- pominięcie parametru nr2 spowoduje, że poszukiwania będą prowadzone do ostatniej linii tekstu, która zmieściła się w pamięci operacyjnej (lub do końca zbioru - jeśli zmieścił się w całości).
- pominięcie parametru 'ciąg' spowoduje, że edytor będzie poszukiwał linii z ciągiem znaków wyspecyfikowanych w ostatniej komendzie REPLACE TEXT lub SEARCH TEXT.

Należy zwrócić uwagę, że edytor EDLIN odróżnia, w poszukiwanym ciągu znaków, duże litery od małych.

Przykład:

Niech dany będzie następujący zbiór tekstowy:

```
1: This is a sample file
2: used to demonstrate
3: the Search Text commands
4: This includes the
5: optional parameter ?
6: and required string
7: #parameter.
```

Aby odszukać pierwszą linię, w której występuje ciąg znaków 'and' należy użyć jednej z trzech poniższych postaci komendy SEARCH TEXT:

```
albo: 1,7 Sand
albo: 1, Sand
albo: 1Sand
```

Po odnalezieniu takiej linii edytor wyświetli ją na ekranie monitora:

```
3: the Search Text command.
```

*

Aby dalej kontynuować poszukiwania linii zawierających ten sam ciąg znaków 'and' wystarczy teraz wywoływać komendę SEARCH TEXT bez podawania parametrów:

```
*1,7 Sand
3: the Search Text command.
```

*S

```
6: and required string
```

*

Wskaźnik zostanie ustawiony zawsze w ostatniej odnalezionej linii, zawierającej poszukiwany ciąg znaków. Poszukiwania mogą być także prowadzone w trybie pytanie/odpowiedź. W tym celu należy użyć komendy SEARCH TEXT o postaci:

\$1,7 ? Sand
3: the Search Text command.
O.K.? N
6: and required string
O.K.? Y
\$

8.16. Komenda T (TRANSFER LINES).

Postać komendy:

[nr]T:[d:]nazwa zbioru

Opis:

Komenda TRANSFER LINES pozwala na wstawienie przed linią o numerze nr tekstu stanowiącego zawartość zbioru o wyspecyfikowanej nazwie. Jeśli nie podano parametru nr, to edytor wstawi tekst przed linię ze wskaźnikiem.

U W A G A :

Wstawiany zbiór będzie czytany z urządzenia 'd:' (urządzenia roboczego, jeśli nie podano tego parametru), z katalogu, który był ustawiony w momencie wywoływania edytora EDLIN.

8.17. Komenda W (WRITE LINES).

Zastosowanie komendy WRITE LINES jest uzasadnione w przypadku, gdy zbiór nie mieści się w 75% pamięci operacyjnej. Po wykonaniu poprawek w pierwszych n liniach tekstu, można przepisać je do pamięci masowej (zawsze od pierwszej linii tekstu znajdującego się w pamięci operacyjnej) wykorzystując komendę WRITE LINES o postaci:

[n]W

Wykonanie tej komendy zwalnia miejsce w pamięci operacyjnej na dalszą część poprawianego zbioru. Jeśli w specyfikacji linii komendy pominięty zostanie parametr 'n', to edytor przepisze zawartość pierwszych 25% komórek pamięci zawierających poprawiany zbiór, o ile zbiór ten wykracza poza ten obszar. W przeciwnym wypadku edytor nie spowoduje żadnych zmian w położeniu zbioru. Wszystkie pozostałe w pamięci operacyjnej linie zostaną przenieśmerowane w ten sposób, że pierwsza linia nie przepisana do pamięci masowej będzie oznaczona numerem 1.

8.18. Zestawienie komend edytora EDLIN.

APPEND LINES	- przepisywanie z pamięci masowej do pamięci operacyjnej,
COPY LINES	- kopiowanie określonych linii przed określoną linią,
DELETE LINES	- kasowanie określonych linii ze zbioru,
EDIT LINE	- wykonywanie poprawek wewnątrz określonej linii,
END EDIT	- zakończenie pracy z edytorem z zachowaniem zbioru źródłowego i zbioru wynikowego,
INSERT LINES	- wstawianie dodatkowego tekstu z klawiatury,
LIST LINES	- listowanie bez zmiany położenia wskaźnika,
QUIT EDIT	- zakończenie pracy z edytorem z zachowaniem zbioru źródłowego, ale bez zachowania zbioru wynikowego,
MOVE LINES	- przeniesienie określonych linii w określone miejsce w tekście,
PAGE	- listowanie ze zmianą położenia wskaźnika,
REPLACE TEXT	- zastępowanie w tekście określonego ciągu znaków innym ciągiem,
SEARCH TEXT	- poszukiwanie w tekście określonego ciągu znaków,
TRANSFER LINES	- dopisywanie zbioru do tekstu poddawanego edycji,
WRITE LINES	- przepisywanie z pamięci operacyjnej do pamięci masowej.

9. PROGRAM KONSOLIDACJI ZBIORÓW LINK.

9.1. Wprowadzenie do programu LINK.

Każdy program zredagowany w dowolnym języku programowania (z wyjątkiem BASIC-a) musi zostać przetworzony z postaci symbolicznej na postać kodu maszynowego. Czynność tę, nazywaną kompilacją (lub asemblacją), wykonują określone dla każdego języka programowania kompilatory (lub asemblery). W wyniku działania kompilatora (lub asemblera) otrzymuje się tzw. moduł wynikowy o rozszerzeniu .OBJ, będący programem w kodzie maszyny, ale nie nadający się jeszcze do bezpośredniego uruchamiania. Kompilacji (lub asemblacji) można poddawać nie tylko całe programy, ale także ich poszczególne, odrębne części (np. procedury typu SUBROUTINE w języku FORTRAN). W szczególnym przypadku program napisany w jednym języku może odwoływać się do procedur zdefiniowanych w innym języku, bądź też korzystać z podprogramów bibliotecznych. W rezultacie użytkownik może mieć do dyspozycji kilka modułów wynikowych o rozszerzeniu .OBJ i kilka podprogramów bibliotecznych o rozszerzeniu .LIB. Aby uzyskać na ich podstawie zbiór binarny, będący programem nadającym się do bezpośredniego uruchamiania, należy użyć programu konsolidatora systemowego PC DOS o nazwie LINK. Oprócz tego, program LINK pozwala na uzyskiwanie mapy przydziału pamięci, pomocnej przy bezpośrednim uruchamianiu programu, a także wyświetla na ekranie monitora komunikaty o błędach (wynikających najczęściej z niezdefiniowania pewnych procedur lub funkcji wewnątrz podanego dla programu LINK ciągu zbiorów o rozszerzeniu .OBJ i .LIB).

Podczas pracy program LINK może wykorzystywać całą dostępną pamięć operacyjną, która niekiedy okazuje się zbyt mała. Wówczas na urządzeniu roboczym utworzony zostaje tymczasowy, pomocniczy zbiór pośredni o nazwie VM.TMP, który w momencie zakończenia pracy przez program LINK zostaje automatycznie skasowany. O utworzeniu takiego zbioru użytkownik powiadamiany jest następującym komunikatem, wyświetlanym na ekranie monitora:

VM.TMP has been created
Do not change diskette in drive x

co ma zapobiec jakimkolwiek manipulacjom z urządzeniem roboczym (np. usuwaniu dyskietki z roboczej stacji dysków). Należy pamiętać o tym, że jeśli na urządzeniu roboczym istnieje zbiór o nazwie VM.TMP, to zostanie on skasowany podczas tworzenia przez program LINK tymczasowego, pomocniczego zbioru pośredniego.

9.2. Opisy zbiorów wejściowych i wynikowych programu LINK.

Zbiory wejściowe o rozszerzeniu .OBJ

Zbiorami wejściowymi dla programu LINK są zbiory utworzone przez kompilatory (lub asemblery) o rozszerzeniu .OBJ (rozszerzenie może być inne, ale zbiory te muszą mieć format zbiorów wyjściowych z kompilatorów lub asemblerów). Te zbiory, które będą poddane obróbce przez program LINK, muszą zostać wymienione w linii komendy w ten sposób, że ich nazwy są od siebie odseparowane spacjami lub znakami plusa '+'. Przy wymienianiu nazw zbiorów nie trzeba podawać ani urządzenia, na których znajdują się (o ile jest to urządzenie robocze), ani ich rozszerzenia. Ta ostatnia uwaga odnosi się tylko do zbiorów o rozszerzeniu .OBJ. Jeśli zbiorem wejściowym jest zbiór o innym rozszerzeniu niż .OBJ, to musi zostać on wyspecyfikowane. Nazwy zbiorów wejściowych nie mogą rozpoczynać się znakiem @ (jest to znak o specjalnym znaczeniu - patrz poniżej).

W przypadku, gdy wyspecyfikowany zbiór nie znajduje się na podanym urządzeniu roboczym to:

1. Jeśli urządzeniem tym jest dysk twardy, to na ekranie monitora pojawia się komunikat:

Cannot find file object module

Należy wówczas przerwać sesję pracy z programem LINK.

2. Jeśli urządzeniem tym jest stacja dyskietek, to wykonanie programu zostaje zawieszone, a na ekranie monitora pojawia się następujący komunikat:

Cannot find file object module
change diskette <hit ENTER>

Należy wówczas założyć właściwą dyskietkę i potwierdzić to wciśnięciem klawisza 'Enter'. Pozwala to na wymianę kolejnych dyskietek, zawierających wymagane zbiory wejściowe. Wymiana taka jest jednak możliwa tylko wtedy, gdy program LINK nie utworzył wcześniej na tej dyskietce pomocniczego, tymczasowego zbioru pośredniego VM.TMP. Wówczas sesja pracy z programem LINK zostaje przerwana a na ekranie monitora pojawia się następujący komunikat:

Cannot find file object module

Zbiory wejściowe o rozszerzeniu .LIB:

Zbiory o rozszerzeniu .LIB są to skompilowane procedury realizujące pewne określone funkcje biblioteczne. Użytkownik może korzystać z nich w dwojaki sposób:

- wyszczególniając w linii komendy programu LINK określone zbiory o rozszerzeniu .LIB w ten sposób, że ich nazwy będą oddzielone od siebie spacjami lub znakami plusa '+'. Ten sposób pozwala na wyszczególnienie od jednego do ośmiu zbiorów o rozszerzeniu .LIB. Jeśli wyszczególniono dwa lub więcej zbiorów o takich samych nazwach (znajdujących się na np. różnych urządzeniach), wówczas konsolidator dołączy do zbioru wynikowego ten zbiór, który pierwszy wymieniono w specyfikacji komendy.
- odpowiadając, na pytanie konsolidatora o zbiory biblioteczne, tylko wciśnięciem klawisza 'Enter'. Konsolidator domyślnie dołączy wówczas odpowiednie zbiory, znajdujące się w bibliotece systemowej na urządzeniu roboczym. Jeśli ich tam nie znajdzie, wtedy dokona przeglądu biblioteki na urządzeniu wskazanym przez kompilator. Automatyczne przeglądy bibliotek nie są podejmowane przy konsolidacji programów uzyskanych z makroasemblera ASMB6.

Jeżeli program LINK nie znalazł odpowiedniego zbioru bibliotecznego, to na ekranie monitora pojawia się następujący komunikat:

Cannot find library A:library file
Enter new drive letter:

po którym użytkownik powinien podać urządzenie, zawierające wskazany zbiór biblioteczny.

Zbiór wynikowy o rozszerzeniu .EXE:

Pierwszym ze zbiorów wynikowych programu LINK jest zbiór o rozszerzeniu .EXE. Jest to program w języku maszyny, nadający się do bezpośredniego uruchamiania. Użytkownik może podać własną nazwę dla tego programu, a może też w specyfikacji komendy programu LINK pominąć ją. W tym przypadku zbiór zostanie nazwany tak tak jak pierwszy wymieniony moduł wejściowy. Nie ma natomiast możliwości zmiany rozszerzenia zbioru wynikowego poprzez odpowiednią specyfikację w linii komendy wywołującej program LINK. Bez względu na żądane rozszerzenie będzie ono zawsze .EXE.

Zbiór wynikowy o rozszerzeniu .MAP:

Drugim ze zbiorów wynikowych programu LINK jest zbiór o rozszerzeniu .MAP. Jest on tworzony tylko na wyraźne życzenie użytkownika, polegające na podaniu nazwy tego zbioru. Domyślnie konsolidator nie tworzy tego zbioru. Zbiór o rozszerzeniu .MAP zawiera m.in. listę symboli publicznych (PUBLIC, patrz opis assemblera ASMB6) nazw i etykiet z przyporządkowanymi im adresami (zależnie od wyszczególnionych w linii komendy parametrów pracy konsolidatora LINK).

9.3. Parametry pracy programu LINK.

Aby wyjaśnić znaczenie niektórych parametrów programu LINK należy zdefiniować pewne pojęcia z nazewnictwa assemblera ASMB6.

Segment:

Segment jest to ciągły obszar pamięci o długości do 64 kB, który można ulokować od dowolnego miejsca w pamięci o adresie podzielonym przez 16. To ostatnie założenie wynika ze sposobu adresowania komórek w pamięci 1 megabajtowej – adres fizyczny tworzy suma 16-bitowego adresu (tzw. offsetu), badającego kolejnym numerem komórki pamięci względem początku segmentu i zawartości rejestru segmentowego pomnożonego przez 16. Kompilator lub assembler nie określają ani położenia zbioru wynikowego wewnątrz pamięci, ani zawartości jej poszczególnych komórek, gdyż wartości te mogą ulec zmianie w procesie konsolidacji. Mikroprocesor 8086 (8088) zapewnia jednoczesną możliwość adresacji 4 różnych segmentów, poprzez cztery rejestry segmentowe: CS, DS, ES i SS. Segmenty mogą zachodzić na siebie.

Grupa:

Grupa jest zbiorem segmentów umieszczonych wewnątrz 64 kbajtów pamięci od komórki o adresie podzielonym przez 16 (czyli sama jest segmentem). Poszczególne segmenty przyporządkowuje do danej grupy asembler lub kompilator. Dany program może składać się z jednej lub kilku grup. Przyporządkowanie kilku segmentów do jednej grupy pozwala na adresowanie komórek pamięci przy wykorzystaniu jednego rejestru segmentowego i offsetu określającego położenie względem początku grupy.

Klasa:

Klasa jest zbiorem segmentów. Przyporządkowanie segmentów do danej klasy jest podstawą uporządkowania i wzajemnego położenia segmentów względem siebie w pamięci operacyjnej. Nazwę klasy określa asembler lub kompilator. Wszystkie części programu, przyporządkowane do tej samej klasy, zostają umieszczone w sposób ciągły w pamięci. O uporządkowaniu segmentów względem siebie decyduje kolejność wyspecyfikowania zbiorów wejściowych w linii komendy programu LINK. Klasa nie ma ograniczonego rozmiaru (poza ograniczeniem wynikającym z pojemności samej pamięci) i jest podzielona na grupy.

Parametry programu LINK:

Działanie programu LINK można wymusić podaniem w linii komendy odpowiednich parametrów. Decydują one o położeniu kodu maszynowego programu wewnątrz pamięci operacyjnej, a także pozwalają na uzyskiwanie pewnych informacji, użytecznych przy uruchamianiu programu. Wszystkie wymienione parametry konsolidator identyfikuje poprzez ich pierwszą literę, poprzedzoną znakiem '/'.

Program LINK rozróżnia następujące parametry:

1. Parametr /DSALLOCATION (/D) nakazuje umieścić wszystkie dane, przyporządkowane do grupy danych, na końcu tej grupy. Jeśli dodatkowo użyto parametru /HIGH, wówczas cała dostępna pamięć, poniżej pola zajętego przez dane, może być przyporządkowywana dynamicznie i adresowana przez ten sam rejestr segmentowy. Jeśli nie podano parametru /D, wówczas konsolidator umieści dane, przyporządkowane do grupy danych, na początku grupy (od adresu o offsecie 0).
2. Parametr /HIGH (/H) nakazuje umieścić kod programu od najwyższego możliwego adresu w pamięci operacyjnej, tak jednak, aby nie nałożyć go na część nietrwałą programu COMMAND.COM (patrz rozdział pt. 'STRUKTURA SYSTEMU OPERACYJNEGO PC DOS'). Jeśli parametr /HIGH nie zostanie użyty, wówczas konsolidator umieści kod programu od

najniższego możliwego adresu w pamięci.

3. Parametr /LINE (/L) nakazuje umieścić w zbiorze wyjściowym, o rozszerzeniu .MAP, adresy komórek pamięci zawierających pierwsze bajty wyszczególnionych w programach źródłowych instrukcji (np. oznaczonych etykietami instrukcji asemblera ASMB6). Opcja ta działa dla programów napisanych tylko w niektórych językach.
4. Parametr /MAP (/M) nakazuje umieścić w zbiorze wyjściowym, o rozszerzeniu .MAP listę wszystkich publicznych (PUBLIC) symboli zdefiniowanych w zbiorach wejściowych wraz z podaniem ich adresów (offsetów) wewnątrz kodu wynikowego programu umieszczonego w pamięci operacyjnej.
5. Parametr /PAUSE (/P) pozwala na zmianę dyskietki w urządzeniu, na którym ma zostać zapisany zbiór wyjściowy .EXE. Konsolidator zawiesza wówczas pracę, wyświetlając następujący komunikat:

About to generate .EXE file
Change disks <hit enter>

i czeka na wymianę dyskietki, potwierdzoną wciśnięciem klawisza 'Enter' (o tym na którym urządzeniu ma zostać zapisany zbiór wyjściowy decyduje odpowiednia specyfikacja w linii komendy wywołującej program LINK).

6. Parametr /STACK:rozmiar pozwala na dziesiętne zdefiniowanie rozmiarów stosu ('rozmiar' może być liczbą od 1 do 65535, przy czym podanie liczby większej od zera, ale mniejszej od 512 zawsze spowoduje przyjęcie za rozmiar liczby 512). Jeśli nie podano parametru /S, to program LINK przyjmie za rozmiar stosu wartość podaną przez kompilator lub asembler (podczas uruchomienia programu może to spowodować trudne do przewidzenia skutki). W przypadku, gdy podany przez użytkownika, kompilator, lub asembler rozmiar stosu jest liczbą nieparzystą, to podczas wykonywania konsolidacji program LINK zmniejszy go o 1. Jeśli użytkownik zamierza podać program wynikowy .EXE przekształceniu na postać relokalną .COM (rozkaz EXE2BIN systemu operacyjnego PC DOS), to przynajmniej jeden moduł wejściowy .OBJ musi zawierać wyrażenie ustawiające rozmiar stosu. Wyrażenie to automatycznie generują kompilatory, natomiast w programach napisanych w języku asemblera ASMB6 należy dodatkowo zdefiniować segment stosu (STACK SEGMENT). Brak tego segmentu nie jest błędem, ale powoduje wyświetlenie komunikatu:

Warning: No Stack statement

7. Parametr /X pozwala określić całkowitą liczbę segmentów, które może zawierać zbiór wyjściowy .EXE. Może ona zmieniać się od 0 do 1024 (wartość domyślna wynosi 256) i odpowiada liczbie oddzielnych segmentów, zawartych we wszystkich zbiorach źródłowych .OBJ i .LIB. Chociaż granica całkowitej liczby segmentów może być 1024 to praktycznie liczba ta może wynosić co najwyżej 1000, o ile nie są to tzw. segmenty absolutne (ang. absolute segments) - segmenty, których położenie w pamięci operacyjnej jest z góry określone).
8. Parametr /O pozwala na wykonywanie konsolidacji programów wynikowych .OBJ, utworzonych przez wersję 1 kompilatora języka Pascal lub przez wersję 1 kompilatora języka FORTRAN.

9.4. Sesja pracy z programem LINK.

Uruchamianie programu LINK i podawanie informacji niezbędnych dla jego pracy może odbywać się w następujący sposób:

W trybie konwersacji pytanie/odpowiedź:

Użytkownik wywołuje program LINK rozkazem wprowadzonym z klawiatury:

LINK

Po załadowaniu się do pamięci operacyjnej program LINK wyświetla na ekranie monitora szereg pytań, na które użytkownik kolejno udziela odpowiedzi:

Pytania programu LINK: Odpowiedzi użytkownika:

[illegible]

```
Run File [nazwa1.EXE]: [[d:][ścieżka]nazwa3[.rozszerzenie3]
                        [parametry]].
```

```
List File [NUL.MAP]:      [[d:][ścieżka]nazwa4[.rozszerzenie4]
                           [parametry]]
```

```
Libraries [LIB]:      [d:]([ścieżka]nazwa5[.rozszerzenie5]  
                        [+([d:]([ścieżka]nazwa6[.rozszerzenie6]))...]
```

• Po udzieleniu odpowiedzi na zadane pytanie, użytkownik przed wciśnięciem klawisza 'Enter' może, po przecinku, od razu, nie czekając na kolejne pytanie, udzielić na nie odpowiedzi. Jeśli wyspecyfikowaną odpowiedź użytkownik zakończył znakiem ';', program LINK za odpowiedzi na kolejne pytania będzie przyjmował wartości domniemane.

W trybie podawania informacji w jednej linii komendy:

```
LINK objlist,runfile,mapfile,liblist[parameter]...
```

gdzie:

- objlist określa zbiory (w formacie zbiorów wyjściowych z kompilatorów lub asemblerów), oddzielonych od siebie spacjami lub znakami plus '+'.
- runfile określa zbiór .EXE (program binarny) nadający się do bezpośredniego uruchamiania,
- mapfile określa zbiór będący mapą przydziału pamięci, tworzony przez program LINK,
- liblist określa zbiory (w formacie zbiorów z rozszerzeniem .LIB), oddzielonych od siebie spacjami lub znakami plus '+'.

Linia komendy może być niekompletna, tzn. może brakować pewnych informacji dotyczących zbiorów wyjściowych. Wówczas:

1. Jeśli linie komendy użytkownik zakończył znakiem ';', to konsolidator podstawia w miejsce pominiętych specyfikacji ich wartości domyślne (z wyjątkiem przypadku opisanego w punkcie 3).

2. Jeśli linia komendy nie została zakończona znakiem ';', to konsolidator przechodzi do trybu konwersacyjnego z użytkownikiem i wyświetla na ekranie monitora pytania o informacje dotyczące zbiorów pominiętych w linii komendy.
3. Jeśli w miejscu przeznaczonym na specyfikację pewnych zbiorów wyjściowych (np. zbioru .MAP) użytkownik postawił przecinek (,), to konsolidator utworzy ten zbiór z właściwym rozszerzeniem i nazwą pierwszego podanego zbioru w formacie .OBJ.

Przykłady:

- LINK MODULE - zbiorem wejściowym jest zbiór MODULE z domyślnym rozszerzeniem .OBJ. Konsolidator wyświetli kolejno pytania o zbiory .EXE, .MAP i .LIB.
- LINK MODULE; - zbiorem wejściowym jest zbiór MODULE z domyślnym rozszerzeniem .OBJ. Konsolidator utworzy zbiór MODULE.EXE, natomiast pominie zbiór o rozszerzeniu .MAP.
- LINK MODULE,;; - zbiorem wejściowym jest zbiór MODULE z domyślnym rozszerzeniem .OBJ. Konsolidator utworzy zbiory: MODULE.EXE i MODULE.MAP.
- LINK MODULE,, - zbiorem wyjściowym jest zbiór MODULE z domyślnym rozszerzeniem .OBJ. Konsolidator utworzy zbiór MODULE.EXE i zada pytanie dotyczące zbioru .MAP.
- LINK MODULE,,NUL; - zbiorem wyjściowym jest zbiór MODULE z domyślnym rozszerzeniem .OBJ. Konsolidator utworzy tylko zbiór MODULE.EXE i nie będzie wyświetlał żadnych pytań.

W trybie automatycznego podawania informacji:

LINK @[d:][ścieżka]nazwa[.rozszerzenie]

Sposób ten jest szczególnie wygodny wtedy, gdy wielokrotnie zachodzi potrzeba wywoływania programu LINK dla konsolidacji tych samych zbiorów, z tymi samymi parametrami. Przed użyciem tej opcji należy utworzyć zbiór (na urządzeniu d:, z nazwą i ewentualnym rozszerzeniem, podawanymi w linii komendy po znaku @), który będzie zawierał linię tekstu, będącego kolejnymi informacjami dla programu LINK.

Przykład:

Niech zbiory RESP1 i RESP2 mają następującą postać:

RESP1: MODA+MODB+MODC+
MODD+MODE+MODF

RESP2: runfile/p

Wówczas rozkaz:

```
LINK @RESP1+mymod,@resp2;
```

jest równoważny rozkazowi:

```
LINK MODA+MODB+MODC+MODD+MODE+MODF+mymod,runfile/p;
```

Konsolidator ze zbiorów: MODA, MODB, MODC, MODD, MODE, MODF oraz mymod, o domyślnym rozszerzeniu .OBJ, utworzy tylko jeden zbiór wyjściowy runfile.EXE. Parametr /P umożliwi zmianę dyskiety w urządzeniu roboczym, na której zostanie zapisany zbiór runfile.EXE.

U W A G A :

Wykonywanie programu LINK można zawsze przerwać wciskając kombinację klawiszy 'Ctrl/Break'.

W przypadku wyspecyfikowania parametru /LINE zbiór wynikowy .MAP może zostać przesłany bezpośrednio na drukarkę (urządzenie PRN). Ilustruje to poniższy przykład pełnej sesji pracy z programem LINK ('Enter' oznacza wciśnięcie klawisza):

```
B:LINK 'Enter'
```

```
IBM Personal Computer Linker  
Version 2.30 (C)Copyright IBM Corp. 1981, 1985
```

```
Object Modules [.OBJ]: example 'Enter'  
Run File [EXAMPLE.EXE]: /map 'Enter'  
List File [NUL.MAP]: prn/line 'Enter'  
Libraries [B:] 'Enter'
```

10. PROGRAM DEBUG (URUCHAMIANIE PROGRAMÓW W JEZYKU ASMB6).

Program DEBUG jest programem systemowym PC DOS, pomocnym przy uruchamianiu innych programów, w szczególności napisanych w języku asemblera ASMB6. Umożliwia on m.in.:

- wykonywanie programów w kodzie maszynowym,
- uruchamianie programów od określonego adresu, instrukcja po instrukcji, zmianę zawartości poszczególnych rejestrów i komórek pamięci,
- ładowanie do pamięci operacyjnej poszczególnych zbiorów, wyświetlanie ich zawartości na ekranie monitora i wykonywanie zmian w tych zbiorach.

10.1. Uruchamianie programu DEBUG.

Program DEBUG można uruchamiać następującym rozkazem:

```
DEBUG [d:][ścieżka][nazwa[.rozszerzenie]][parametr1][parametr2]
```

Użytkownik może zatem wywoływać program debuggera dwoma sposobami:

- z określeniem zbioru, z którym będzie odbywać się sesja pracy z programem DEBUG. Zbiór ten zostanie następnie ściągnięty do pamięci operacyjnej, co pozwala, od razu, poprzez wykonanie odpowiednich komend debuggera, wyświetlać go na ekranie monitora, wykonywać w nim zmiany, uruchamiać od określonego adresu etc,
- bez określania w linii komendy nazwy zbioru. Sesja pracy z programem DEBUG będzie odbywać się albo z aktualną zawartością pamięci operacyjnej, albo ze zbiorem ściągniętym do niej przy pomocy komend samego debuggera.

Opcjonalne parametry 1 i 2 są parametrami, których określenia wymaga program (zbiór), z którym będzie odbywać się sesja pracy z debuggerem, np.

DEBUG DISCOMP.COM A: B:

W momencie uruchomienia programu debuggера do rejestrów mikroprocesora zostaną wpisane następujące wartości:

- do rejestrów segmentowych CS, DS, ES i SS wartość adresu początku pierwszego wolnego segmentu po programie DEBUG, przesunięta o cztery bity w prawo (segment, z założenia, zawsze zaczyna się od adresu podzielnego przez 16),
- do wskaźnika instrukcji IP wartość 0100H,
- do wskaźnika stosu SP wartość: albo określająca koniec segmentu, albo określająca koniec nietrwałej części programu loadera (w zależności od tego która z tych wartości jest mniejsza).
- do pozostałych rejestrów tj. do AX, BX, CX, DX, BP, SI i DI wartość 0000H. Jeśli w linii komendy wywołującej program DEBUG podano, z jakim zbiorem odbędzie się sesja pracy, wówczas rejestr CX zawierać będzie długość tego zbioru w bajtach (jeśli zbiór jest dłuższy niż 64kB, to długość programu będzie wpisana do rejestrów BX i CX w ten sposób, że rejestr BX będzie zawierał bardziej znaczącą część wpisywanej wartości).
- do rejestru wskaźników będą wpisane następujące wartości flag: NV, UP, EI, PL, NZ, NA, PO, NC (ich znaczenie ilustruje tabela umieszczona w podrozdziale opisującym komendę REGISTER).

Zbiory o rozszerzeniu .COM i .EXE są zawsze ładowane do pamięci operacyjnej bezpośrednio po programie debuggера. Ich załadowanie poprzedza ustawienie tzw. PSP (Program Segment Prefix). Jeśli zbiór ładowany do pamięci operacyjnej ma rozszerzenie .EXE, wówczas zostaje wykonana jego niezbędna realokacja połączona z ustawieniem zdefiniowanych w tym zbiorze wartości rejestrów segmentowych, wskaźnika stosu i wskaźnika instrukcji. Zbiory z rozszerzeniem .HEX (są to zbiory kodowane w formacie INTEL-HEX) przed załadowaniem do pamięci operacyjnej zostają odpowiednio rozkodowane.

10.2. Ogólne informacje dotyczące komend programu debuggera.

1. Wszystkie komendy debuggera DEBUG są wywoływane poprzez podanie pojedynczej litery.
2. Komendy i parametry mogą być pisane małymi, dużymi lub kombinacją małych i dużych liter.
3. Komendy i parametry mogą być oddzielone separatorami. Separatory są zawsze wymagane pomiędzy dwoma kolejnymi wartościami hexadecymalnymi, np. trzy poniższe komendy są sobie równoważne:

```
D CS:100 110
Dcs:100 110
D,CS:100,110
```

4. Debugger rozpoczyna wykonywanie każdej komendy po wciśnięciu klawisza 'Enter'.
5. Wykonywanie każdej komendy można przerwać wciskając kombinację klawiszy 'Ctrl/Break'.
6. Jeśli dana komenda powoduje wyświetlenie na ekranie monitora dużej ilości zbędnych informacji, to można zawiesić to wyświetlanie wciskając kombinację klawiszy 'Ctrl/NumLock' (zawieszenie wyświetlania nie zawiesza wykonywania samej komendy). Powrót do poprzedniego stanu następuje po wciśnięciu klawisza dowolnej litery.
7. Program debuggera zezwala na używanie wszystkich klawiszy funkcyjnych, akceptowanych przez system operacyjny PC DOS.
8. Błąd syntaktyczny w składni linii komendy sygnalizuje następujący komunikat, wyświetlany na ekranie monitora:

^error

9. Program debuggera. każdorazowo zgłasza gotowość przyjęcia komendy znakiem '-', wyświetlanym w pierwszej kolumnie.
10. W opisie poszczególnych komend ich części opcjonalne zostały umieszczone w nawiasach kwadratowych [], natomiast informacje zawarte w nawiasach zwykłych () zawsze odnoszą się do pojedynczego parametru.

10.3. Komenda A (ASSEMBLE).

Postać komendy:

A[adres]

Opis:

Komenda ASSEMBLE przekształca instrukcje napisane w języku makroassemblera ASM86 na odpowiadające im postaci hexadecymalne. Następnie tak uzyskany kod zostaje wpisany do pamięci operacyjnej od adresu określonego parametrem 'adres'. Parametr ten jest opcjonalny i jeśli nie zostanie podany, to:

- debugger wpisze hexadecymalny kod instrukcji od adresu określonego ostatnią wykonaną komendą,
- jeśli nie wykonywano dotychczas żadnej komendy debuggera, wpisywanie rozpocznie się od adresu CS:0100H.

Wyspecyfikowanie każdej instrukcji w języku makroassemblera ASM86 musi zostać zakończone wciśnięciem klawisza 'Enter'. Debugger, po wykonaniu translacji, i załadowaniu kodu do pamięci zgłasza się wypisaniem na ekranie monitora adresu kolejnej komórki pamięci. Użytkownik może wówczas:

- wyspecyfikować następną instrukcję, która chce poddać asemblacji,
- zakończyć tryb pracy asemblacji wciskając klawisz 'Enter'. Debugger zgłasza wówczas gotowość przyjęcia dowolnej komendy wyświetleniem znaku '-'.

Komenda ASSEMBLE może być stosowana w zasadzie do wszystkich instrukcji (instrukcji, nie dyrektyw) makroasemblera ASMB6. Wszelkie wyjątki wynikają z faktu, że debugger ma do czynienia z pojedynczymi instrukcjami, a nie z całym programem (ograniczenia wymieniono w instrukcji fabrycznej). Dodatkowo można korzystać z dwóch 'najpopularniejszych' pseudoinstrukcji: DB i DW (define byte, define word).

Przykład:

```
C>debug
a 200
08B4:0200 XOR AX,AX
08B4:0202 MOV [BX],AX
08B4:0204 RET
08B4:0205
```

10.4. Komenda C (COMPARE).

Postać komendy:

C adres1 L (liczba bajtów hexadecymalnie) adres3

lub:

C adres1 adres2 adres3

Opis:

Komenda COMPARE porównuje bajt po bajcie zawartość dwóch bloków pamięci. Specyfikacja linii komendy musi zawsze zawierać adresy początkowe obu bloków (parametry: 'adres1', 'adres3') oraz zakres wykonywania porównania. Zakres ten można określić podając:

- albo adres końca pierwszego bloku (parametr adres2),
- albo długość pierwszego bloku (parametr 'liczba bajtów hexadecymalnie' po literze L).

Wykrycie różnicy pomiędzy obydwoma blokami debugger sygnalizuje, wyświetlając adresy komórek pamięci zawierających różne wartości wraz z podaniem tych wartości:

adr1 bajt1 adr2 bajt2

Jeśli w parametrach 'adres1' i 'adres3' użytkownik poda tylko wartości offsetów, to debugger domyślnie przyjmie, że adres fizyczny ma zostać wyznaczony w oparciu o rejestr segmentowy DS. Należy pamiętać, że fizyczny adres końca pierwszego bloku debugger wyznacza zawsze na podstawie tego samego rejestru segmentowego (lub tej samej wartości podawanej przed dwukropkiem), co przy wyznaczaniu fizycznego adresu początku tego bloku - stąd wystarczy podać tylko offset.

Przykład:

C 100 L20 200

Zawartość kolejnych 32 (hexadecymalnie 20) 1-bajtowych komórek pamięci od adresu DS:100 zostanie porównana z zawartością kolejnych 32 komórek od adresu DS:200. Powyższą komendę można podać w równoważnej postaci:

C 100 11F 200

10.5. Komenda D (DUMP).

Postać komendy:

D adres

lub:

D adres1 adres2

lub:

D adres1 L (liczba bajtów hexadecymalnie)

Opis:

Komenda DUMP wyświetla na ekranie monitora zawartość bloku pamięci w następującym formacie: zawartość poszczególnej komórki jest podawana hexadecymalnie i poprzez odpowiadający jej w kodzie ASCII znak. Jeśli kod odpowiada znakowi niedrukowalnemu (np. znakowi sterującemu), wówczas zamiast niego debugger wyświetla znak kropki '.'. Dodatkowo na początku każdej linii program DEBUG zamieszcza adres komórki pamięci, od której rozpoczyna się wyświetlanie w tej linii. W każdej linii można wyświetlić zawartość osmiu lub szesnastu 1-bajtowych komórek pamięci (w zależności od formatu wyświetlania: 40 lub 80 kolumnowego). Komendę DUMP można wywoływać w następujący sposób:

- podając tylko nazwę komendy, lub nazwę komendy z adresem:

D

lub:

D adres

Debugger wyświetli, w zależności od formatu wyświetlania, zawartość kolejnych 40 (lub 80) komórek pamięci od podanego adresu, lub od adresu wynikającego z ostatnio wykonanych komend debugera (jeśli nie wykonywano dotychczas żadnych komend to od adresu o offsecie 100H).

- podając nazwę komendy i zakres wyświetlania:

D adres1 L (liczba bajtów hexadecymalnie)

lub:

D adres1 adres2

Debugger wyświetli zawartość bloku pamięci od adresu określonego parametrem 'adres1' aż do końca podanego zakresu. Zakres można określić podając: albo adres końcowy (parametr 'adres2'), albo liczbę komórek pamięci (parametr 'liczba bajtów hexadecymalnie').

Jeśli któryś z parametrów 'adres' zawiera tylko offset (bez specyfikacji rejestru segmentowego), to debugger domyślnie obliczy adres fizyczny na podstawie zawartości rejestru segmentu danych DS.

Przykład:

D CS:100 10C

Debugger wyświetli zawartość pamięci od adresu CS:100 do adresu CS:10C. Komenda ta jest równoważna komendzie:

D CS:100 L 0D

10.6. Komenda E (ENTER).

Postać komendy:

E adres [lista]

Opis:

Komenda ENTER pozwala na zastępowanie zawartości pamięci RAM, od adresu określonego wartościami wymienionymi w linii komendy (parametr 'list') lub wartościami wpisywanymi bezpośrednio z klawiatury (bez parametru 'list'). Jeśli parametr 'adres' zawiera tylko offset adresu, to debugger domyślnie przyjmie, że adres fizyczny ma zostać wyznaczony w oparciu o rejestr segmentowy DS. Komenda ENTER może być wykorzystywana w dwojaki sposób:

1. Z podaniem po parametrze 'adres' parametru 'lista':

Spowoduje to wypełnienie komórek pamięci, od podanego adresu, wartościami wymienionymi w parametrze 'lista', np.

E CS:100 F3 "ABC" 8D

spowoduje, że do kolejnych komórek pamięci RAM od adresu fizycznego obliczonego na podstawie zawartości rejestru segmentowego CS i offsetu 100H będą wpisane następujące wartości hexadecymalne: F3, 41, 42, 43 (kolejne kody ASCII liter A, B, C), 8D.

2. Bez podawania parametru 'lista' w linii komendy.

Debugger wyświetli wówczas zawartość komórki pamięci o adresie podanym w linii komendy i oczekuje na odpowiedź użytkownika:

- wciśnięcie klawisza 'Enter' spowoduje zakończenie wykonywania komendy ENTER i zgłoszenie przez debugger oczekiwania na następną instrukcję,
- wciśnięcie klawisza spacji spowoduje wyświetlenie zawartości komórki pamięci o adresie o jeden większym,
- wciśnięcie klawisza ze znakiem '-' spowoduje wyświetlenie zawartości komórki pamięci o adresie o jeden mniejszym,
- jeśli po wyświetleniu informacji o zawartości komórki pamięci użytkownik poda jednobajtową liczbę hexadecymalną, a następnie wciśnie jeden z podanych powyżej klawiszy, to debugger zastąpi tą liczbą dotychczasową wartość przechowywaną w tej komórce, a następnie wykona funkcję związaną z wciśniętym klawiszem.

10.7. Komenda F (FILL).

Postać komendy:

F adres1 adres2 lista

lub:

F adres1 L (liczba bajtów hexadecymalnie) lista

Opis:

Komenda FILL pozwala na wpisanie do kolejnych komórek pamięci od adresu określonego parametrem 'adres1' wartości określonych parametrem 'lista'. Zakres wpisywania można określić dwoma sposobami, podając:

- albo adres ostatniej komórki, z której będą odczytywane dane do przepisania (parametr 'adres2'),

- albo długość bloku pamięci, który będzie przepisywany (parametr 'liczba bajtów hexadecymalnie', koniecznie po literze L).

Jeśli lista zawiera mniej wartości jednobajtowych niż podany zakres wpisywania, to wpisywanie będzie powtarzane do wypełnienia całego zakresu. Parametr 'adres1' może zawierać tylko offset adresu - wówczas debugger wyznaczy adres fizyczny w oparciu o wartość rejestru segmentowego DS.

Przykład:

F 4BA:100 L 9 F3 "ABC" 8D

Wykonanie powyższego polecenia spowoduje wpisanie do kolejnych dziewięciu komórek pamięci od adresu 4BA:100 (adres fizyczny 4CA0H) kolejno następujących wartości: F3, 41, 42, 43, 8D, F3, 41, 42, 43.

10.8. Komenda G (G0).

Postać komendy:

G [=adres1] adres2 [adres3]....

Opis:

Komenda G0 pozwala na wykonanie programu do instrukcji (włącznie z tą instrukcją), której pierwszy bajt znajduje się w komórce pamięci o adresie określonym przez parametr 'adres2'. Wykonywanie programu rozpocznie się od instrukcji, której pierwszy bajt znajduje się:

- albo w komórce pamięci o adresie wyznaczonym przez aktualną zawartość rejestrów CS i IP (jeśli nie podano parametru 'adres1' poprzedzonego znakiem '='),
- albo w komórce pamięci o adresie określonym przez parametr 'adres1' (parametr ten musi być poprzedzony znakiem '=').

Po wykonaniu ostatniej instrukcji debugger wyświetla zawartość poszczególnych rejestrów, stan wszystkich wskaźników (flag) oraz mnemonikę następnej do wykonania instrukcji.

W linii komendy GO można w dowolnej kolejności podać do 10 adresów komórek pamięci zawierających pierwsze bajty różnych instrukcji. Zawieszenie wykonywania programu nastąpi w chwili wykonania pierwszej takiej napotkanej instrukcji. Każdy z parametrów: 'adres1', 'adres2', ... może zawierać tylko offset - debugger wyznaczy wówczas adres fizyczny w oparciu o zawartość rejestru segmentowego CS.

U W A G A :

1. Parametry 'adres1', 'adres2',... muszą zawsze określać adres pierwszego bajtu instrukcji.
2. Przed wykonaniem komendy GO na stosie musi być miejsce przynajmniej na 6 bajtów.

Niespełnienie powyższych warunków zawsze spowoduje trudne do przewidzenia skutki.

10.9. Komenda H (HEXARITHMETIC).

Postać komendy:

H (liczba 1 hexadecymalnie) (liczba 2 hexadecymalnie)

Opis:

Komenda HEXARITHMETIC pozwala na obliczenie sumy i różnicy dwóch liczb hexadecymalnych.

Przykład:

H OF 8

Powyższe polecenie spowoduje, że na ekranie monitora zostaną wyświetlone w jednej linii dwie liczby hexadecymalne:

17 07

z których pierwsza jest sumą dwóch liczb hexadecymalnych 0F i 8, a druga - ich różnicą.

10.10. Komenda I (INPUT).

Postać komendy:

I (adres portu)

Opis:

Przy pomocy komendy INPUT można odczytać, a następnie wyświetlić na ekranie monitora hexadecymalną zawartość jedno lub dwubajtowego portu o podanym adresie.

Przykład:

I 2FB
6B - z portu o adresie 2FB odczytano hexadecymalną wartość 6B.

10.11. Komenda L (LOAD).

Postać komendy:

L [adres [(oznaczenie dysku) sektor (liczba sektorów)]

Opis:

Komenda LOAD pozwala na wczytanie do pamięci operacyjnej danych zapisanych na nośniku dyskowym. Maksymalnie pojedynczą instrukcją LOAD można wczytać dane z 128 (80H) sektorów dyskowych, każdy o pojemności 512 bajtów. Jeśli podczas przepisywania wystąpią błędy odczytu z dysku (program DEBUG wyświetla wówczas odpowiedni komunikat), operacja może zostać powtórzona poprzez kolejne wciśnięcie klawiszy: 'F3' i 'Enter'. Komendę LOAD można używać w następujący sposób:

1. Podając wszystkie opcjonalne parametry:

L adres (oznaczenie dysku) sektor (liczba sektorów)

gdzie:

- parametr 'adres' określa adres komórki pamięci, od której rozpocznie się wczytywanie. Jeśli 'adres' określa tylko offset, to debugger dla określenia adresu fizycznego wykorzysta zawartość rejestru segmentowego CS,
- parametr 'oznaczenie dysku' określa stację dyskową, na której znajduje się nośnik z danymi (0 oznacza stację A:, 1 - stację B: itd.),
- parametr 'sektor' określa hexadecymalny numer sektora dyskowego, od którego dane będą odczytywane,
- parametr 'liczba sektorów' określa hexadecymalną liczbę sektorów, z których dane będą przepisywane do pamięci operacyjnej (max. 128 sektorów).

2. Podając tylko parametr 'adres', lub nie podając żadnego parametru:

L adres

lub:

L

W tym przypadku debugger przepisze do pamięci operacyjnej od adresu CS:100 (lub od adresu określonego parametrem 'adres') zawartość całego zbioru, którego nazwę i położenie (urządzenie i katalog w którym znajduje się ten zbiór) podano albo podczas wywoływania programu DEBUG, albo przez wykonanie odpowiedniej komendy NAME. Jeśli przepisywanymi zbiorami są zbiory o rozszerzeniu .EXE lub .COM, to będą zawsze ładowane do pamięci od adresu CS:100, bez względu na parametr 'adres'.

Po zakończeniu przepisywania rejestry BX i CX zawierają liczbę przepisanych bajtów, co w przypadku zbiorów o rozszerzeniu .EXE i .COM jednocześnie określa rozmiar programu.

Przykład 1:

L 4BA:100 1 0F 6D

Powyższa komenda spowoduje odczytanie z dyskietki znajdującej się w urządzeniu B: 109 (6DH) kolejnych sektorów, poczynając od 15 (0FH) sektora i wpisanie odczytanych danych do pamięci operacyjnej od adresu 4CA0H (4BA:100).

Przykład 2:

DEBUG
-N MYPROG
-L
-

Powyższy ciąg komend spowoduje wywołanie programu debuggера, zdefiniowanie nazwy zbioru (komenda N), a następnie załadowanie go do pamięci operacyjnej od adresu CS:100.

10.12. Komenda M (MOVE).

Postać komendy:

M adres1 L (liczba bajtów hexadecymalnie) adres3

lub:

M adres1 adres2 adres3

Opis:

Komenda MOVE umożliwia przepisanie bajt po bajcie zawartości bloku pamięci rozpoczynającego się od komórki o adresie 'adres1' w inne miejsce, od adresu określonego parametrem 'adres3'. Zakres przepisywania można określić podając:

- albo adres końcowy bloku pamięci (parametr 'adres2'),
- albo długość bloku w bajtach (parametr 'liczba bajtów hexadecymalnie')

Podczas przepisywania nie ulegają zmianie wartości wpisane do komórek źródłowych, chyba że blok pamięci, będący źródłem danych częściowo pokrywa się z blokiem pamięci, do której dane będą wpisywane. Jeśli w parametrach: 'adres1' i 'adres3' podano tylko wartości offsetów, to debugger domyślnie przyjmie, że adres fizyczny ma zostać wyznaczony w oparciu o rejestr segmentowy DS. Należy pamiętać, że fizyczny adres końca bloku pamięci debugger zawsze wyznacza na podstawie tego samego rejestru segmentowego (lub tej samej wartości podawanej przed dwukropkiem), co początku tego bloku – stąd przy parametrze 'adres2' wystarczy podać tylko offset.

Przykład:

```
M CS:100 110 500
```

Powyższa komenda spowoduje, że zawartość 17 komórek pamięci od adresu CS:100 do adresu CS:110 zostanie przepisana do kolejnych komórek pamięci od adresu DS:500.

10.13. Komenda N (NAME).

Postać komendy:

```
N [d:]Iścieszka[nazwa].rozszerzenie]
```

Opis:

Komenda NAME, pozwalająca na określenie zbioru dla komend LOAD i WRITE, formatuje tzw. bloki kontrolne zbioru (ang. FCBs – File Control Blocks, patrz instrukcja FCBS w rozdziale pt. 'KONFIGUROWANIE SYSTEMU OPERACYJNEGO') dla pierwszych dwóch nazw zbiorów (od adresów: CS:5C i CS:6C; jeśli wywołano program DEBUG z podaniem zbioru, z którym odbywać się będzie sesja pracy, to także zostaje zaformatowany FCBs od adresu CS:5C). Pozwala to na określenie zbioru, z którym odbywać się będzie sesja pracy z debuggerem (jeśli nie określono tego zbioru podczas wywoływania debuggera) lub na zmianę wcześniej określonego zbioru.

Przykład 1:

```
DEBUG
-N A:MYPROG
-L
-
```

Powyższy ciąg komend spowoduje wywołanie programu DEBUG, zdefiniowanie nazwy zbioru, znajdującego się w katalogu roboczym na urządzeniu A:, a następnie załadowanie go do pamięci operacyjnej od adresu CS:100.

Przykład 2:

```
DEBUG A:MYPROG
-N file1 file2
```

W powyższym przykładzie debugger zostanie wywołany do pracy z programem MYPROG. Następnie komenda N spowoduje zaformatowanie dwóch FCB-sów: od adresu CS:5C dla zbioru 'file1' i od adresu CS:6C dla zbioru 'file2'. Dalsza sesja pracy z programem DEBUG może odbywać się tylko z tymi zbiorami.

10.14. Komenda O (OUTPUT).

Postać komendy:

O (adres portu) bajt

Opis:

Komenda OUTPUT pozwala na przesłanie wyspecyfikowanej jedno lub dwubajtowej liczby hexadecymalnej do portu o podanym adresie.

Przykład:

O 2FB 4F - hexadecymalna liczba 4FH zostanie przesłana do portu o adresie 2FBH.

10.15. Komenda P (PROCEED).

Postać komendy:

P [=adres] [liczba instrukcji hexadecymalnie]

Opis:

Komenda PROCEED pozwala na wykonywanie następujących instrukcji (ze zbioru instrukcji asemblera ASMB6): wywołanie podprogramu, wykonanie pętli, powtórzenie ciągu instrukcji, przerwanie etc, w ten sposób, że zostaje wykonana nie tylko sama instrukcja, lecz także czynność, do której użytkownik poprzez nią odwołuje się. Np. wykonanie instrukcji wywołania podprogramu oznacza tutaj nie tylko skok do początku tego podprogramu, lecz także jego wykonanie, powrót do programu wywołującego i ustawienie rejestrów CS i IP na pierwszą instrukcję po instrukcji CALL. Komenda PROCEED, bez podania parametrów, spowoduje wykonanie tylko tej instrukcji (wraz ze związanymi z nią czynnościami), którą aktualnie wskazują rejestry CS i IP. Parametry 'adres' i 'liczba bajtów hexadecymalnie' mają następujące znaczenie:

- parametr 'adres', podawany koniecznie wraz z poprzedzającym go znakiem "=", oznacza wykonanie tej instrukcji, której pierwszy bajt znajduje się w komórce o adresie wyznaczonym na podstawie tego parametru,
- parametr 'liczba instrukcji hexadecymalnie' oznacza ilość instrukcji (wraz ze związanymi z nimi czynnościami), którą ma wykonać debugger (nie mylić z ilością instrukcji w wywoływanym programie, wykonywanej pętli etc.).

Przykład:

Niech dany będzie następujący program:

```
04BA:0100  MOV AX,BX
04BA:0102  CALL 1000H
04BA:0105  JC 2000H
.          .
.          .
.          .
04BA:1000  XOR AX,AX
.          .
04BA:1XXX  RET
```

Niech rejestry CS i IP wskazują na instrukcję CALL 1000H (CS=04BA, IP=0102). Wykonanie komendy:

P = 0100 2

spowoduje wykonanie instrukcji MOV AX,BX, wywołanie podprogramu rozpoczynającego się od adresu 04BA:1000, powrót do programu wywołującego i ustawienie rejestrów CS i IP tak, że będą wskazywały instrukcję: JC 2000H (tj. CS=04BA, IP=0105).

10.16. Komenda Q (QUIT).

Postać komendy:

Q

Opis:

Komenda QUIT kończy sesję pracy z programem DEBUG i przekazuje sterowanie do systemu operacyjnego PC DOS, przy czym zbiór, który znajdował się w pamięci operacyjnej nie zostanie przepisany do pamięci masowej (dla jego zachowania należy przedtem użyć komendy WRITE).

10.17. Komenda R (REGISTER)

Postać komendy:

R [nazwa rejestru]

Opis:

Komenda R pozwala na wyświetlenie i zmianę zawartości poszczególnych rejestrów i wskaźników (flag). Możliwe są dwa następujące sposoby wykorzystania tej komendy:

- poprzez wywołanie jej bez podania parametru 'nazwa rejestru':

R

Debugger wyświetli wówczas na ekranie monitora zawartość poszczególnych rejestrów, stan wszystkich wskaźników zgodnie z podaną poniżej tabelą oraz mnemonikę instrukcji, wskazywanej przez rejestry CS i IP. Sposób wyświetlania zależy od ustawionego modu pracy ekranu monitora (w 80 lub 40 kolumnach). Opcja wywołania komendy REGISTER, bez podania parametrów, nie pozwala na żadne zmiany zawartości rejestrów i stanów poszczególnych wskaźników.

- poprzez wywołanie jej z podaniem parametru 'nazwa rejestru':

R (nazwa rejestru)

Debugger wyświetli wówczas na ekranie monitora zawartość żadanego rejestru i w nowej linii, po dwukropku zgłosi się oczekując na wpisanie nowej wartości. Jeśli zawartość rejestru ma pozostać niezmieniona, należy wcisnąć klawisz 'Enter'. Zmiana zawartości rejestru polega na podaniu jedno, dwu, trzy lub czterocyfrowej liczby hexadecimalnej i potwierdzenie tego wciśnięciem klawisza 'Enter':

R AX

AX F1E4

:nowa wartość 'Enter'

Stan poszczególnych wskaźników (flag) określa dwuliterowy kod stanu każdego wskaźnika, zgodny z poniższą tabelą:

Nazwa wskaźnika (flagi)	Wskaźnik zapalony	Wskaźnik zgaszony
Nadmiar (tak/nie) (ang. overflow)	OV	NV
Kierunek (maleńie/wzrost) (ang. direction)	DN	UP
Przerwania (możliwe/zablokowane) (ang. interrupt)	EI	DI
Znak (ujemny/dodatni) (ang. sign)	NG	PL
Zero (tak/nie) (ang. zero)	ZR	NZ
Pomocnicze przeniesienie (tak/nie) (ang. auxiliary carry)	AC	NA
Parzystość (parzystość/nieparzystość) (ang. parity)	PE	PO
Przeniesienie (tak/nie) (ang. carry)	CY	NC

Ustawianie stanu poszczególnych wskaźników, podobne do zmiany zawartości rejestru, polega na:

- wywołaniu komendy R z jednoliterowym oznaczeniem F (rejestr wskaźników):

R F

Debugger wyświetla aktualny stan wskaźników i oczekuje na odpowiedź użytkownika, np.

OV DN EI NG ZR AC PE CY -

- jeśli stan wskaźników ma pozostać niezmienny, to wystarczy wcisnąć klawisz 'Enter'. Zmianę stanu wskaźników można wykonać podając w dowolnym porządku, tylko dla wskaźników które mają ulec zmianie, taki stan, jaki ma zostać ustawiony (można nie używać separatorów). Np. W powyższym przykładzie zmianę stanu wskaźnika 1 (czyli

nadmiaru), 3 (czyli przerwania), 5 (czyli zera) i 7 (czyli parzystości) można uzyskać w następujący sposób:

OV DN EI NG ZR AC PE CY - PONZDINV

10.18. Komenda S (SEARCH).

Postać komendy:

S adres1 adres2 lista

lub:

S adres1 L (liczba bajtów hexadecymalnie) lista

Opis:

Komenda SEARCH pozwala na poszukiwanie wartości określonych parametrem 'lista' w żądanym obszarze pamięci operacyjnej. Parametr 'lista' może być pojedynczym bajtem, ciągiem znaków ASCII ujętych w apostrofy, ciągiem kilku bajtów etc. Zakres poszukiwań wyznaczają parametry 'adres1', 'adres2' lub 'adres1' i 'liczba bajtów hexadecymalnie'. Ich znaczenie jest następujące:

- parametr 'adres1' określa adres komórki pamięci, od której będą prowadzone poszukiwania,
- parametr 'adres2' określa adres komórki pamięci, do której będą prowadzone poszukiwania,
- parametr 'liczba bajtów hexadecymalnie', podawany zawsze po literze L, określa kolejny numer komórki pamięci, od komórki o adresie 'adres1', do której będą prowadzone poszukiwania.

Parametr 'adres1' może być tylko offsetem. Wówczas, dla określenia fizycznego adresu, zostanie wykorzystana wartość rejestru segmentu danych DS. Adres fizyczny końca zakresu jest zawsze wyznaczany na podstawie tego samego rejestru segmentowego (lub tej samej wartości podawanej przed dwukropkiem) co adres początku zakresu, stąd zamiast podawania całego adresu wystarczy podać tylko offset.

Po odnalezieniu komórki pamięci zawierającej żądaną wartość (lub komórek pamięci zawierających żądany ciąg wartości) debugger wyświetla jej adres (może być kilka takich miejsc w pamięci). Jeśli w wyspecyfikowanym obszarze pamięci operacyjnej brak jest takich komórek, wówczas program DEBUG wyświetli na ekranie monitora zgłoszenie oczekiwania na wprowadzenie następnej komendy.

Przykłady:

- S CS:100 110 41 - poszukiwana będzie hexadecymalna wartość 41H wewnątrz obszaru pamięci ograniczonego adresami: CS:100 i CS:110.
- S CS:100 L 11 41 'AB' E - poszukiwany będzie ciąg wartości hexadecymalnych: 41, 41, 42, 0E w przedziale pamięci o szerokości 11 bajtów, od adresu CS:100.

10.19. Komenda T (TRACE).

Postać komendy:

T [=adres][liczba instrukcji hexadecymalnie]

Opis:

Komenda TRACE pozwala na wykonywanie pojedynczych instrukcji w języku maszyny (assembler ASMB6), wyświetlając po każdej wykonanej instrukcji zawartość poszczególnych rejestrów, stany wszystkich znaczników (flag) oraz mnemonikę następnej do wykonania instrukcji. Wszystkie parametry komendy TRACE są opcjonalne i mają następujące znaczenie:

- parametr 'adres', zawsze poprzedzony znakiem '=', określa adres komórki pamięci zawierającej pierwszy bajt pierwszej do wykonania instrukcji. Jeśli nie podano tego parametru, wówczas debugger rozpocznie wykonywanie komendy TRACE od instrukcji wskazywanej przez rejestry CS i IP,
- parametr 'liczba instrukcji hexadecymalnie' określa ilość kolejnych instrukcji do wykonania. Jeśli nie podano tego parametru, to wykonana będzie tylko jedna instrukcja.

Wykonując, przy pomocy komendy TRACE, większą liczbę instrukcji można zawiesić wyświetlanie komunikatów na ekranie monitora kombinacją klawiszy 'Ctrl/NumLock'. Wznowienie wyświetlania nastąpi po wciśnięciu klawisza dowolnej litery.

U W A G A :

1. Komenda TRACE blokuje wszelkie przerwania hardware na czas wykonywania każdej instrukcji.
2. Komendy TRACE nie powinno używać się do zmiany maski przerwania układu 8259 (porty 20 i 21).
3. Jeśli jedną z wykonywanych instrukcji jest INT3, to dalsze wykonywanie programu będzie zawieszane.

10.20. Komenda U (UNASSEMBLE).

Postać komendy:

U [adres1]

lub:

U adres1 adres2

lub:

U adres1 L (liczba bajtów hexadecymalnie)

Opis:

Komenda UNASSEMBLE pozwala na zamianę instrukcji w hexadecymalnym kodzie maszynowym na ich postać mnemoniczną, wyświetlając adresy komórek, zawierających pierwsze bajty każdej instrukcji, ich kody hexadecymalne i postać mnemoniczną, np.

```
04BA:0100 206472 AND [SI+72],AH
04BA:0103 FC      CLD
04BA:0104 7665     JBE 016B
```

Komendę UNASSEMBLE można używać w następujący sposób:

1. Bez podawania żadnego parametru lub z podaniem tylko parametru 'adres1':

U

lub:

U adres1

W pierwszym przypadku deasemblacja rozpocznie się od adresu, na którym debugger zakończył wykonywanie ostatniej komendy UNASSEMBLE. Jeśli jest to pierwsza w tej sesji pracy komenda UNASSEMBLE, to operacja rozpocznie się od adresu CS:0100H. W drugim przypadku deasemblacja będzie wykonywana od adresu określonego parametrem 'adres1'. Parametr ten może być tylko offsetem (dla wyznaczenia adresu fizycznego wykorzystywana jest wówczas zawartość rejestru segmentu programu CS), ale bezwzględnie musi wskazywać komórke pamięci zawierająca pierwszy bajt instrukcji. W obu przypadkach zakres wykonywanej deasemblacji zależy od ustawionego modu pracy monitora:

- jeśli pracuje on w trybie 40-kolumnowym, to deasemblacja obejmie 16 kolejnych bajtów,
- jeśli pracuje on w trybie 80-kolumnowym, to deasemblacja obejmie 32 kolejne bajty.

2. Z podaniem parametrów określających długość zakresu deasemblacji:

U adres1 adres2

lub:

U adres1 L (liczba bajtów hexadecymalnie)

gdzie:

- parametr 'adres1' ma to samo znaczenie co w punkcie 1,
- parametr 'adres2' określa adres komórki pamięci na której ma zostać zakończona deasemblacja,

- parametr 'liczba bajtów hexadecymalnie', podawany zawsze po literze L, określa kolejny numer komórki pamięci od komórki o adresie 'adres1' do której będą prowadzone poszukiwania.

Adres fizyczny końca zakresu deasemblacji jest zawsze wyznaczany na podstawie tego samego rejestru segmentowego (lub tej samej wartości podawanej przed dwukropkiem), co adres początku zakresu, stąd zamiast podawania całego adresu wystarczy podać tylko offset. Ponieważ koniec zakresu deasemblacji może określać komórkę pamięci, która nie zawiera ostatniego bajtu instrukcji, to debugger automatycznie przesunie zakres do tego bajtu. Np. działanie trzech następujących komend: U 04BA:100 L 7, U 04BA:100 L 8, U 04BA:100 L 9 w poniższym przypadku jest równoważne:

```
04BA:0100 206472 AND [SI+72],AH
04BA:0103 FC      CLD
04BA:0104 7665     JBE 016B
04BA:0106 207370 AND [BP+DI+70],DH
```

10.21. Komenda W (WRITE).

Postać komendy:

W [adres [(oznaczenie dysku) sektor (liczba sektorów)]

Opis:

Komenda WRITE pozwala na zapisywanie na dysku danych z pamięci operacyjnej. Maksymalnie pojedynczą instrukcją WRITE można wpisać dane do 128 (80H) sektorów dyskowych, o pojemności 512 bajtów każdy. Jeśli podczas przepisywania wystąpią błędy zapisu na dysk (program DEBUG wyświetla wówczas odpowiedni komunikat), to operację można powtórzyć poprzez kolejne wciśnięcie klawiszy: 'F3' i 'Enter'. Komendę WRITE można używać w następujący sposób:

1. Podając wszystkie opcjonalne parametry:

W adres (oznaczenie dysku) sektor (liczba sektorów)

gdzie:

- parametr 'adres' określa adres komórki pamięci, od której rozpocznie się przepisywanie. Jeśli 'adres' określa tylko offset, to debugger, dla określenia adresu fizycznego, wykorzysta zawartość rejestru segmentowego CS,
- parametr 'oznaczenie dysku' określa stację dyskową, na której dane będą zapisywane (0 oznacza stację A:, 1 - stacje B: itd.),
- parametr 'sektor' określa hexadecymalny numer sektora dyskowego, od którego dane będą zapisywane,
- parametr 'liczba sektorów' określa hexadecymalną liczbę sektorów, do których dane będą przepisywane z pamięci operacyjnej (max. 128 sektorów).

Należy pamiętać, że w powyższym przypadku posługiwania się komendą WRITE, dane dotychczas zapisane w podanych sektorach ulegają skasowaniu.

2. Podając tylko parametr 'adres', lub nie podając żadnego parametru:

W adres

lub:

W

W tym przypadku debugger przepisze z pamięci operacyjnej od adresu CS:100 (lub od adresu określonego parametrem 'adres') zawartość całego zbioru, którego położenie określa zbiór kontroli bloku (File Control Block - FCB, znajdujący się w pamięci operacyjnej od adresu CS:5C, a formatowany: albo podczas wywoływania programu DEBUG, albo przez wykonanie odpowiedniej komendy NAME). Dodatkowo do rejestrów BX i CX powinna być wpisana liczba przepisywanych bajtów.

Zbiór zostaje przepisywany na dysk zachowując nazwę, z którą wywołano program DEBUG, lub tę którą podano w komendzie NAME. Zbiory o rozszerzeniu .EXE i .HEX muszą być zapisywane na dysku w specjalnym formacie, którego nie tworzy debugger. W związku z tym, jeśli ma być modyfikowana przy pomocy debuggera zawartość zbioru o rozszerzeniu .EXE lub .HEX, to sposób postępowania jest następujący:

- zmiana rozszerzenia zbioru przed załadowaniem go do pamięci operacyjnej (rozkaz RENAME - patrz rozkazy systemu operacyjnego),
- wywołanie debuggera, odpowiednia modyfikacja zbioru, ale bez korzystania z komend GO, PROCEED lub TRACE,
- przepisanie zbioru na dysk i powtórna zmiana jego rozszerzenia (na .EXE lub .HEX).

10.22. Zestawienie komend programu DEBUG.

Komenda A (ASSEMBLE)	- przekształcanie instrukcji assemblera ASM86 z postaci mnemonicznej na hexadecymalny kod maszynowy,
Komenda C (COMPARE)	- porównywanie zawartości dwóch bloków pamięci operacyjnej,
Komenda D (DUMP)	- wyświetlanie zawartości bloku pamięci operacyjnej,
Komenda E (ENTER)	- wyświetlanie i zmiana zawartości komórek pamięci RAM,
Komenda F (FILL)	- wpisywanie wartości do komórek pamięci RAM,
Komenda H (HEXARITHMETIC)	- dodawanie i odejmowanie dwóch liczb hexadecymalnych,
Komenda G (GO)	- wykonywanie programu do wskazanej instrukcji,
Komenda I (INPUT)	- odczytywanie danych z określonego portu,
Komenda L (LOAD)	- ładowanie danych z pamięci masowej do pamięci operacyjnej,

Komenda M (MOVE)	- przepisywanie zawartości określonego bloku pamięci operacyjnej w inne miejsce,
Komenda N (NAME)	- definiowanie nazwy zbioru,
Komenda O (OUTPUT)	- wysyłanie danych do określonego portu,
Komenda P (PROCEED)	- wykonywanie pojedynczych instrukcji z zatrzymaniem tego wykonywania na następnej w kolejności instrukcji,
Komenda Q (QUIT)	- zakończenie pracy z programem DEBUG,
Komenda R (REGISTER)	- wyświetlanie i zmiana zawartości rejestrów,
Komenda S (SEARCH)	- poszukiwanie w pamięci operacyjnej określonych wartości,
Komenda T (TRACE)	- wykonywanie pojedynczych instrukcji krok po kroku z wyświetlaniem zawartości rejestrów,
Komenda U (UNASSEMBLE)	- zamiana instrukcji z hexadecymalnego kodu maszynowego na postać mnemoniczną,
Komenda W (WRITE)	- przepisywanie danych z pamięci operacyjnej do pamięci masowej.

11. LITERATURA.

Bliższe dane dotyczące budowy i oprogramowania mikrokomputerów z rodziny IBM PC można znaleźć w następujących pozycjach:

"Disk Operating System Version 3.00" - copyright IBM Corporation, 1984.

"Microsoft MS-DOS version 3.10 Operating System" - copyright Microsoft Corporation, 1984.

"Disk Operating System Version 3.20 References" - copyright IBM Corporation, 1986.

"Disk Operating System User's Guide and Technical Reference" - copyright IBM Corporation, 1984 (i późniejsze).

"Macro Assembler Version 2.00" - copyright IBM Corporation, 1984.

"Macro Assembler Version 2.00 References" - copyright IBM Corporation, 1984.

Peter Norton: "Inside The IBM PC - Access To Advanced Features And Programing" - copyright Robert J. Brady Co, Bowie Maryland 20715.

Peter Morse: "The 8086/8088 Primer" - Hayden, 1980.

Robert Stards: "8087 Applications And Programing For The IBM PC" - copyright Robert J. Brady Co.

P. Misiurewicz: "Układy mikroprocesorowe i struktura oprogramowania" - WNT, 1984.