

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP

Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81

OAH Osrodek Automatyki Elektrycznej

Pracownia Oprogramowania Wieloprocesorowych Systemów
Automatyki i Robotów

Główny wykonawca mgr inż. Andrzej Aderek

Wykonawcy mgr inż. Krzysztof Czarnomski, mgr Mirosław Dembek
mgr inż. Magdalena Porada, mgr inż. Teresa Zinkiewicz

Konsultant

Nr zlecenia

UR 01.03.01 L

Wykonanie asemblera skrótnego dla
mikroprocesora Intel 8086 na mini-
komputer SM-4.

Etap 4. Opracowanie programów
assemblera w wersji docelowej.

Zleciennodawca *problem zgodny*

Kierownik zespołu

Pracę rozpoczęto dnia 1984-03-30

A. Syrczyński
dr inż. A. Syrczyński
zakończono dnia 85-12-15

Kierownik Pracowni Z-ca Dyrektora
d/s Automatyki

Kierownik Osrodka

Aderek
mgr inż. A. Aderek

T. Miszala
prof. dr inż. T. Miszala

Gałązka
dr inż. T. Gałązka

Praca zawiera:

Rozdzielnik - ilość egz:

stron 10

Egz. 1 BOINTE

rysunków

Egz. 2 OAE-83

fotografii

Egz. 3

tabel

Egz. 4

tablic

Egz. 5

załączników 1

Egz. 6

Nr rejestr. 5562

Analiza deskryptorowa

Oprogramowanie + assembler + mikroprocesor

Analiza dokumentacyjna

Praca zawiera opisy funkcjonalne i instrukcje operatorskie programów assemblera skróconego i skróconego konsolidatora relokującego mikroprocesora Intel 8086 na minikomputer SM-4.

Tytuły poprzednich sprawozdań

681.3:06 Oprogramowanie

681.32:621.347-181.48 Mikroprocesor

UKD

WAP-252/02-6000

SPIS TRESCI.

1. Wstep.	1
2. Opis funkcjonalny makroassemblera ASMB6/SM.	2
3. Opis funkcjonalny konsolidatora relokujacego LOC86/SM.	3
4. Instrukcja operatorska makroassemblera ASMB6/SM.	4
5. Instrukcja operatorska konsolidatora relokujacego LOC86/SM.	5
6. Literatura.	6
7. Zalacznik - tabulogramy programow ASMB6/SM i LOC86/SM (tylko w eszemplarzu autorskim).	

1. WSTEP.

Praca zawiera krotkie opisy oraz instrukcje operatorskie programow makroassemblera i konsolidatora skrosnego dla mikroprocesora Intel 8086, dzialajacych na minikomputerze SM-4 pod kontrola systemu operacyjnego RSX-11M. Programy te odpowiadaja elementom systemu uruchomieniowego MCS-86 firmy Intel, ktorych opisy wymienione sa w rozdziale 6. Programy zostaly napisane w jezyku pascal, co umozliwia przeniesienie ich na inne komputery posiadajace kompilator tego jezyka.

Zamierzeniem autorow byla mozliwie pelna zszodnosc definicji assemblera z opisem [1]. Zszodnie z zalozeniami (etap 2 niniejszej pracy) niektore elementy assemblera zostana wykonane w razie podpisania umowy wdrozeniowej z producentem komputera. Z pewnych mniej uzytecznych a pracochlonych w realizacji funkcji wykonywanych przez oryginalne programy mozna zrezygnowac, bez szkody dla efektywnosci programowania. Naleza do nich: definiowanie wlasnych codemacro'sow, stale rzeczywiste o dlugosciach wiekszych od 4 bajtow, 17-bitowa arytmetyka fazy translacji, relokacja typu HIGH i LOW, niektore informacje zawarte na listinsu.

Obecna wersja konsolidatora skrosnego zapewnia obsluge relokacji typu base, tzn. rozmieszczanie zawartosci segmentow (bez przedresowywania) od adresow fizycznych bedacych wielokrotnoscia liczby 16. Wskazane byloby uzupelnienie konsolidatora o obsluge relokacji typu offset, tzn. przedresowywanie zawartosci segmentow w razie umieszczenia jej w dowolnym miejscu pamieci (assembler obsluguje oba rodzaje relokowalnosci).

H

2. OPIS FUNKCJONALNY MAKROASEMBLERA ASMB6/SM.

Makroassembler skrosny ASMB6/SM dla mikroprocesora Intel 8086 został zrealizowany na wzor makroassemblera ASM-86, bedacego elementem systemu uruchomieniowego MCS-86 firmy INTEL. Pełny opis tego języka znajduje się w [1]. Poniżej podane są najważniejsze odchylenia od tej definicji:

- nie można definiować własnych codemacros'ów,
- argumenty operatorów HIGH i LOW nie mogą być relokowalne,
- nie zrealizowano dyrektywy GROUP,
- w wyrażeniach DUP występujących w strukturach nie można umieszczać symboli relokowalnych, nie można również zagnieżdżać takich DUP w sobie,
- nie zrealizowano dyrektyw DQ i DT w zakresie dużych liczb rzeczywistych,
- arytmetyka w wyrażeniach fazy asemblacji jest 16-bitowa, zamiast oryginalnej 17-bitowej.

Listings jest zgodny z opisem zawartym w [4], Chapter 4. Nie jest wypełniane pole object na listingsu dla linii, które nie tworzą objectu. Inny jest tekst zawarty w nagłówku każdej strony.

Tablica symboli zgodna jest z opisem zawartym w [4], Chapter 4, z wyjątkiem braku listy odwołań (XREFS) oraz braku niektórych informacji dla symboli, które są segmentami oraz tych, które były zdefiniowane dyrektywą EQU.

Numery i komentarze do błędów zgodne są z opisem zawartym w [4], Appendix A. Wykrywane są błędy o numerach 1 do 94, 107 do 125, 301 do 320. W konkretnych przypadkach assembler może reasować na błędy nieco inaczej, niż program oryginalny.

3. OPIS FUNKCJONALNY KONSOLIDATORA RELOKUJACEGO LOC86/SM.

Skrośny konsolidator relokujący LOC86/SM dla mikroprocesora Intel 8086 został zrealizowany na wzór programów konsolidatora LINK86 i programu relokującego LOC86, będących elementem systemu uruchomieniowego MCS-86 firmy INTEL. Zasady łączenia segmentów przez LINK86 i relokowania ich przez LOC86 znajdują się w [3]. W stosunku do nich znacznie zmieniony został sposób relokacji. Pewne zmiany dotyczą również łączenia segmentów.

3.1. RELOKACJA SEGMENTÓW.

1. Segmenty wyspecyfikowane w kluczu 'order' są lokowane w przestrzeni adresowej mikroprocesora 8086 w kolejności określonej w kluczu.
2. Klasy wyspecyfikowane w kluczu 'order' są lokowane w przestrzeni adresowej mikroprocesora 8086 w kolejności określonej w kluczu.
3. Pozostałe segmenty są lokowane w kolejności ich wystąpienia na wejściu programu za segmentami lokowanymi na podstawie klucza 'order'.
4. Segmenty są grupowane w klasy, tzn. kolejny segment o danej klasie jest lokowany za poprzednim segmentem o tej samej klasie. Nie dotyczy to segmentów wyspecyfikowanych w kluczu 'order'.
5. Dla segmentów i klas wyspecyfikowanych w kluczu 'order' operator może podać adresy początkowe przy pomocy klucza 'addresses'. W kluczu 'addresses' nie można specyfikować segmentów lub klas nie wyspecyfikowanych w kluczu 'order'. Również segmenty absolutne występujące w modułach wejściowych muszą być wyspecyfikowane w kluczu 'order'.
6. Program relokuje segmenty poczynając od adresu 0.
7. Segmenty o 'align type': byte, word, paragraph są lokowane od początku paragrafu (tzn. adresu, którego 4 najmłodsze bity są równe 0).
8. Segmenty o 'align type' page są lokowane od początku strony (tzn. od adresu, którego młodszy bajt = 0).
9. Po dojściu do końca przestrzeni adresowej mikroprocesora 8086 (0FFFFFFH), a nie ulokowaniu wszystkich segmentów podanych w modułach wejściowych konsolidator relokujący sygnalizuje błąd i kończy pracę.
10. Pierwszy napotkany na wejściu segment o 'combine type' memory jest lokowany na końcu, za wszystkimi innymi

segmentami, niezależnie od jego klasy. Sposób lokowania tego segmentu może być zmieniony przy pomocy klucza 'order'. Następne napotkane segmenty memory są traktowane tak jak segmenty o 'combine type' common.

3.2. LACZENIE (LINKING) SEGMENTOW.

Skrosny konsolidator relokujący łączy segmenty o tych samych nazwach i tych samych klasach tak, jak to jest przedstawione w [3], z tą różnicą, że nie łączy segmentów typu 'public', tzn. tych, które są łączone tak, że początek następnego jest lokowany za końcem poprzedniego. Segmenty o 'combine type': common, stack, memory są łączone w sposób podany w [3].

3.3. WEJSCIE I WYJSCIE KONSOLIDATORA

Skrosny konsolidator relokujący przetwarza dane z plików wejściowych zawierających moduły skompilowane przez asembler ASMB6/SM. Plik może zawierać jeden moduł.

Konsolidator produkuje plik wyjściowy i, opcjonalnie, plik mapy. Plik wyjściowy zawiera kod absolutny uzyskany po lokacji segmentów wejściowych w postaci "hexadecimal object file format" z bitem parzystości. Plik mapy zawiera mapę pamięci modułu wyjściowego oraz listę symboli. Postać mapy jest zgodna z opisem zawartym w [3].

4. INSTRUKCJA OPERATORSKA MAKROASEMLERA ASMB6/SM.

Asembler ASMB6/SM dostarczany jest w postaci dwóch plików o nazwach ASMB6.TSK i OASMB6.TSK. Pierwsze zadanie analizuje linie wywołania asemblera oraz realizuje funkcje makro. Drugie wykonuje pozostałe funkcje asemblera. Informacje do drugiego zadania są przekazywane przez pliki robocze ASMB6MAC.TMP i ASMB6LST.TMP, które po zakończeniu pracy asemblera są automatycznie kasowane.

Użytkownik wywołuje zadanie ASMB6, które zsyła się pisać na monitorze "ASMB6)". Po zakończeniu przebiegu makro zadanie OASMB6 zostanie wykonane automatycznie pod warunkiem, że:

1. Znajduje się na tym samym dysku i koncie, z którego zostało wywołane.
2. Zostało uprzednio zainstalowane pod nazwą OASMB6.

Można również wykonać zadanie OASMB6 bez spełniania tych warunków, wywołując je dyrektywą RUN umieszczoną np. w pliku rozkazowym razem z dyrektywą uruchamiającą ASMB6. Nie ma sensu wykonywanie jednego z tych zadań bez wykonania drugiego. Zadanie OASMB6 tworzy plik roboczy ASMB6WOR.TMP, kasowany automatycznie po zakończeniu przebiegu.

Postać linii rozkazu dla asemblera, po zgłoszeniu gotowości napisem "ASMB6)" jest zgodna ze standardem systemu RSX-11M. Linia składa się z części definiującej pliki wyjściowe (object i listings) i specyfikacji pliku wejściowego (źródłowego), rozdzielonych znakiem "=", oraz z kluczy, według schematu:

```
ASMB6)object,listings=source/klucz1/klucz2...
```

W linii wywołania mogą występować zarówno małe jak i duże litery. Nie są one rozróżniane przez asembler.

Obligatoryjne są:

- znak "=",
- nazwa pliku source,
- znak ",", jeśli występuje plik listings,
- poprzedzający znak "/" jeśli występuje klucz.

Brak specyfikacji plików object i listings spowoduje, że pliki te nie będą utworzone. Brak specyfikacji pliku listings spowoduje wyprowadzenie informacji o błędach na monitor, bez względu na wartość kluczy.

Specyfikacja plików jest standardowa. Zdefiniowane są domyślne typy plików:

- .OBJ dla obiektu,
- .LST dla listingu
- .ASM dla pliku źródłowego.

Klucze mogą wystąpić po specyfikacji dowolnego pliku (niekoniecznie źródłowego). Można użyć następujących kluczy ("*" oznacza wartości domyślne):

- /se - listing będzie zawierał makrodefinicje, makrowywołania i makrorozwinięcia;
- * /so - listing będzie zawierał tylko makrorozwinięcia;
- /nose - listing będzie zawierał makrodefinicje i makrowywołania, bez makrorozwinięć;
- * /li - będzie produkowany pełny listing, chyba że plik listing nie jest wyspecyfikowany;
- /noli - w pliku listingsowym będą umieszczone tylko informacje o błędach (plik listing musi być wyspecyfikowany);
- /ln - na monitor będzie wprowadzona informacja o tym, który przebieg jest wykonywany oraz będzie wyświetlany numer linii programu źródłowego, analizowanej aktualnie przez asembler. Użycie tego klucza spowoduje pewne spowolnienie pracy asemblera;
- * /noln - nie będą wprowadzane informacje opisane dla klucza /ln;
- * /mr - normalny przebieg makro;
- /nomr - przebieg makro nie analizuje treści programu źródłowego, ogranicza się do utworzenia plików roboczych dla następnych przebiegów;
- /sb - na zakończenie pracy asemblera na plik listingsowy zostanie wprowadzona tablica symboli (plik listing musi być wyspecyfikowany);
- * /nosb - nie będzie produkowana tablica symboli;

/tt - na monitor będzie wyprowadzona informacja o tym, który przebieg jest wykonywany oraz w razie wystąpienia błędu zostanie wyświetlona błędna linia i komunikat o błędzie, oddzielnie w każdym przebiegu.

* /nott - nie będą wyprowadzane informacje opisane dla klucza /tt.

W razie stwierdzenia błędów linii rozkazu lub niemożności otwarcia plików będzie wyprowadzony odpowiedni komunikat. W trakcie pisania linii rozkazu można w każdej chwili zakończyć pracę asemblera wprowadzając CTRL-W i CR.

Na zakończenie pracy asembler wyprowadza na monitor komunikat:

ASSEMBLY COMPLETE, NO ERRORS FOUND
lub ASSEMBLY COMPLETE, nn ERRORS FOUND .

5. INSTRUKCJA OPERATORSKA KONSOLIDATORA RELOKUJACEGO LOC86/SM

Skrosny konsolidator relokujacy LOC86/SM dostarczany jest w postaci pliku o nazwie LOC86.TSK. Uzytkownik wywoluje zadanie LOC86, ktore zglasza sie piszac na monitorze "LOC86>". Oznacza to sctowosc i oczekiwanie zadania na linie wywolania.

Postac linii wywolania dla konsolidatora jest zgodna ze standardem systemu RSX-11M. Linia sklada sie z czesci definiujacej pliki wyjsciowe (object i map) i specyfikacji plikow wejsciowych oraz z kluczy wedlug schematu:

```
LOC86>object, map=source1, ..., /klucz1/klucz2...
```

Obligatoryjne sa:

- znak "=",
- i nazwa pliku source,
- znak ",", jezeli wystepuje plik map,
- poprzedzajacy znak "/" Jesli wystepuje klucz.

W linii wywolania moga wystepowac zarowno male jak i duze litery. Nie sa one rozrozniane przez konsolidator.

Nie podanie nazwy pliku map spowoduje nie wyprodukowanie przez konsolidator pliku mapy. Nie podanie nazwy pliku object spowoduje nadanie mu nazwy pierwszego pliku source. Dla pliku o nieokreslonym przez operatora typie, konsolidator przyjmuje typ domyslny. Typy te sa nastepujace:

- dla pliku object - .T86,
- dla pliku map - .M86,
- dla plikow source- .O86.

Klucze moga wystapic po specyfikacji plikow source. Kazdy klucz musi byc poprzedzony znakiem "/" .Zdefiniowane sa dwa klucze: addresses i order.

11

KLUCZ ADDRESSES.

Postac klucza:

addresses(a1(x1(w1),...),...)

gdzie:

a = slowo kluczowe segments lub slowo kluczowe classes,

x = nazwa segmentu (klasy),

w = adres segmentu (klasy).

Slowa kluczowe segments i classes okreliaja, czy dane zawarte za nim w nawiasie odnosza sie do segmentow, czy do klas. Nazwa segmentu (klasy) jest jedna z nazw wystepujacych w modulach wejsciowych. Adres jest liczba z zakresu 0..10488575 w postaci dziesietnej, binarnej, oktalnej lub hexadecymalnej. Ta sama nazwa nie moze wystapic wiecej niz jeden raz.

Zdefiniowane sa nastepujace skroty slow kluczowych:

addresses = ad,

segments = sm,

classes = cs.

Klucz addresses okresla adresy absolutne, na jakich zostana ulokowane wymienione w kluczu segmenty i klasy.

W linii wywolania konsolidatora klucz addresses moze wystapic wielokrotnie. Jego dzialanie jest addytywne.

Wszystkie segmenty i klasy wymienione w kluczu addresses musza byc rowniez wymienione w kluczu order.

KLUCZ ORDER.

Postac klucza: .

order(a1(x1,...),...)

gdzie:

a = slowo kluczowe segments lub slowo kluczowe classes,

x = nazwa segmentu (klasy).

Słowa kluczowe segments i classes określają, czy dane zawarte za nim w nawiasie odnoszą się do segmentów, czy do klas. Nazwa segmentu (klasy) jest jedną z nazw występujących w modułach wejściowych. Ta sama nazwa nie może wystąpić więcej niż jeden raz.

Zdefiniowane są następujące skróty słów kluczowych:

order = od
segments = sm,
classes = cs.

Klucz order określa kolejność w jakiej mają być lokowane w przestrzeni adresowej mikroprocesora 8086 wymienione w kluczu segmenty i klasy.

Klucz order może wystąpić w linii wywołania jeden raz.

KOMENTARZE I LINIE KONTYNUACJI

Linia wywołania konsolidatora może być kontynuowana i może zawierać komentarze. Wprowadzony przez operatora znak "&" oznacza, że będzie linia kontynuacji. Po naciśnięciu CR konsolidator zgłasza się "**" i można wprowadzać linie kontynuacji. Linii tych może być do 20. Wszystkie znaki wprowadzone pomiędzy "&" a końcem linii traktowane są jako komentarz i nie są analizowane.

KOMUNIKATY O BŁĘDACH

Komunikaty o błędach są drukowane na monitorze operatora ("TI:") oraz na pliku mapy, jeżeli taki plik jest otwarty. Numery i komentarze do błędów zgodne są z opisem zawartym w [3], Appendix A. W konkretnych przypadkach konsolidator może reagować na błędy nieco inaczej niż oryginalne programy konsolidatora i lokatera.

6. LITERATURA.

- [1] "8086/8087/8088 Macro Assembly Language Reference Manual for 8080/8085 - Based Development Systems", Manual order no. 121623-001 rev. A
Copyright 1980, Intel Corporation.
- [2] "MCS-86 Absolute Object File Formats", An Intel technical specification, order no. 9800821 A
Copyright 1978, Intel Corporation.
- [3] "MCS-86 Software Development Utilities Operating Instructions for ISIS-II Users", Manual order no. 9800639 B
Copyright 1978, Intel Corporation.
- [4] "MCS-86 Macro Assembler Operating Instructions for ISIS-II Users" Manual order no. 9800641-02
Intel Corporation.