

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP
Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81

OSRODEK AUTOMATYZACJI KOMPLEKSOWEJ I SYSTEMOW CYFROWYCH

440 Pracownia Sprzetu Cyfrowego OAK - 31

BE 10

Główny wykonawca

Wykonawcy mgr K. Najar¹³⁶ ; mgr inż. R.D.Tyrcha¹¹⁵

Konsultant

Nr zlecenia 28/83

Adaptacja crossasemblera INTEL-8080
dla maszyn serii SM/PDP 11
do potrzeb aplikacji mikroprocesorowych
/Instrukcja eksploatacji wprowadzonych
zmian/

Zleceniodawca praca własna

Pracę rozpoczęto dnia 5.07.83
Kierownik Pracowni

mgr inż. S.Szwaglis

Z-ca Dyrektora
d/s Automatyki

dr inż. T.Gałązka

zakończono dnia 5.09.83
Kierownik Osrodka

mgr inż. J.Hawryluk

Praca zawiera:

stron 46

rysunków

fotografii

tabel

tablic

załączników

Rozdzielnik - ilość egz:

Egz. 1BOINTE

Egz. 2 OAK - 31

Egz. 3

Egz. 4

Egz. 5

Egz. 6

Nr rejestr. 5081

Analiza deskryptorowa

URZADZENIA AUTOMATYCZNEJ REGULACJI I STEROWANIA KSAP + MIR - PROWAY +
MIKROPROCESOR + OPROGRAMOWANIE

Analiza dokumentacyjna

Instrukcja użytkownika zmian oprogramowania wspomagającego
crossasemblera mikroprocesora INTEL - 8080 .

Praca zawiera instrukcję użytkownika wprowadzonych zmian
oraz listingi opracowanych programów /z komentarzem /.

Tytuły poprzednich sprawozdań

62-50 Teoria - podla ...

UKD

MERA-PIAP/TW 331/78 5000

ADAPTACJA CROSSASSEMBLERA MIKROKOMPUTERA
=====

INTEL 8080 DLA MASZYN SERII SM/PDP 11
=====

DLA POTRZEZ APLIKACJI MIKROPROCESOROWYCH.
=====

U W A G A
=====

- 1) Opracowanie ponizsze jest jedynie dokumentacja zmian wykonanych w celu ulatwienia wykorzystania posiadanego crossassemblera mikroprocesora INTEL 8080.
Dotychczas bowiem pisane i asemlowane programy na minikomputerze MERA 400 nie mogly byc wczytywane (z taśmy papierowej) do minikomputera SM 50/50 z uwagi na inny delimiter lini oraz inna strukture rekordu.
Ponadto pisane oprogramowanie standardowo wykorzystuje liczby hexadecymalne ktore w crossassemblerze na minikomputerze SM 50/50 jest niedopuszczalne (wymasany jest format liczb oktalnych). Wreszcie kod wynikowy generowany przez crossassembler nie odpowiada formatom taśmy standardowo czytanej przez programy mikroprocesora.
Powyzsze problemy zostaly rozwiazane przez ponizej dokumentowane programy.
- 2) Dla zapoznania sie z crossassemblerem i symulatorem mikroprocesora INTEL 8080 przeznaczona jest posiadana w OAK-31 dokumentacja:
 - a) Instytut Badan Jadrowych
Realizacja na maszynie PDP 11/45 systemu assemblacji dla mikroprocesora INTEL 8080
 - b) Instytut Badan Jadrowych
Program symulatora mikroprocesora INTEL 8080 zrealizowany na maszynie PDP 11/45.
- 3) Do poznania maszyny oraz oprogramowania SM 50/50 przeznaczone sa posiadane w OAK-31 odpowiednie tomy dokumentacji (ktorych liste zawiera strona 5).

1. WSTEP

Celem niniejszego opracowania jest przedstawienie zmian w systemie przygotowania programow dla mikrokomputerow INTEL 8080.

Oprogramowanie dziala w systemie RSX - 11M. Dzieki

wysodnym narzedziom programowym dostepnym w systemie uruchomienie programow INTELowskich jest rzecza duzo szybsza niz w oryginalnym systemie mikrokomputerowym.
Na mc. SM lub PDP dostepny jest m. in.

- wysodny program EDITOR pozwalajacy latwo manipulowac tekstem programu zrodlowego?
- wysodny i prosty sposob manipulacji i przechowywania zbiorow roznnych typow.

2. WYMAGANIA NA SYSTEM CYFROWY.

Dla poprawnej pracy oprogramowania potrzebna jest nastepujaca minimalna konfiguracja systemu cyfrowego:

- procesor SM50/50-1 lub SM - 4?
- dysk systemowy?
- system operacyjny RSX - 11M?
- terminal.

3. STRUKTURA SYSTEMU WSPOMAGANIA.

System wspomagajacy opracowywanie programow INTELowskich sklada sie z nastepujacych programow:

- PRR - program czytajacy tasme symboliczna MERY-400?
- CHX - program dokonujacy konwersji liczb heksadecymalnych na postac oktalna?
- CON - program dokonujacy konwersji pliku wynikowego asemblera INTELowskiego na postac akceptowana przez program ladujacy mikrokomputera.

Realizacja asemblacji języka symbolicznego mikrokomputera INTEL 8080 dokonywana jest za pomocą rodziny makrodefinicji - INTHAS - pozwalających za pomocą asemblera MACRO - 11 dokonywać translacji programów ze składnia asemblera mikrokomputera INTEL 8080.

Tekst źródłowy dla asemblera INTHAS czytany jest z pliku źródłowego umieszczonego na dowolnym nośniku systemu RSX lub z czytnika za pomocą programu PRR.

Program PRR czyta dowolną taśmę symboliczną i zapisuje ją w pliku, który może być plikiem wejściowym asemblera INTHAS, preprocesora liczb heksadecymalnych CHX lub innych programów systemowych (np. EDITOR).

Program CHX dokonuje konwersji argumentów heksadecymalnych używanych w programie źródłowym na postać oktalaną, która jest akceptowana przez MACRO-11.

Produkt asemblacji programu źródłowego zapisany na dysku przez asembler MACRO - 11 podlega przetworzeniu przez program CON, w wyniku czego produkowany jest na dysku zbiór w formacie akceptowanym przez program ładujący mikrokomputera. Zbiór ten można przesłać na dowolne urządzenie systemu (perforator, terminal, taśma magnetyczna, dyskietka itp.) za pomocą programu systemowego FLX.

Każdy z prezentowanych programów wywoływany jest (po zainstalowaniu w systemie) w sposób identyczny:

```
>CON nazwa-zb-wyjsciowego=nazwa-zb-wejsciowego  
>CHX nazwa-zb-wyjsciowego=nazwa-zb-wejsciowego  
>PRR nazwa-zb-wyjsciowego=PR:
```

Dla programu PRR zbiorem wejściowym niezależnie od podanej nazwy jest czytnik PR0:

Opis sposobu wywoływania programów systemowych znajduje się w bibliotece systemu.

Ponizej przedstawiona zostanie przykładowa sekwencja czynności operatorskich pozwalająca uzyskać z papierowej taśmy symbolicznej papierową taśmę wynikową w formacie INTEL.

1. Instalacja programów : CON, CHX, PRR, MAC, FLX
2. Założenie papierowej taśmy symbolicznej do czytnika.
3. Wywołanie programu PRR czytającego taśmę.

```
>PRR TEST.1XT=PR:
```

4. Konwersja argumentow heksadecymalnych.
>CHX TEST.MAC=TEST.1XT
5. Asemblacja.
>MAC TEST,LP:=INTMAC,TEST,ENDMAC
6. Konwersja na postac INTELowska.
>CON TEST.INT=TEST.OBJ
Przyrostki .OBJ i .INT moga byc w tym przypadku
pominiete.
7. Perforacja tasmv wynikowej.
>FLX PP:=TEST.INT/RS

Dla uproszczenia opisanej tu procedury utworzony zostal plik rozkazowy , ktory automatycznie wywoluje programy wspomazajace. Plik ten nosi nazwe INTEL.CMD). Wywolwany jest w sposob nastepujacy :

>GINTEL

Teskt pliku przedstawiony jest na listinsu. Wymaga on od uzytkownika podania nazwy zbioru wejsciowego, wyjsciowego oraz odpowiedzi czy czynnosc okreslona w pytaniu ma zostac wykonana czy nie.

Proponowany system wspomazajacy powinien przyspieszyc i znacznie ulatwic wykonywanie wszelkich prac zwiazanych z mikrokomputerem INTEL-8080.

STROVA 5
 No. 5081

RSX-11M/RSX-11S DOCUMENTATION DIRECTORY

Table 1
 RSX-11M/RSX-11S Documentation Set

Volume	Document Title	Order Number	
		DEC-11-	
Volume 1A System Reference	RSX-11M/RSX-11S Documentation Directory	OMUGA-C-D	20.
	RSX-11M/RSX-11S Release Notes	OMRNA-C-D	
	Introduction to RSX-11M	OMIEA-B-D	61.
	RSX-11M System Generation Manual	OMGIA-C-D	72.
	* RSX-11S System Generation and Installation Guide	OSGIA-B-D	
	RSX-11M Executive Reference Manual	OMERA-C-D	80.
Volume 1B System Reference	IAS/RSX-11 MACRO-11 Reference Manual	OIMRA-B-D	55
	RSX-11M Task Builder Reference Manual	OMTBA-C-D	93.
Volume 1C System Reference	RSX-11M I/O Drivers Reference Manual	OMDRA-C-D	
	IAS/RSX-11 I/O Operations Reference Manual	OIORA-B-D	
Volume 2A System Procedures	RSX-11 Utilities Procedures Manual	OXMDA-A-D	
	RSX-11M Operator's Procedures Manual	OMOGA-C-D	
	IAS/RSX-11 ODT Reference Manual	OIODA-A-D	11.
Volume 2B System Procedures	RSX-11M Guide to Writing an I/O Driver	OMWDA-C-D	13.
	RSX-11M User Mode Diagnostics Reference Manual	OMUMA-A-D	27.
	RSX-11M Error Logging Reference Manual	OMELA-B-D	
	IAS/RSX-11 Disk Save and Compress User's Guide	OIRUA-A-D	3.
	PDP-11 Preservation Utility (PRESRV) User's Guide	UPRMA-B-D	5.
	RSX-11M Pocket Reference Card	OMPRA-A-D	17.
Volume 3A BASIC Language	* BASIC-11 Language Reference Manual	LIBBB-A-D	
	* IAS/RSX BASIC User's Guide	LIBIA-B-D	
	* IAS/RSX BASIC Installation Guide	LIBJA-B-D	

* Available under separate license.

SM 57152-1 OPIS TECHNICIAN

4

01234567890123456789 ** RSX-11M V3.2 ** 5-SEP-83 08:56:26 DRK0:L111,1111INTEL.CHM135 01234567890123456789
 01234567890123456789 ** RSX-11M V3.2 ** 5-SEP-83 08:56:26 DRK0:L111,1111INTEL.CHM135 01234567890123456789
 01234567890123456789 ** RSX-11M V3.2 ** 5-SEP-83 08:56:26 DRK0:L111,1111INTEL.CHM135 01234567890123456789

IIIIII	NR	NR	TTTTTTTT	EEEEEEEE	LL	
IIIIII	NR	NR	TTTTTTTT	EEEEEEEE	LL	
II	NR	NR	TT	EE	LL	
II	NR	NR	TT	EE	LL	
II	NNNN	NR	TT	EE	LL	
II	NNNN	NR	TT	EE	LL	
II	NR	NR	TT	EEEEEEEE	LL	
II	NR	NR	TT	EEFELEE	LL	
II	NR	NNNN	TT	EE	LL	
II	NR	NNNN	TT	EE	LL	
II	NR	NR	TT	EE	LL
II	NR	NR	TT	EE	LL
IIIIII	NR	NR	TT	EEEEEEEE	LLLLLLLL
IIIIII	NR	NR	TT	EEEEEEEE	LLLLLLLL

CCCCCCCC	MM	MM	DDDDDDDD	????	333333	55555555
CCCCCCCC	MM	MM	DDDDDDDD	????	333333	55555555
CC	NNNN	NNNN	DD	DD	???	33 33 55
CC	NNNN	NNNN	DD	DD	???	33 33 55
CC	MM	MM	DD	DD		33 555555
CC	MM	MM	DD	DD		33 555555
CC	MM	MM	DD	DD	????	33 55
CC	MM	MM	DD	DD	????	33 55
CC	MM	MM	DD	DD	????	33 33 55
CC	MM	MM	DD	DD	????	33 33 55
CC	MM	MM	DD	DD	??	33 33 55 55
CC	MM	MM	DD	DD	??	33 33 55 55
CCCCCCCC	MM	MM	DDDDDDDD	??	333333	555555
CCCCCCCC	MM	MM	DDDDDDDD	??	333333	555555

01234567890123456789 ** RSX-11M V3.2 ** 5-SEP-83 08:56:26 DRK0:L111,1111INTEL.CHM135 01234567890123456789
 01234567890123456789 ** RSX-11M V3.2 ** 5-SEP-83 08:56:26 DRK0:L111,1111INTEL.CHM135 01234567890123456789
 01234567890123456789 ** RSX-11M V3.2 ** 5-SEP-83 08:56:26 DRK0:L111,1111INTEL.CHM135 01234567890123456789


```

.ASK INS CZY POTRZEBUJESZ OBJASNIEN
.IFF INS .GOTO 100
; ASSEMBLER INTELA 8080.
;
; UWAGA !!!
;
; Nazwy plikow nalezy podawac bez przyrostkow.
; Dla pliku wejsciowego przyjmowany jest przyrostek '.HAC'.
; Po konwersji liczb hexadecymalnych plik wyjsciowy zachowuje
; przyrostek '.HAC'.
; Plik wyjsciowy po asemblacji ma przyrostek '.OBJ'.
; Po zakonczeniu konwersji pliku na postac INTELowska
; otrzymujemy plik z przyrostkiem '.INT'.
;
;
; Translacja dokonywana jest w dwoch etapach.
;
;
;Etap 1.
; Wszystkie argumenty hexadecymalne sa sprawdzane i tlumaczone
; na postac oktalna. Argumenty bledne zastepowane sa swiazdkami.
;ETAP 2.
; Wlasciwa translacja tekstu programu.
;
.100:.ENABLE SUBSTITUTION
.ASKS PLIKO NAZWA PLIKU WEJSCIOWEGO (BEZ PRZYROSTKA!)
.ASKS PLIKI NAZWA PLIKU WYJSCIOWEGO (BEZ PRZYROSTKA!)
.ASK E11 WYKONAC KONWERSJE ARGUMENTOW HEX
.IFF E11 .GOTO 200
INS CONHEX
CHX 'PLIKI'.HAC='PLIKO'.HAC
.200:
.ASKS PLIKL LISTING NA L NL: - BEZ LISTINGU I
MAC 'PLIKI','PLIKL'=SY:1111,1111INTMAC,'PLIKO',EPDMAC
INS HOX
CON 'PLIKI'.INT='PLIKI'.OBJ
.ASK E13 PERFOROWAC TASME WYNIKOWA
.IFF E13 .GOTO 300
FLX PP:='PLIKI'.INT/RS
.300:
REM ...CON
REM ...CHX

```

STRONA 2
STRON 46
N. REJ 5081

PROGRAM COUNT (INPUT OUTPUT)

STRONA 9
STRON 46
N= 025 5084

PROG: PDP-11 VER: 1.01 1983-08-30 16:14:06 PAGE 1
COURT.PAS

```
1  (*)
2  (*****
3  (*)
4  (*) THIS PROGRAM READS EVERY ASCII PAPER TAPE (*)
5  (*) AND CONVERTS IT TO STANDARDS OF RSX (*)
6  (*) TEXT FILE. (*)
7  (*)
8  (*)
9  (*) DATE = 30.08.83 (*)
10 (*) AUTHOR = R. DONALD TYRCHA (*)
11 (*)
12 (*)
13 (*****
14 (*)
15 CONST
16     MAXNAM = 20; (* NAME LENGTH *)
17     MAXTBL = 15; (* TABLE SIZE *)
18 TYPE
19     TABL = ARRAY [0 .. MAXTBL] OF CHAR;
20     NAMTAB = ARRAY [0 .. MAXNAM] OF CHAR;
21 VAR
22     INNAM,OUNAM : NAMTAB;
23     INDEV,OUDEV,INACNT,OUACNT : TABL;
24     OUTP1 : TEXT;
25     CH :ASCII ;
26 (*#I+GETNAM*)
```

```

1  (* THIS COMMENT IS NEEDED BY COMPILER *)
2  PROCEDURE NAME (VAR INDEV, INACNT : TABL; VAR OUDEV, OUACNT : TABL;
3      VAR INNAM, OUNAM : NAMTAB);
4  (* *)
5  (*****
6  (* *)
7  (* *)
8  (* *)
9  (* GLOBAL DATA : *)
10 (* CONST MAXNAM = ? -- NAME LENGTH *)
11 (* NAXTEL = ? -- ARRAY SIZE *)
12 (* TYPE TABL = ARRAY [0 .. MAXTEL] OF CHAR; *)
13 (* NAMTAB = ARRAY [0 .. MAXNAM]
14 (* OF CHAR *)
15 (* VAR INNAM : NAMTAB -- INPUT FILE NAME *)
16 (* OUNAM : NAMTAB -- OUTPUT FILE NAME *)
17 (* INDEV : TABL -- INPUT DEVICE NAME *)
18 (* OUDEV : TABL -- OUTPUT DEVICE *)
19 (* INACNT : TABL -- INPUT ACCOUNT *)
20 (* OUACNT : TABL -- OUTPUT ACCOUNT *)
21 (* *)
22 (* *)
23 (*****
24 (* GET INPUT AND OUTPUT FILE SPECIFICATION. *)
25 (* *)
26 (* *)
27 (* VERSION 00 DATE : 29.08.83 *)
28 (* AUTHOR : R. DONALD TYRCHA *)
29 (* *)
30 (*****
31 (* *)
32 CONST
33     MAXLIN = 80; (* LINE BUFFER SIZE *)
34 TYPE
35     LINE = ARRAY ( 0 .. MAXLIN ) OF CHAR;
36 VAR
37     CMD : LINE;
38     INDEV, INACNT, OUDEV, OUACNT : INTEGER; (* LENGTHS *)
39     I, PTR, MAX : INTEGER;
40     RES : INTEGER;
41     INNMLE, OUNMLE : INTEGER; (* LENGTHS *)
42     OK : BOOLEAN;
43 (* *)
44 (*****
45 (* *)
46 PROCEDURE SKIP ( VAR PTR : INTEGER; MAX : INTEGER);
47 (* SKIP SPACES *)
48 BEGIN (* SKIP *)
49     WHILE (CMD [PTR] = ' ') AND (PTR <= MAX) DO PTR := PTR+1;
50 END; (* SKIP *)
51 (* *)
52 (*****
53 (* *)
54 PROCEDURE REANAM ( VAR DEV : TABL; VAR DEVL : INTEGER;
55     VAR ACNT : TABL; VAR ACNLE : INTEGER;
56     VAR NAM : NAMTAB; VAR NAME : INTEGER;

```

PASCAL POP 11 VERSION 6.01 1983-08-30 16:14:06 PAGE 3
 UETAH.PAS NAME

```

57         VAR PTR : INTEGER; MAX : INTEGER;
58         VAR OK : BOOLEAN );
59 (* READ NAME AND PUT ITS PARTS INTO APPROPRIATE ARRAYS *)
60 VAR
61     PTC,I,J : INTEGER;
62     AL : BOOLEAN;
63 BEGIN (* REANAM *) OK := TRUE; SKIP (PTR,MAX); PTC := PTR;
64     WHILE NOT (CMD LPTC) IN [' ',' ',' ',' ','L']) AND (PTC <= MAX) DO
65         PTC := PTC + 1;
66         IF CMD LPTC) = ' ' THEN
67             BEGIN J := 0;
68                 FOR I:=PTR TO PTC DO
69                     BEGIN DEV LJI := CMD LIT; J := J+1
70                     END; (* FOR *)
71                 DEVLE := J; PTR := PTC+1; SKIP (PTR,MAX); PTC := PTR
72             END; (* IF *)
73         WHILE NOT (CMD LPTC) IN ['L',']',' ',' ','L']) AND (PTC <= MAX) DO
74             PTC := PTC + 1;
75             IF CMD LPTC) = 'L' THEN
76                 BEGIN J := 0;
77                     WHILE ( CMD LPTC) (<) 'L') AND (PTR <= MAX) DO
78                         BEGIN ACNT LJI := CMD LPTC; PTR := PTR+1; J := J+1
79                         END; (* WHILE *)
80                     IF CMD LPTC) = 'L' THEN
81                         BEGIN ACNT LJI := CMD LPTC; PTR := PTR +1; J := J+1
82                         END (* IF *)
83                     ELSE
84                         BEGIN OK := FALSE; WRITELN(' -- '']' HISSING. ');
85                         END (* ELSE *)
86                     ACNTLE := J; SKIP (PTR,MAX)
87                 END (* IF *)
88             ELSE
89                 IF CMD LPTC) = ']' THEN
90                     BEGIN OK := FALSE; WRITELN(' -- ']' HISSING. ');
91                     END; (* IF *) J := 0;
92                 WHILE (CMD LPTC) (<) '='') AND (PTR <= MAX) AND (CMD LPTC) (<) ' ') DO
93                     BEGIN NAM LJI := CMD LPTC; J := J+1; PTR := PTR + 1
94                     END; (* WHILE *) NAMLE := J;
95                     SKIP (PTR,MAX);
96                     IF CMD LPTC) = '=' THEN PTR := PTR + 1
97                 END; (* REANAM *)
98         (*
99         (*****
100        (*
101        PROCEDURE GCHL (VAR S : LINE; VAR LEN : INTEGER);
102        EXTERN;
103        (* GET MCR COMMAND LINE *)
104        (*
105        (*****
106        (*
107        BEGIN (* NAME *)
108            FOR I := 0 TO MAXNAM DO
109                BEGIN OUNAM LID := ' '; INNAM LID := ' '
110                END; (* FOR *)
111            FOR I := 0 TO MAXTEL DO
112                BEGIN INDEV LID := ' '; OUDDEV LID := ' '; TRASH LID := ' ';
  
```

PASCAL PDF-11 VERSION 6.01
GETNAM.PAS

1983-08-30
NAME

16:14:06

PAGE 4

```
113     QUACNT [I] := ' '
114     END; (* FOR *)
115     GCHL (CMD,MAX); PTR := 5; (* CMD POINTER *)
116     REANAM (OUDEV,OUDELE,OUACN),OUACLE,OURAN,OURMLE,PTR,MAX,OK);
117     IF OK THEN
118         REANAM (INDEV,INDELE,INACN),INACLE,INRAN,INRMLE,PTR,MAX,OK)
119     END; (* NAME *)
```

STRONA 13
STRON 46
N=REJ 5084

Pracul. PDP 11 VERSION 6.01 1903 00 30 1641400. PAGE 1
CONTR.PAS COIII

```
27  
28 PROCEDURE RDASCT(VAR CH:ASCII)EXTERNAL  
29 BEGIN (* MAIN *)  
30 NAME (INDEV,TRACHT,OUDEV,OUTACT,TRHAI,OURAI) ;  
31 REWRITE (OUTP,OURAI,OUTACT,OUDEV) ;  
32 CH:=? ? ;  
33 WHILE ORD(CH) <> 37 DO  
34 BEGIN (* WHILE *)  
35 RDASCT(CH) ;  
36 IF (ORD(CH) = 140) AND (ORD(CH) = 40) OR (ORD(CH) = 11) THEN  
37 WRITE (OUTP, CH) ;  
38 ELSE  
39 IF (ORD(CH) = 12) OR (ORD(CH) = 15) THEN  
40 WRITELN (OUTP) ;  
41 END (* WHILE *)  
42 END.
```

NO ERROR DETECTED
TOTAL PROGRAM SIZE 004372
OUTERMOST DATA SIZE 000540
RESERVED STACK & HEAP 002222

PAS ,CONTR/LW120=CONINT

01234567890123456789	** RSX-11H V3.2 **	30-AUG-83	16:20:13	DKO:11,25JRDASCI.1.9112	01234567890123456789
01234567890123456789	** RSX-11H V3.2 **	30-AUG-83	16:20:13	DKO:11,25JRDASCI.1.9112	01234567890123456789
01234567890123456789	** RSX-11H V3.2 **	30-AUG-83	16:20:13	DKO:11,25JRDASCI.1.9112	01234567890123456789

RRRRRRRR	DDDDDDDD	AAAAAA	SSSSSSSS	CCCCCCCC	IIIIII	
RRRRRRRR	DDDDDDDD	AAAAAA	SSSSSSSS	CCCCCCCC	IIIIII	
RR RR	DD DD	AA AA	SS	CC	II	
RR RR	DD DD	AA AA	SS	CC	II	
RR RR	DD DD	AA AA	SS	CC	II	
RR RR	DD DD	AA AA	SS	CC	II	
RRRRRRRR	DD DD	AA AA	SSSSSS	CC	II	
RRRRRRRR	DD DD	AA AA	SSSSSS	CC	II	
RR RR	DD DD	AAAAAAAAAA	SS	CC	II	
RR RR	DD DD	AAAAAAAAAA	SS	CC	II	
RR RR	DD DD	AA AA	SS	CC	II
RR RR	DD DD	AA AA	SS	CC	II
RR RR	DDDDDDDD	AA AA	SSSSSSSS	CCCCCCCC	IIIIII
RR RR	DDDDDDDD	AA AA	SSSSSSSS	CCCCCCCC	IIIIII

LL	SSSSSSSS	TTTTTTTT	TTTT	?????	
LL	SSSSSSSS	TTTTTTTT	TTTT	?????	
LL	SS	TT	TTTT	??	??
LL	SS	TT	TTTT	??	??
LL	SS	TT			??
LL	SS	TT			??
LL	SSSSSS	TT	TTTT	??	??
LL	SSSSSS	TT	TTTT	??	??
LL	SS	TT	TTTT	??	??
LL	SS	TT	TTTT	??	??
LL	SS	TT	??	??	??
LL	SS	TT	??	??	??
LLLLLLLLLL	SSSSSSSS	TT	??	?????????	
LLLLLLLLLL	SSSSSSSS	TT	??	?????????	

01234567890123456789	** RSX-11H V3.2 **	30-AUG-83	16:20:13	DKO:11,25JRDASCI.1.9112	01234567890123456789
01234567890123456789	** RSX-11H V3.2 **	30-AUG-83	16:20:13	DKO:11,25JRDASCI.1.9112	01234567890123456789
01234567890123456789	** RSX-11H V3.2 **	30-AUG-83	16:20:13	DKO:11,25JRDASCI.1.9112	01234567890123456789

STRONA 15
 STRON 46
 N=RES 5081

```

1      ;
2      ; PROCEDURE R0ASCL
3      ;
4      ; READ ONE ASCII CHAR FROM PR:
5      ;
6      ; INPUT = R5:2  ADDRESS OF RESULT BYTE
7      ;          R5  LINK
8      ;
9      ; OUTPUT = R5 STACK CLEARED
10     ; READ CHAR TRANSFERRED TO SPECIFIED LOCATION
11     ;          END OF DATA = CHARACTER (377)
12     ;
13     ;
14     ; .RCALL QIO#,ALIB#S,DIR#,R5I0#,R5I1#,R5I2#
15 000000 R0ASCL: ALIB#S 21# PR:40      ;CLR 1 IS PR:
16 000020 005725      ;LST (R5):      ;CLR STACK
17 000022      ;DIR# 20PB      ;READ CHAR
18 000030 103004      ;RDC 10#      ;(QUE)
19 000032 004767 000042      ;CALL STCHK      ;CHECK STATUS
20 000034 103432      ;RCS EXT      ;ERRR
21 000040 000770      ;RR PRI      ;TRY AGAIN
22 000042      ;10#: R5I2# 21      ;WAIT FOR IO COMPLETION
23 000054 126777 000052 000000G      ;CMP# S15,115,SUC      ;SUCCESS?
24 000062 001020      ;DNE EXT      ;EXIT TASK
25 000064 042767 000700 000044      ;DIC #200,BUF      ;CLEAR PARITY BIT
26 000072 116735 000040      ;MOVE BUF,#(R5)+      ;TRANSFER CHAR
27 000074 000207      ;RETURN      ;RETURN TO PROGRAM
28 000100 126727 000000G 000000G STCHK:      ;CMP# #0S4,#1E,UPN      ;INSUFFICIENT DYNAMIC
29 000104 000261      ;SEC      ;ERRR INDICATOR
30 000110 001004      ;DNE 20#      ;EXIT
31 000112      ;R5I2#      ;WAIT FOR SIGNIFICANT E
32 000120 000241      ;CLC      ;IO.K
33 000122 000207      ;20#: RETURN      ;RETURN
34 000124 112735 000377      ;EXT: MOVE #377,#(R5)+      ;END OF DATA
35 000130 000207      ;RETURN      ;ERRR RETURN
36 000132      ;S15: .BLKW 2      ;STATUS BLOCK
37 000134 000000      ;BUF: .WORD 0      ;BUFFER
38 000140      ;DPB: QIO# IO.RVR,1,1,,S15,,(BUF,1)      ;INDRECTIVE PARAMETERS
39      ; .END

```

STRONA 16
STRON 46
N: REJ 5081

MEMR. MARKU 1111a 30 011. 60 10020 PAGE 1 1
SYMBOL TABLE

BUF	000130R	BU.RVB- ***** LA	U.IUEF	000000	U.IUPE	000007	SIS	000137R	
DFB	000140R	IS.SHE ***** GX	U.IUFI	000007	U.IUSE	000010	IRSI	***** GX	
EXI	000124R	PR1	000022R	U.IUIH	000004	IRASCI	000000R	\$\$\$AKR	000007
IE.UFN	***** GX	U.IU6E	000012	U.IUFI	000014	SIRK	000100R	\$\$\$USI	000014

. ABS. 000000 000
000170 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 1022 WORDS (4 PAGES)

DYNAMIC MEMORY: 3086 WORDS (11 PAGES)

ELAPSED TIME: 00:00:11

,RDASCI=RDASCI

STRON A-17
STRON 4600w
N=REJ 5084

01234567890123456789	** RSX-11M V3.2 **	30-AUG-83	08:35:33	DKO:11,25JCONHEX.18112	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	30-AUG-83	08:35:33	DKO:11,25JCONHEX.18112	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	30-AUG-83	08:35:33	DKO:11,25JCONHEX.18112	01234567890123456789

CCCCCCCC	000000	RR	RR	HH	HH	EEEEEEEE	XX	XX	
CCCCCCCC	000000	RR	RR	HH	HH	EEEEEEEE	XX	XX	
CC	00	00	RR	RR	HH	HH	EE	XX	XX
CC	00	00	RR	RR	HH	HH	EE	XX	XX
CC	00	00	RRRR	RR	HH	HH	EE	XX	XX
CC	00	00	RRRR	RR	HH	HH	EE	XX	XX
CC	00	00	RR	RR	RR	HHHHHHHHHH	EEEEEEEE	XX	
CC	00	00	RR	RR	RR	HHHHHHHHHH	EEEEEEEE	XX	
CC	00	00	RR	RRRR	RR	RR	EE	XX	XX
CC	00	00	RR	RRRR	RR	RR	EE	XX	XX
CC	00	00	RR	RR	RR	RR	EE	XX	XX
CC	00	00	RR	RR	RR	RR	EE	XX	XX
CCCCCCCC	000000	RR	RR	HH	HH	EEEEEEEE	XX	XX
CCCCCCCC	000000	RR	RR	HH	HH	EEEEEEEE	XX	XX

LL	SSSSSSSS	TTTTTTTT	TTTTTTTT	222222	
LL	SSSSSSSS	TTTTTTTT	TTTTTTTT	222222	
LL	SS	TT	TTTT	22	22
LL	SS	TT	TTTT	22	22
LL	SS	TT	TTTT		22
LL	SS	TT	TTTT		22
LL	SSSSSS	TT	TTTT	22	
LL	SSSSSS	TT	TTTT	22	
LL	SS	TT	TTTT	22	
LL	SS	TT	TTTT	22	
LL	SS	TT	TTTT	22	
LL	SS	TT	TTTT	22	
LLLLLLLLLL	SSSSSSSS	TT	TTTT	22222222	
LLLLLLLLLL	SSSSSSSS	TT	TTTT	22222222	

wotany ja

01234567890123456789	** RSX-11M V3.2 **	30-AUG-83	08:35:33	DKO:11,25JCONHEX.18112	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	30-AUG-83	08:35:33	DKO:11,25JCONHEX.18112	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	30-AUG-83	08:35:33	DKO:11,25JCONHEX.18112	01234567890123456789

STRONA-18
STRON 46
N. RE. 5081

PROGRAM CONHEX (INPUT & OUTPUT) #

```

1  (* THIS PROGRAM SCANS INPUT FILE LOOKING FOR HEX NUMBERS *)
2  (* AND CONVERTING THEM TO DECIMAL *)
3  (* *)
4  (* *)
5  (* AUTHOR : R. DONALD TYRCHA *)
6  (* *)
7  CURST
8      MAXNUM=5 ; (* MAX NUMBER OF DIGITS IN HEX NUMBER *)
9      MAXNAM = 20 ; (* NAME LENGTH *)
10     MAXTBL=15 ; (* TABLE SIZE *)
11     MAXREAL=1.7014111E+30 ;
12 TYPE
13     VECT = ARRAY ( 0 .. MAXNUM ) OF CHAR ;
14     TABL = ARRAY ( 0 .. MAXTBL ) OF CHAR ;
15     NAMTAB = ARRAY ( 0 .. MAXNAM ) OF CHAR ;
16     HEX = 'A' .. 'F' ; (* HEX DIGITS *)
17     DEC = '0' .. '9' ; (* DECIMAL DIGITS *)
18 VAR
19     INNAM, OURNAM : NAMTAB ;
20     INDEC, OUDDEC, IRACCT, OUDACT : TABL ;
21     HEXDIG : SET OF HEX ;
22     DECDIG : SET OF DEC ;
23     RUN : VECT ;
24     OCTNUM : INTEGER ;
25     TBLX : INTEGER ;
26 (* USE FILE *)

```

```

1 (* THIS MODULE IS INPUT BY COMPUTER *)
2 PROCEDURE NAME (VAR INDEV; INACT; TABL; VAR OUDEV; OUACT; TABL;
3     VAR INNAM; OUNAM : NAME);
4 (*
5 (*****
6 (*
7 (*
8 (*
9 (* GLOBAL DATA :
10 (* CONST MAXNAM = ? -- NAME LENGTH
11 (* MAXTAB = ? -- ARRAY SIZE
12 (* TYPE TABL = ARRAY [0 .. MAXTAB] OF CHAR;
13 (* INNAM = ARRAY [0 .. MAXNAM]
14 (* OF CHAR
15 (* VAR INNAM : NAME; -- INPUT FILE NAME
16 (* OUNAM : NAME; -- OUTPUT FILE NAME
17 (* INDEV : TABL; -- INPUT DEVICE NAME
18 (* OUDEV : TABL; -- OUTPUT DEVICE
19 (* INACT : TABL; -- INPUT ACCOUNT
20 (* OUACT : TABL; -- OUTPUT ACCOUNT
21 (*
22 (*
23 (*****
24 (* GET INPUT AND OUTPUT FILE SPECIFICATION.
25 (*
26 (*
27 (* VERSION 00 DATE : 29.08.83
28 (* AUTHOR : R. DONALD TYRCHA
29 (*
30 (*****
31 (*
32 CONST
33     MAXLIN = 80; (* LINE BUFFER SIZE *)
34 TYPE
35     LINE = ARRAY [0 .. MAXLIN] OF CHAR;
36 VAR
37     CMD : LINE;
38     INDEV, INACT, OUDEV, OUACT : INTEGER; (* LENGTHS *)
39     I, PTR, MAX : INTEGER;
40     RES : INTEGER;
41     INNAM, OUNAM : NAME; (* LENGTHS *)
42     OK : BOOLEAN;
43 (*
44 (*****
45 (*
46 PROCEDURE SKIP ( VAR PTR : INTEGER; MAX : INTEGER);
47 (* SKIP SPACES *)
48 BEGIN (* SKIP *)
49     WHILE (CMD [PTR] = ' ') AND (PTR <= MAX) DO PTR := PTR+1;
50 END; (* SKIP *)
51 (*
52 (*****
53 (*
54 PROCEDURE REANAM ( VAR DEV : TABL; VAR DEVLE : INTEGER;
55     VAR ACT : TABL; VAR ACTLE : INTEGER;
56     VAR NAM : NAME; VAR NAMLE : INTEGER;

```

STRONA 21
 STRON 46
 N=PEJ 50 81

```

57      VAR PTR : INTEGER; MAX : INTEGER;
58      VAR OK : BOOLEAN;
59      (* READ NAME AND PUT ITS PARTS INTO APPROPRIATE ARRAYS *)
60      VAR
61          PTC,I,J : INTEGER;
62          AC : BOOLEAN;
63      BEGIN (* REANAM *) OK := TRUE; SKIP (PTR,MAX); PTC := PTR;
64      WHILE NOT (CMD [PTC] IN [' ','-','_','/','\']) AND (PTC <= MAX) DO
65          PTC := PTC + 1;
66      IF CMD [PTC] = '?' THEN
67          BEGIN J := 0;
68              FOR I:=PTR TO PTC DO
69                  BEGIN DEV [J] := CMD [I]; J := J+1
70              END; (* FOR *)
71              DEVLE := J; PIR := PTC+1; SKIP (PIR,MAX); PTC := PIR
72          END; (* IF *)
73      WHILE NOT (CMD [PTC] IN ['L','J','=','/','\']) AND (PTC <= MAX) DO
74          PTC := PTC + 1;
75      IF CMD [PTC] = 'L' THEN
76          BEGIN J := 0;
77              WHILE ( CMD [PTR] (> 'J') AND (PTR <= MAX) ) DO
78                  BEGIN ACNT [J] := CMD [PTR]; PIR := PIR+1; J := J+1
79              END; (* WHILE *)
80              IF CMD [PTR]='J' THEN
81                  BEGIN ACNT [J] := CMD [PIR]; PIR := PIR +1; J := J+1
82              END (* IF *)
83              ELSE
84                  BEGIN OK := FALSE; WRITELN(' -- ''J'' MISSING. ');
85              END (* ELSE *)
86              ACNTLE := J; SKIP (PIR,MAX)
87          END (* IF *)
88          ELSE
89              IF CMD [PTC] = 'J' THEN
90                  BEGIN OK := FALSE; WRITELN(' -- ''L'' MISSING. ');
91              END; (*IF *) J := 0;
92              WHILE (CMD [PTR] (> '=') AND (PTR <= MAX) AND (CMD[PTR] (> ' ')) ) DO
93                  BEGIN NAM [J] := CMD [PTR]; J := J+1; PIR := PIR + 1
94              END; (* WHILE *) NAME := J;
95              SKIP (PTR,MAX);
96              IF CMD [PTR] = '=' THEN PTR := PTR + 1
97          END; (* REANAM *)
98      (*
99      (*****
100     (*
101     PROCEDURE GCML (VAR S : LINE; VAR LEN : INTEGER);
102     EXTERN;
103     (* GET HCR COMMAND LINE *)
104     (*
105     (*****
106     (*
107     BEGIN (* NAME *)
108     FOR I := 0 TO MAXNAM DO
109     BEGIN OUNAM [I] := ' '; INNAM [I] := ' '
110     END; (* FOR *)
111     FOR I := 0 TO MAXTEL DO
112     BEGIN INDEV [I] := ' '; OUDEV [I] := ' '; INACNT [I] := ' '

```

PASCAL PDP-11 VERSION 6.01
GETNAM.PAS

1983-08-30
NAME

08:33:02

PAGE 4

```
113     QUACNT [I] := ' '  
114     END; (* FOR *)  
115     GCNL (CMD,MAX); PTR := 5; (* CMD POINTER *)  
116     REAHAM (OUDEV,OUDELE,OUACNT,OUACLE,OUNAM,OUNMLE,PTR,MAX,OK);  
117     IF OK THEN  
118         REAHAM (INDEV,INDELE,INACHT,INACLE,INNAM,INMML,PTR,MAX,OK)  
119     END; (* NAME *)
```



```

27
28  (*****)
29  PROCEDURE NEXTCH (VAR CH:CHAR ; VAR NXLIN : BOOLEAN ; VAR ETX : BOOLEAN ) ;
30  (* READ NEXT CHAR AND TELL IF EOLN ENCOUNTERED*)
31  BEGIN (* NEXTCH *)
32    CH := ' ' ; NXLIN := FALSE ; ETX := FALSE ;
33    IF NOT EOF THEN
34      IF NOT EOLN THEN READ (CH)
35      ELSE
36        BEGIN (* ELSE *)
37          READLN ;
38          IF NOT EOF THEN READ ( CH) ELSE ETX := TRUE ;
39          NXLIN := TRUE
40        END (* ELSE *)
41      ELSE ETX := TRUE
42    END ; (* NEXTCH *)
43  (*****)
44  PROCEDURE READHX (VAR HEXNUM : VECT ; VAR OK : INTEGER ) ;
45  (* READ NEXT HEX NUMBER FROM INPUT *)
46  VAR
47    DIG : CHAR ;
48    NEWLIN : BOOLEAN ;
49    EOLP,ETX : BOOLEAN ;
50    I : INTEGER ;
51  BEGIN (* READHX *)
52    ETX := FALSE ; EOLP := FALSE ;
53    WHILE NOT EOLP DO
54      BEGIN (* WHILE *)
55        FOR I := 0 TO MAXNUM DO HEXDIG (. I .) := '0' ;
56        OK := -1 ;
57        NEXTCH ( DIG, NEWLIN, ETX ) ;
58        WHILE NOT ( DIG IN DEC DIG ) AND NOT ETX DO
59          BEGIN (* WHILE *)
60            IF NEWLIN THEN WRITELN ;
61            IF NOT ETX THEN WRITE ( DIG ) ;
62            NEXTCH ( DIG, NEWLIN, ETX )
63          END ; (* WHILE *)
64          WHILE (DIG IN DEC DIG + HEXDIG) AND (OK < MAXNUM)
65            AND NOT NEWLIN AND NOT ETX DO
66            BEGIN (* WHILE *)
67              OK := OK + 1 ;
68              HEXNUM (. OK .) := DIG ;
69              NEXTCH ( DIG, NEWLIN, ETX )
70            END ; (* WHILE *)
71            IF OK >= MAXNUM THEN
72              BEGIN (* IF *)
73                FOR I := 0 TO OK DO WRITE ( HEXNUM (. I .) ) ;
74                IF NEWLIN THEN BEGIN WRITELN ; WRITE (DIG) END
75              ELSE
76                BEGIN (* ELSE *)
77                  WHILE NOT EOLN AND NOT EOF DO
78                    BEGIN (* WHILE *)
79                      WRITE (DIG) ; READ (DIG)
80                    END (* WHILE *) ;
81                    WRITE (DIG) ;
82                  END (* ELSE *)
83                END (* IF *)

```

```

84     ELSE
85     IF NOT E1X THEN
86     IF ( DIG (> 'H' ) OR NEWLIN THEN
87     BEGIN (* IF *)
88     FOR I := 0 TO OK DO WRITE ( HEXNUM ( . I . ) ) ;
89     IF NEWLIN THEN WRITELN ;
90     WRITE ( DIG )
91     END (* IF *)
92     ELSE EOLP := TRUE
93     ELSE BEGIN OK := -1 ; EOLP := TRUE END
94     END (* WHILE NOT EOLP *)
95     END ; (* READHX *)
96     (*****
97     PROCEDURE CONVHX ( VAR HEXINT:VECT ; GO : INTEGER ) ;
98     (* CONVERT HEX NUMBER FROM HEXINT TO DECIMAL *)
99     (* AND PRINT IT IN OUTPUT FILE *)
100    VAR
101    RESINT : INTEGER ;
102    PROCEDURE CVHXOC ( VAR RESULT : INTEGER ; GO:INTEGER ;
103    VAR HEXINT : VECT ) ; EXTERN ;
104    BEGIN (* CONVHX *)
105    IF ( GO >= 0 ) AND ( GO <= MAXNUM ) THEN
106    BEGIN (* IF GO *)
107    RESINT := 0 ;
108    CVHXOC ( RESINT , GO-1 , HEXINT ) ;
109    (* RESINT = CONVERTED NO. *)
110    WRITE ( RESINT : 6 : 0 )
111    END (* IF GO *)
112    END ; (* CONVHX *)
113    (*****
114    BEGIN (* MAIN *)
115    NAME ( INDEV , INACR1 , OUDEV , OUACR1 , INNAH , OUNAH ) ;
116    RESET ( INPUT , INNAH , INACR1 , INDEV ) ;
117    REWRITE ( OUTPUT , OUNAH , OUACR1 , OUDEV ) ;
118    ISHEX := 0 ;
119    DECDIG := ['0' .. '9'] ;
120    HEXDIG := DECDIG : ['A' .. 'F'] ;
121    WHILE ISHEX >= 0 DO
122    BEGIN (* WHILE *)
123    READHX ( NUM , ISHEX ) ;
124    CONVHX ( NUM , ISHEX )
125    END (* WHILE *)
126    END.

```

NO ERROR DETECTED
 TOTAL PROGRAM SIZE 007246
 OUTERMOST DATA SIZE 000210
 RESERVED STACK & HEAP 001710

PAS ,CONNEX/LW120=CONNEX

```

1  C-
2  C-
3  C-
4  C-
5  C-  UNIVERZITA OBR FEE
6  C-  IN FTEL STADUARD.
7  C-
8  C-  VERJUD O  DAF # 12345678
9  C-  AUTHORS # KAROL HAJAR
10 C-  R. JORNAL TYCHA
11 C-
12 C-
13 C-
14 CONST
15     MAXLIN = 80 (* LINE BUFFER SIZE *)
16     MAXTBL = 15 (* NAME BUFFER SIZE *)
17 TYPE
18     LINE = ARRAY (. 0 .. MAXLIN .) OF CHAR?
19     TABL = ARRAY (. 0 .. MAXTBL .) OF CHAR?
20 VAR
21     INDEV, INACT  : TABL?
22     OUDEV, OUACT  : TABL?
23     INNAM, ONAM, OND : LINE?
24     INDELE, INACTE, OUDELE, OUACTE : INTEGER? (* LENGTHS *)
25     I, PTR, MAX : INTEGER?
26     RES : INTEGER?
27     INNAMLE, ONAMLE : INTEGER? (* LENGTHS *)
28     OK : BOOLEAN?
29 (*
30 (*
31 (*
32 PROCEDURE PARADIG (VAR OUNN : LINE? OUNALE : INTEGER?
33     VAR OUNACT : TABL? OUNACTE : INTEGER?
34     VAR OUDEV : TABL? OUDELE : INTEGER?
35     VAR INNN : LINE? INNAMLE : INTEGER?
36     VAR INACT : TABL? INACTE : INTEGER?
37     VAR INDEV : TABL? INDELE : INTEGER?
38     VAR RESLRR : INTEGER?)
39 EXTERN?
40 (*
41 (*
42 (*
43 PROCEDURE SKIP ( VAR PTR : INTEGER? MAX : INTEGER?)
44 (* SKIP SPACES *)
45 BEGIN (* SKIP *)
46     WHILE (CHR (PTR) = ' ') AND (PTR <= MAX) DO PTR := PTR+1;
47 END (* SKIP *)
48 (*
49 (*
50 (*
51 PROCEDURE REANAM ( VAR DEV : TABL? VAR DEVLE : INTEGER?
52     VAR ACT : TABL? VAR ACTLE : INTEGER?
53     VAR NAM : LINE? VAR NAMLE : INTEGER?
54     VAR PTR : INTEGER? MAX : INTEGER?
55     VAR OK : BOOLEAN )?
56 (* READ NAME AND PUT ITS PARTS INTO APPROPRIATE ARRAYS *)

```

```

57 VAR
58     PTC:1;J = INITIAL?
59     AC = BOOLEAN?
60 BEGIN (* REARRANGE *) OR := TRUE? SKIP (PTR,MAX); PTC := PTR?
61     WHILE NOT (CHD LEFT) IN L?"?"?"?" ??"?" AND (PTR (= MAX) DO
62         PTC := PTC + 1?
63     IF CHD LEFT) = "?" THEN
64         BEGIN J := 0?
65             FOR I = PTR TO PTC DO
66                 BEGIN DEV LJI := CHD LID? J = J+1
67                     END? (* FOR *)
68                 DEVLE := J? PTR := PTC+1? SKIP (PTR,MAX)? PTC := PTR
69             END? (* IF *)
70         WHILE NOT (CHD LEFT) IN L?"?"?"?" ??"?" AND (PTR (= MAX) DO
71             PTC := PTC + 1?
72         IF CHD LEFT) = "?" THEN
73             BEGIN J := 0?
74                 WHILE ( CHD LEFT) (> ??) AND (PTR (= MAX) DO
75                     BEGIN ACN1 LJI := CHD LEFT)? PTR := PTR+1? J = J+1
76                         END? (* WHILE *)
77                     IF CHD LEFT) = "?" THEN
78                         BEGIN ACN1 LJI := CHD LEFT? PTR := PTR+1? J = J+1
79                             END? (* IF *)
80                         ELSE
81                             BEGIN OR := FALSE? WRITE(LR?" - "?" MISSING.")
82                                 END? (* ELSE *)?
83                         ACNILE := J? SKIP (PTR,MAX)
84                     END? (* IF *)
85                 ELSE
86                     IF CHD LEFT) = "?" THEN
87                         BEGIN OR := FALSE? WRITE(LR?" - "?" MISSING.")
88                             END? (* IF *)? J = 0?
89                     WHILE (CHD LEFT) (> ??) AND (PTR (= MAX) AND (CHD LEFT) (> ??) DO
90                         BEGIN NAY LJI := CHD LEFT)? J = J+1? PTR := PTR + 1
91                             END? (* WHILE *)? NAYLE := J?
92                         SKIP (PTR,MAX)?
93                     IF CHD LEFT) = "?" THEN PTR := PTR + 1
94                 END? (* REARRANGE *)
95             (*
96             (*****
97             *)
98             PROCEDURE GCHL (VAR G = LINE? VAR LEN = INTEGER)?
99             EXTERN?
100             (* GET NUR CURRENT LINE *)
101             (*
102             (*****
103             *)
104             BEGIN (* MAIN *)
105                 FOR I = 0 TO MAXLIN DO
106                     BEGIN CURAM LII := ? ?? INNAM LII := ? ??
107                         END? (* FOR *)
108                 FOR I = 0 TO MAXTEL DO
109                     BEGIN INDEV LII := ? ?? OUBEV LII := ? ?? INAKO LII := ? ??
110                         QUACT LII := ? ??
111                     END? (* FOR *)
112                 GCHL (CHD,MAX)? PTR := 5? (* CHD POINTER ?)

```

STRONA 27
STRON 46
N=REG, 5084

PHYSICAL PAGE=11 VERSION 2.01 0003-08-00 10054106 PAGE 1
NAME.PAN NAME

```
113 RETURN (CODE%*INDELE%*CINACT%*COUNCLE%*COUNR%*CENR%*P%*INVTAX%*DRO%*  
114 IF OR THEN  
115 BEGIN  
116 RETURN (CODE%*INDELE%*CINACT%*CINACT%*INACT%*LIBR%*CENR%*C%*INVTAX%*DRO%*  
117 IF OR AND (INR%LE > ?) THEN  
118 MANAGER (CINR%*COUNR%*COUNR%*COUNR%*COUNR%*COUNR%*COUNR%*COUNR%*  
119 INR%*INR%*INR%*INR%*INR%*INR%*INR%*INR%*INR%*INR%*INR%*INR%*INR%*  
120 END (* IF *)  
121 END.
```

NO ERROR DETECTED
TOTAL PROGRAM SIZE 004057
OUTERHOST DATA SIZE 000516
RESERVED STACK & HEAP 001530

PAS /NAME/LW120-NAME

STRONA 28
STRON 46
N-REF. 5081

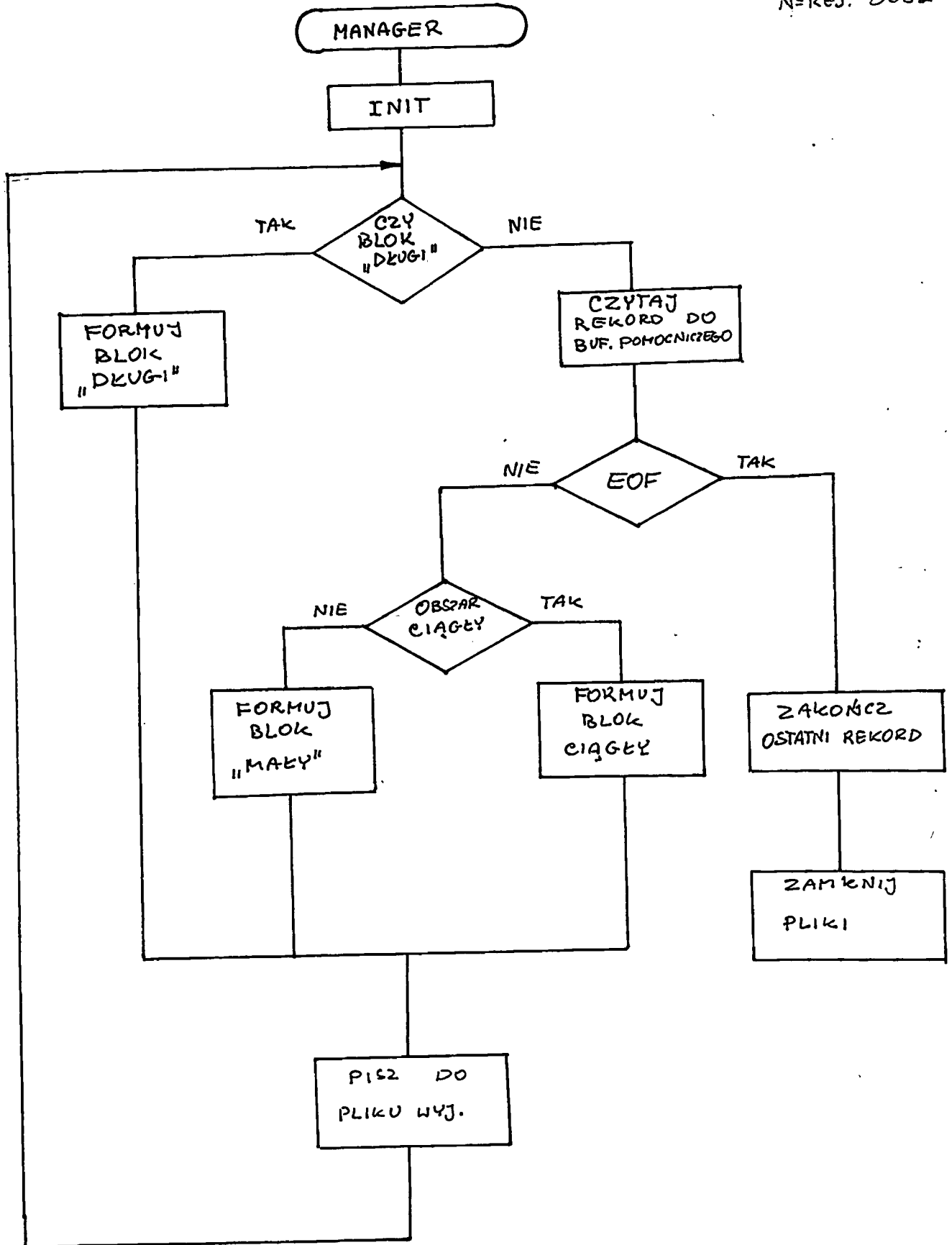
PROGRAM NAME (TTY)

01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	07:55:48	DK0: E7, 2021MANAGER.LST#1	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	07:55:48	DK0: E7, 2021MANAGER.LST#1	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	07:55:48	DK0: E7, 2021MANAGER.LST#1	01234567890123456789

MM	MM	AAAAAA	NN	NN	AAAAAA	GGGGGGGG	EEEEEEEEEE	RRRRRRRR					
MM	MM	AAAAAA	NN	NN	AAAAAA	GGGGGGGG	EEEEEEEEEE	RRRRRRRR					
MMMM	MMMM	AA	AA	NN	NN	AA	AA	GG	EE	RR	RR		
MMMM	MMMM	AA	AA	NN	NN	AA	AA	GG	EE	RR	RR		
MM	MM	MM	AA	AA	NNNN	NN	AA	AA	GG	EE	RR	RR	
MM	MM	MM	AA	AA	NNNN	NN	AA	AA	GG	EE	RR	RR	
MM	MM	AA	AA	NN	NN	NN	AA	AA	GG	EEEEEEEE	RRRRRRRR		
MM	MM	AA	AA	NN	NN	NN	AA	AA	GG	EEEEEEEE	RRRRRRRR		
MM	MM	AAAAAAAAAAAA	NN	NNNN	AAAAAAAAAAAA	GG	GGGGGG	EE	RR	RR			
MM	MM	AAAAAAAAAAAA	NN	NNNN	AAAAAAAAAAAA	GG	GGGGGG	EE	RR	RR			
MM	MM	AA	AA	NN	NN	AA	AA	GG	GG	EE	RR	RR
MM	MM	AA	AA	NN	NN	AA	AA	GG	GG	EE	RR	RR
MM	MM	AA	AA	NN	NN	AA	AA	GGGGGG	EEEEEEEEEE	RR	RR	
MM	MM	AA	AA	NN	NN	AA	AA	GGGGGG	EEEEEEEEEE	RR	RR	

LL	SSSSSSSS	TTTTTTTTT	???	11
LL	SSSSSSSS	TTTTTTTTT	???	11
LL	SS	TT	???	1111
LL	SS	TT	???	1111
LL	SS	TT		11
LL	SS	TT		11
LL	SSSSSS	TT	???	11
LL	SSSSSS	TT	???	11
LL	SS	TT	???	11
LL	SS	TT	???	11
LL	SS	TT	??	11
LL	SS	TT	??	11
LLLLLLLLLL	SSSSSSSS	TT	??	111111
LLLLLLLLLL	SSSSSSSS	TT	??	111111

01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	07:55:48	DK0: E7, 2021MANAGER.LST#1	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	07:55:48	DK0: E7, 2021MANAGER.LST#1	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	07:55:48	DK0: E7, 2021MANAGER.LST#1	01234567890123456789



Przeptyw informacji w programie "MANAGER"


```

1          .TITLE  MANAGER
2
3          ;PROGRAM PRZYBOTOWIJACY PARAMETRY
4          ;DO FORMALOWANIA INFORMACJI
5          ;W POSTACI TASYMI INIELA
6          ;
7          ;AUTOR KAROL NAJAR
8          ;
9          ;WARSAWA
10         ;
11         ;HERA - PIAP
12         ;
13         ;AUGUST 1983 ROK
14         ;
15         ;
16         ;
17 000000      MANAGER::
18 000000 004767 000000G      CALL INIT          ;INICJACJA ZBIOROW
19 000004 004767 000000G      CALL OPNE          ;OTWARCIE ZBIORU WEJSCIOWEGO
20 000010 016701 000520      MOV ABUF,R1          ;PIERWSZY BUFOR
21 000014 012705 000520      MOV #BDR,R5          ;ADRES PARAMETROW FORMALOWANIA
22 000020 010104              MOV R1,R4          ;KOPIA
23 000022 004767 000000G      CALL READ          ;CZYTAJ PIERWSZY REKORD
24 000026 005721              TST (R1)+          ;PRZYBOTOWIJANIE POINTERA
25 000030 010164 177774      MOV R1,-4(R4)      ;ZAPIS POINTERA
26 000034 162764 000002 177776      SUB #2,-2(R4)      ;ZMIANA LICZNIKA
27 000042 016704 000466      1#: MOV ABUF,R4          ;OBROBKA GLOWNEGO POINTERA
28 000046 004767 000420      CALL MODULO          ;
29 000052 026402 177776      CMP -2(R4),R2      ;LICZNIK
30 000056 103416              BLO 2#           ;NIEJSZY
31 000060 016415 177774      MOV -4(R4),(R5)    ;PRZEPISANIE POINTERA
32 000064 011465 000004      MOV (R4),4(R5)    ;      "      ADRESU
33 000070 010265 000002      MOV R2,2(R5)     ;      "      LICZNIKA
34 000074 060264 177774      ADD R2,-4(R4)    ;AKTUALIZACJA POINTERA
35 000100 160264 177776      SUB R2,-2(R4)    ;      "      LICZNIKA
36 000104 060214              ADD R2,(R4)       ;      "      ADRESU
37 000106 004767 000000G      CALL FORBLO      ;FORMATOWANIE
38 000112 000753              BR 1#           ;
39
40 000114 016701 000416      2#: MOV ABUF+2,R1     ;BUFOR POMOCNICZY
41 000120 010103              MOV R1,R3          ;KOPIA
42 000122 004767 000000G      CALL READ          ;CZYTAJ REKORD
43 000126 005721              TST (R1)+          ;PRZYBOTOWIJANIE POINTERA
44 000130 010163 177774      MOV R1,-4(R3)     ;POINTER
45 000134 162763 000002 177776      SUB #2,-2(R3)     ;MODYFIKACJA LICZNIKA
46 000142 105763 177772      TSTB -6(R3)      ;CZY EOF ?
47 000146 100026              BPL 3#           ;NIE
48 000150 005764 177776      TST -2(R4)       ;CZY COS ZOSTALO ?
49 000154 001411              BEQ 21#          ;NIE
50 000156 016415 177774      MOV -4(R4),(R5)   ;TAK - POINTER
51 000162 016465 177776 000002      MOV -2(R4),2(R5) ;LICZNIK
52 000170 011465 000004      MOV (R4),4(R5)   ;ADRES
53 000174 004767 000000G      CALL FORBLO      ;FORMATOWANE
54 000200 005015              CLR (R5)          ;BLOK KONCA
55 000202 005065 000002      CLR 2(R5)        ;
56 000206 005065 000004      CLR 4(R5)        ;
57 000212 004767 000000G      CALL FORBLO      ;FORMATOWANIE

```

58	000216	004767	000000G		CALL CLOSE	;%ZACONCZENIE ZAMKNIETIEM ZMIOROW
59	000222	000207			RETURN	;
60	000224	005764	177776	3#:	1ST -2(R4)	;%BLOK PUSTY ?
61	000230	001005			BNE 31#	;%NIE
62	000232	010367	000276		MOV R3,ABUF	;
63	000236	010467	000274		MOV R4,ABUF+2	;%ZMIANA BUFORA
64	000242	000677			BR 1#	;
65	000244	011401		31#:	MOV (R4),R1	;%KOPIA ADRESU
66	000246	066401	177776		ADD -2(R4),R1	;%WYLICZENIE NASTEPNEGO ADRESU
67	000252	020113			CHP R1,(R3)	;%LACZNY ?
68	000254	001420			REQ 4#	;%TAK
69	000256	016465	177776	000002	MOV -2(R4),2(R5)	;%NIE - PRZEPISZ LICZNIK
70	000264	005064	177776		CLR -2(R4)	;%MODYFIKACJA PARAMETROW BUFORA
71	000270	016415	177774		MOV -4(R4),(R5)	;%POINTER
72	000274	011465	000004		MOV (R4),4(R5)	;%ADRES
73	000300	004767	000000G		CALL FORPLU	;%FORMATOWANIE
74	000304	010367	000224		MOV R3,ABUF	;%MODYFIKACJA BUFOROW
75	000310	010467	000222		MOV R4,ABUF+2	;
76	000314	000652			BR 1#	;
77	000316	005764	177776	4#:	1ST -2(R4)	;%ZABEZPIECZENIE BLOKOW KONCA
78	000322	001005			BNE 42#	;%NIE ZEROWY BUFOR
79	000324	010367	000204		MOV R3,ABUF	;%ZEROWY BUFOR
80	000330	010467	000202		MOV R4,ABUF+2	;%MODYFIKACJA
81	000334	000642			BR 1#	;
82	000336	004767	000130	42#:	CALL MODULO	;%BLOK LACZONY
83	000342	016415	177774		MOV -4(R4),(R5)	;%POINTER
84	000346	016465	177776	000002	MOV -2(R4),2(R5)	;%LICZNIK
85	000354	011465	000004		MOV (R4),4(R5)	;%ADRES
86	000360	052765	177400	000002	BIS #-256.,2(R5)	;%LINK WSKAZNIK
87	000366	166402	177776		SUB -2(R4),R2	;%LINK LICZNIK
88	000372	020263	177776		CHP R2,-2(R3)	;
89	000376	103413			BLO 41#	;
90	000400	016365	177774	000006	MOV -4(R3),6(R5)	;%LINK POINTER
91	000406	016365	177776	000010	MOV -2(R3),10(R5)	;%LINK LICZNIK
92	000414	005064	177776		CLR -2(R4)	;%MODYFIKACJA
93	000420	004767	000000G		CALL FORPLU	;%FORMATOWANIE
94	000424	000606			BR 1#	;
95	000426	016365	177774	000006	MOV -4(R3),6(R5)	;%LINK POINTER
96	000434	010265	000010		MOV R2,10(R5)	;%LINK LICZNIK
97	000440	010367	000070		MOV R3,ABUF	;%MODYFIKACJA BUFORA
98	000444	010467	000066		MOV R4,ABUF+2	;
99	000450	060263	177774		ADD R2,-4(R3)	;
100	000454	160263	177776		SUB R2,-2(R3)	;
101	000460	060213			ADD R2,(R3)	;
102	000462	004767	000000G		CALL FORPLU	;%FORMATOWANIE
103	000466	000167	177350		JMP 1#	;
104	000472	012702	000020	MODULO:	MOV #20,R2	;%LICZNIK NA MAX
105	000476	032714	000017		BIT #15.,(R4)	;%CZY NIE MODULO
106	000502	001001			BNE 10#	;
107	000504	000207			RETURN	;
108	000506	011401		10#:	MOV (R4),R1	;%KOPIA ADRESU
109	000510	042701	177760		BIC #-16.,R1	;%OBCIECIE MODULU
110	000514	160102			SUB R1,R2	;%NOWY LICZNIK
111	000516	000207			RETURN	;
112					;	
113	000520			BDB:	.BLKW 6.	
114	000534	000546	000650	ABUF:	.WORD BUF1,BUF2	

115	000540	000000	000550'	000000	PO1:	.WORD	0, BUF1:2,0
116	000546				BUF1:	.BLKB	60.
117	000642	000000	000652'	000000	PO2:	.WORD	0, BUF2:2,0
118	000650				BUF2:	.BLKB	60.
119		000001				.END	

STRONA 34
STRON 46
N.RPS. 5081

MANAGER MACRO M1113 05-SEP-83 07:55 PAGE 1-3
SYMBOL TABLE

ARUF	000534R	BUF2	000650R	INIT = ***** GX	OPENE = ***** GX	P02	000642R
END	000520R	CLOSE = ***** GX	MANAGE	000000RG	P01	000540R	READ = ***** GX
BUF1	000546R	FORBLO= ***** GX	MODULO	000472R			

. ABS. 000000 000
000744 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 154 WORDS (1 PAGES)
DYNAMIC MEMORY: 3086 WORDS (11 PAGES)
ELAPSED TIME: 00:00:07
MANAGER=ZAROU

STRONA 35
STRON 46
ARJ. 5084

01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	08:07:15	DKO:[7,202]FORBL0.LST#1	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	08:07:15	DKO:[7,202]FORBL0.LST#1	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	08:07:15	DKO:[7,202]FORBL0.LST#1	01234567890123456789

FFFFFFFF	000000	RRRRRRRR	BBBBBBB	LL	000000	
FFFFFFFF	000000	RRRRRRRR	BBBBBBB	LL	000000	
FF	00 00	RR RR	BB BB	LL	00 00	
FF	00 00	RR RR	BB BB	LL	00 00	
FF	00 00	RR RR	BB BB	LL	00 00	
FF	00 08	RR RR	BB BB	LL	00 00	
FFFFFFFF	00 00	RRRRRRRR	BBBBBBB	LL	00 00	
FFFFFFFF	00 00	RRRRRRRR	BBBBBBB	LL	00 00	
FF	00 00	RR RR	BB BB	LL	00 00	
FF	00 00	RR RR	BB BB	LL	00 00	
FF	00 00	RR RR	BB BB	LL	00 00
FF	00 00	RR RR	BB BB	LL	00 00
FF	000000	RR RR	BBBBBBB	LLLLLLLLL	000000
FF	000000	RR RR	BBBBBBB	LLLLLLLLL	000000

LL	SSSSSSS	TTTTTTTT	???	11
LL	SSSSSSS	TTTTTTTT	???	11
LL	SS	TT	???	1111
LL	SS	TT	???	1111
LL	SS	TT		11
LL	SS	TT		11
LL	SSSSSS	TT	???	11
LL	SSSSSS	TT	???	11
LL	SS	TT	???	11
LL	SS	TT	???	11
LL	SS	TT	??	11
LL	SS	TT	??	11
LLLLLLLLLLL	SSSSSSS	TT	??	111111
LLLLLLLLLLL	SSSSSSS	TT	??	111111

01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	08:07:15	DKO:[7,202]FORBL0.LST#1	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	08:07:15	DKO:[7,202]FORBL0.LST#1	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	5-SEP-83	08:07:15	DKO:[7,202]FORBL0.LST#1	01234567890123456789

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57

```

.TITLE FORBLU
.MCALL Q10W%S,ALUR%S,EXIT%S

;PROGRAM FORMATOWANIA BLOKU W POSTCI
;BLOKOW HEKSADECYMALNYCH NOTACJI "INTEL"
;AUTOR KAROL NAJAR
;WARSZAWA
;MERA - P1AF
;MAY 1983
;
;BLOK PRZEKAZYWANIA PARAMETROW
;POSIADA POSTAC :
;*****
; POINTER BUFORA ;
;*****
; LICZNIK BAJTOW ; +2
;*****
; ADRES LADOWANIA ( URG ) ; +4
;*****
; LINK POINTER BUFORA ; +6
;*****
; LINK LICZNIK BAJTOW * +10
;*****
;
;
;
;
FORBLD::      MOV R4,-(SP);
MOV R5,--(SP) ;
MOV R3,--(SP) ;
CLR SUM ;ZEROWANIE LICZNIKA SUMY KONTROLNEJ BLOKU
MOV #BHEX,R4 ; USTAWIENIE POINTERA BUFORA
1STB 2(R5) ; CZY BLOK KONCA ?
REQ EOT ; TAK WYKONAJ SKOK
1ST 2(R5) ; KODOWANIE OBSZARU CIAGLEGO?
BPL 1#
MOVB 2(R5),R2 ; PRZYGOTOWANIE LICZNIKA
MOV R2,R0 ; ZACHOWANIE LICZNIKA
ADD 10(R5),R2 ; USTALENIE SUMY
CALL CONHEX ; ZAPIS LICZNIKA BAJTOW
BR 2# ; CONTINUE
1# : MOV 2(R5),R0 ; LICZNIK BAJTOW
MOV R0,R2 ; LICZNIK DLA BLOKU
CALL CONHEX ; ZAPIS LICZNIKA BAJTOW
2# : MOVB #60,(R4)+ ; TYP BLOKU
MOVB #60,(R4)+ ; TYP BLOKU
MOVB 5(R5),R2 ; ADRES LADOWANIA
CALL CONHEX ;
MOVB 4(R5),R2 ; ADRES LADOWANIA
CALL CONHEX ;
MOV (R5),R1 ; ADRES BUFORA WEJSCIOWEGO
3# : MOVB (R1),R2 ; PRZYGOTOWANIE BAJTU DO KODOWANIA
CALL CONHEX ; KODOWANIE
SOB R0,3# ; LICZNIK PETLI

```

```

58 000126 005715          TST (R5)          ; CALY BLOK?
59 000130 100006          BPL 4#            ; TAK ?
60 000132 005015          CLR (R5)          ;***
61 000134 016501 000004   MOV 4(R5),R1      ; NOWY ADRES
62 000140 016500 000006   MOV 6(R5),R0      ; NOWY LICZNIK
63 000144 000764          BR 3#             ; DOKONCZENIE
64 000146                4#:
65 000146 116702 000102   SUCON: MOV8 SUM,R2 ; GENERACJA SUMY KONTROLNEJ
66 000152 105402          NEG8 R2           ; TWORZENIE SUMY
67 000154 004767 000040   CALL CONHEX      ; WSTAWIENIE W BLOK
68 000160 105014          CLR8 (R4)        ; ZAKONCZENIE BUFORA
69 000162 162704 000276   SUB #BUHO,R4     ; DLUGOSC BUFORA
70                          ;ALUN#S #1,#11,#0
71                          ;QIOU#S #10,LVB,#1,#1,.,.,(<#BUHO,R4)
72 000166 004767 000000G   CALL WRITE       ;
73 000172 012603          MOV (SP)+,R3     ;
74 000174 012605          MOV (SP)+,R5     ;
75 000176 012604          MOV (SP)+,R4     ; ODTWORZENIE REJESTROW
76 000200 000207          RETURN          ;
77
78 000202 012702 000060   EOL: MOV #0,R2    ; KONCOWY BLOK
79 000206 012703 000006   MOV #6,R3        ;
80 000212 110224          LOOP1: MOV8 R2,(R4)+ ;
81 000214 077302          SOB R3,LOOP1    ;
82 000216 000753          BR SUCON        ;
83 000220 042702 177400   CONHEX: BIC #-256,R2 ; ZEROWANIE
84 000224 060267 000024   ADD R2,SUM       ; SUMOWANIE
85 000230 010203          MOV R2,R3       ; TYMCZASOWY BUFOR
86 000232 072327 177774   ASH #-4,R3      ; 4* W PRAWO
87 000236 116324 000256   MOV8 TAB(R3),(R4)+ ; PISZ ZNAK
88 000242 042702 177760   BIC #-16,R2     ;
89 000246 116224 000256   MOV8 TAB(R2),(R4)+ ; PISZ ZNAK
90 000252 000207          RETURN          ;
91 000254 000000          SUM: .WORD 0
92                          .NLIST HEX ;
93 000256 060 261 262 TAB: .BYTE 60,261,262,63,264,65,66,267
94 000266 270 071 101 .BYTE 270,71,101,102,303,104,305,306
95 000276 215 012 BUHO: .BYTE 215,12
96 000300 072 BHEX: .ASCII /:/
97 000301          .BLKB 72.
98          .EVEN
99          000001 .END

```

STRONA 38
STRON 46
NREF. 5081

FORBLO MACRO H1113 05-SEP-83 08:07 PAGE 1-2
SYMBOL TABLE

BHEX	000301R	CONHEX	000220R	FORBLO	000000RG	SUCOM	000144R	TAB	000256R
BUHO	000276RG	EOY	000202R	LOOP1	000212R	SUM	000254R	WRITE =	***** BX

. ABS. 000000 000
000412 001

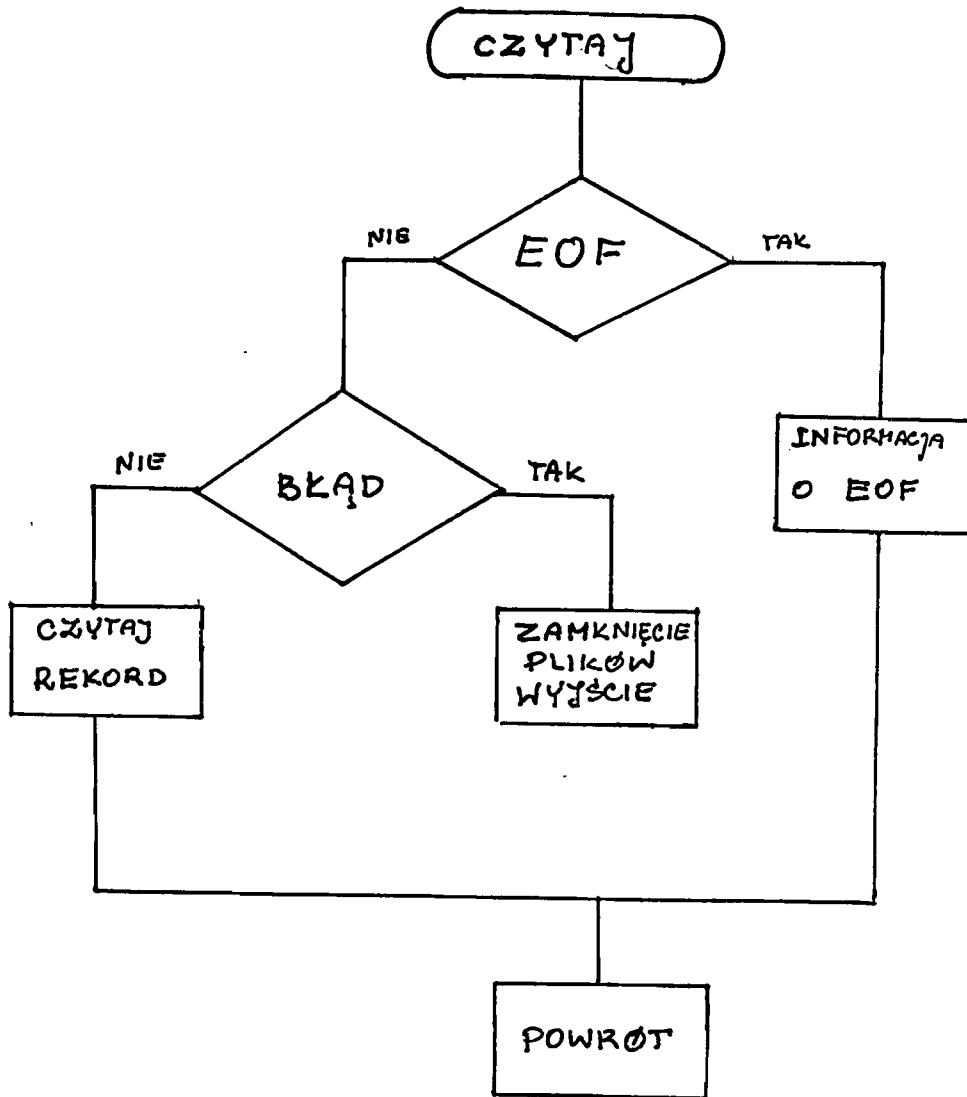
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 356 WORDS (2 PAGES)

DYNAMIC MEMORY: 3086 WORDS (11 PAGES)

ELAPSED TIME: 00:00:07

*FORBLO=FORBLO



Realizacja podprogramów "READ" programu
"FILEOPERATION"

```

1      .TITLE FILEOPERATION
2      .MCALL FDBDF$,FDRRC$,FDRK$,FDRP$,FINIT$
3      .MCALL READ$,OPEN$,CLOSE$,WAIT$,OPENW
4      .MCALL NMBLK$,FJAT$,FSRGSZ$,GET$,FDMF$
5      .MCALL EXIT$,PUT$,RIOW$,ALUN$,GETR
6
7      ;*****
8
9      ;PROCEDURY OPERACJI NA PLIKACH
10     ;
11     ;*****
12
13     ;
14     ;AUTOR KAROL NAJAR
15     ;
16     ;AUGUST 1983
17     ;
18     ;MERA - PIAF
19     ;
20     ;
21     ;*****
22
23     ;ZAWIERA :
24     ;      1/ INICJACJE BLOKU PARAMETROW
25     ;      FDB DLA PODANYCH PLIKOW
26     ;      2/ OTWARCIE PLIKU DO CZYTANIA
27     ;      REKORDAMI
28     ;      3/ OTWARCIE PLIKU DO PISANIA
29     ;      REKORDAMI
30     ;      4/ CZYTANIE PLIKU
31     ;      5/ ZAPISYWANIE PLIKU
32     ;      6/ ZAMKNIECIE SPECYFIKOWANYCH
33     ;      PLIKOW
34
35     ;*****
36 000000 000000 FIRST: .WORD 0
37 000002 000000 000000 BLCNT: .WORD 0,0
38
39
40 000000 .PSECT FCS
41
42 000000 010446 READ:: MOV R4,-(SP)   ;R4 NA STOS
43 000002 010546 MOV R5,-(SP)   ;R5 NA STOS
44 000004 012700 000546 MOV #FDB:TN,R0  ;FDB ADRES DO ODNIESIEN
45 000010 122760 177766 000052 CNPR # -10.,F.ERR(R0)  ;CZY EOF
46 000016 001404 BEQ 1$
47 000020 105760 000052 TSTB F.ERR(R0)  ;CZY HLAD
48 000024 100424 BMI RCL0  ;POWROT 7 HLEDEN
49 000026 000405 BR 2$
50
51 ***** EOF *****
52 000030 012761 177766 177772 1$: NOV # -10.,-6(R1)  ;SET EOF STATUS
53 000036 000167 000076 JMP S10  ;RETURN
54
55 *****
56 000042 2$: GET$ #FDB:IN,R1,#60.
57 000064 016061 000024 177776 NOV F.NRBD(R0),-2(R1)  ;LICZNIK

```

```

58 000072 000167 000042      JMP STO      ;RETURN
59                               ;
60 000076                    RCL0:
61 000076 022626             CMP (SP)+,(SP): ;STOS
62 000100 005726             TST (SP)+      ;
63 000102 116000 000052     MOV F.ERR(R0),R0;COD BLEDU
64 000106 010067 001256     MOV R0,ERC     ;PRZEKAZ DO KOMUNIKATU
65 000112 016705 001254     MOV LINK,R5    ;
66 000116                   CLOSE# #FDBIN ;WEJSCIOUY
67 000126                   CLOSE# #FDBOU ;WYJSCIOUY
68 000136 000511           BR ERCL0      ;
69                               ;
70                               ;***** STO *****
71 000140 012605           STO:  MOV (SP)+,R5
72 000142 012604           MOV (SP)+,R4    ;TWORZENIE REJESTROU
73 000144 000207           RETURN
74                               ;***** OPENE *****
75                               ;
76 000146           OPENE:: FINI# ;OTWARCIE PLIKU WEJSCIOWEGO I WYJSCIOWEGO
77 000152           OPEN#R #FDBIN
78 000170           OPEN#W #FDBOU
79 000206 000207           RETURN
80                               ;***** CLOSE *****
81                               ;
82 000210           CLOSE::  CLOSE# #FDBIN
83 000210           CLOSE# #FDBOU
84 000220           EXIT#S    ;
85 000230           ;***** INIT *****
86                               ;
87 000236 005725           INIT:: TST (R5)+      ;TWORZENIE LINKU
88 000240 012700 001244     MOV #J5IN,R0    ;ADRES BLOKU PARAMETROW
89 000244 012701 000014     MOV #12.,R1    ;LJCZNIK
90                               ;
91 000250 012520           1#:  MOV (R5)+,(R0)+ ;PRZEPISZ
92 000252 077102           SOB R1,1#      ;PETLA
93 000254 012567 001110     MOV (R5)+,ERC  ;ADRES CODU BLEDU
94 000260 010567 001106     MOV R5,LINK    ;
95 000264 000207           RETURN
96                               ;
97                               ;***** WRITE *****
98 000266 010446           WRITE:: MOV R4,-(SP)   ;OCHRONA REJESTROU
99 000270 010546           MOV R5,-(SP)   ;
100 000272           PUT# #FDBOU,#R4,R4 ;
101 000314 105767 000440     TST F.ERR+FDBOU;CZY BLAD
102 000320 100307           BPL STO      ;
103 000322 116700 000432     MOV F.ERR+FDBOU,R0 ;PRZYGIJ COD BLEDU
104 000326 010067 001036     MOV R0,ERC     ;PRZEKAZ COD BLEDU
105 000332 062706 000016     ADD #16,SP     ;STOS
106 000336 016705 001030     MOV LINK,R5    ;
107 000342           CLOSE# #FDBIN ;
108 000352           CLOSE# #FDBOU ;
109                               ;***** ERCL0 *****
110 000362 005467 001002     - ERCL0: NEG ERC     ;KOD BLEDU
111 000366           OPEN#R #FDBIN ; OTWARCIE PLIKU
112 000404 016705 000760     MOV ERC,R5     ;NR DRUKU
113 000410           1#:  GET#R #FDBIN,#R5,#64.,R5
114 000440           ALUN#S #1,#11,#0 ;

```

```

115 000460 QIUW$S #IO.WVR,#1,#1,,,(&BUMS,#24.)
116 000530 CLOSE$ #FDIRS ;
117 000540 EXIT$S ;
118 ;*****
119 ; BLOK FDB
120 ;*****
121 ;
122 000546 FRSZ$ 3,,FCS ;TRZY ZDJORY TYPU RECORD
123 ;POWRON DO P-SEKCJI "FCS"
124 000546 FDBIN: FDBDF$ ;DEFINICJA BLOKU DYREKTYWY
125 000706 FDF$A ;BLOK DLA PUT$ I GET$
126 000706 FDATA$ R.VAR,,,80. ;REKORD ZMIENNEJ DLUGOSCJ
127 ;NAJWIEKSZY CZYTANY PLIK MOZE HIEC 80. BLOKOW !
128 000706 FDRCA$ ;DEF. PLIKU REDROWEGO PAR. TU LUB W PUT/GET
129 000706 FDOP$A 2,DSIN,DEFI,FO.RD
130 ;INFORMACJE O PLIKU :
131 ;1) LUN
132 ;2) DYNAMICZNY BLOK NAZWY PLIKU
133 ;3) STATYCZNY " " "
134 ;4) DOSTEP DO PLIKU
135 000706 FDBOU: FDBDF$
136 001046 FDRCA$ ;
137 001046 FDF$A ;
138 001046 FDATA$ R.VAR,,,5,-2 ;ALOKOWANE 5 BLOKI
139 ;MOZLIWOSC ROZSZEZEN PO 1 BLOKU
140 ;BLOKI NIE TWORZA STRUKTURY CIAGLEJ
141 001046 FDOP$A 3,DSOU,DEFU,FO.APT
142 ;
143 001046 FDBMS: FDRDF$ ;
144 001206 FDF$A ;
145 001206 FDATA$ R.FIX,,,64,,,29.
146 001206 FDRCA$ FJ.RAN ;
147 001206 FDOP$A 4,DEMS,,FO.RD
148 001206 DEFI: INBLK$ INTEL,INT,,SY,0
149 ;STATYCZNY BLOK NAZWY PLIKU
150 ;
151 001244 DSIR: .BLKW 6. ;DYNAMICZNY BLOK NAZWY PLIKU
152 001260 DSOU: .BLKW 6. ; " " "
153 001274 DEFU: INBLK$ INTEL,INT,,SY,0
154 001332 000003 001346* DEMS: .WORD DEVL,DEV ;DEVICE NAME
155 001336 000005 001351* .WORD DIRL,DIR ;DIREKTORY
156 001342 000012 001356* .WORD NAHL,NAH ;FILNAME
157 001346 114 102 072 DEV: .ASCII /LB:/ ;
158 000003 DEVL=-DEV ;
159 001351 133 061 054 DIR: .ASCII /L1,21/ ;
160 001354 062 135 DIRL=.DIR
161 001356 121 111 117 NAH: .ASCII /OIOSYH.MSG/
162 001361 123 131 115
163 001364 056 115 123
164 001367 107 NAHL=-NAH ;
165 000012 .EVEN ;
166 001370 000000 ERC: .WORD 0
167 001372 000000 LINK: .WORD 0
168 001374 012 015 BUMS: .BYTE 12,15
169 001376 040 103 117 .ASCII /CON -- /

```

001401	116	040	055	
001404	055	040		
168 001406				BFHS: .BLK: 64.
169				?
170	000001			.END

STRONA 44
 STRON 46
 NEREJ. 5082

FILEOPERATION MACRO H1113 05-SEP-83 08:47 PAGE 1-4
 SYMBOL TABLE

BFHS	001406R	002	IO.UR1= ***** GX	F.FRAN= 000110	F.UEFH= 000020	STU	000140R	002
BLCH	000002R		F.AC1L= 000076	F.FNR = 000102	F.VRH = 000064	S.HFH= 000020		
BUR0	= ***** GX		F.AL0C= 000040	F.FTYP= 000116	F.VRSZ= 000060	S.FAL= 000016		
BURS	001374R	002	F.PHFS= 000062	F.FVER= 000120	INIT	00023ARG	002	S.LDH = 000140
CLOSE	000210RG	002	F.PH= 000070	F.HID= 000004	IO.PVH= ***** GX	S.FNAM= 000006		
DEF1	001206R	002	F.HHC= 000057	F.LIH = 000042	LINK	001372R	002	S.FNB = 000026
DEFU	001274R	002	F.HKHN= 000076	F.HHC1= 000054	NAM	001354R	002	S.FNAM= 000017
DEHS	001332R	002	F.HKHS= 000020	F.HBC1= 000055	NAM1	= 000012		S.FITY= 000004
DEV	001346R	002	F.HKEF= 000050	F.HRFG= 000056	N.DID = 000024			S.FIYP= 000002
DEVL	= 000003		F.BKP1= 000051	F.HRFD= 000024	N.DVNH= 000032			S.HFEH= 000020
DIR	001351R	002	F.HKST= 000024	F.HREC= 000030	N.FID = 000000	WRITE	000266RG	002
DURL	= 000005		F.BKVB= 000064	F.OVBS= 000020	N.FNAM= 000006	\$\$\$ARG= 000002		
DSIN	001244R	002	F.CHR = 000075	F.RACC= 000016	N.FIYP= 000014	.CLOSE= ***** G		
DSOU	001260R	002	F.CHTG= 000034	F.RAIT= 000001	N.FVER= 000016	.FINI= ***** G		
ERC	001370R	002	F.HFNB= 000046	F.RCNH= 000034	N.NEXT= 000022	.FSRCH= ***** G		
ERCLO	000362R	002	F.DSP1= 000044	F.RCTL= 000017	N.STAT= 000020	.GET = ***** G		
FDBIN	000546R	002	F.DVNH= 000134	F.RSIZ= 000002	N.UNIT= 000034	.OPEN = ***** G		
FDEMS	001046R	002	F.EFBK= 000010	F.RTYP= 000000	OPENE	000146RG	002	.PUT = ***** G
FDEOU	000706R	002	F.EFN = 000050	F.SEGN= 000100	PAR\$\$\$= 000027	...PC1= 001046R		002
FD.RAN=	***** GX		F.EOBB= 000032	F.SPTV= 000072	RCLD	000074R	002	...PC2= 001330R
FIRST	000000R		F.ERR = 000052	F.SPLN= 000074	READ	000000RG	002	...PC3= 001046R
FU.APD=	***** GX		F.FACC= 000043	F.STBK= 000036	R.FIX = ***** GX	...TPC= 000140		002
FU.RD =	***** GX		F.FFBY= 000014	F.UNIT= 000136	R.VAR = ***** GX			

. ABS. 000000 000
 000006 001
 FCS 001506 002
 \$\$\$FSK1 003020 003
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 4807 WORDS (19 PAGES)
 DYNAMIC MEMORY: 6254 WORDS (24 PAGES)
 ELAPSED TIME: 00:01:16
 ,FAIL0P=FILEP

TASK NAME : ...CON
 PARTITION NAME : GEN
 IDENTIFICATION : 0735
 TASK UIC : [7,202]
 STACK LIMITS: 000172 001171 001000 00512.
 PRG XFR ADDRESS: 004264
 TOTAL ADDRESS WINDOWS: 1.
 TASK IMAGE SIZE : 7424. WORDS
 TASK ADDRESS LIMITS: 000000 034737
 R-W DISK BLK LIMITS: 000002 000036 000035 00029.

*** ROOT SEGMENT: HEAN

R/W MEM LIMITS: 000000 034737 034740 14816.
 DISK BLK LIMITS: 000002 000036 000035 00029.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK. : (RW,I,LCL,REL,CON)	001172 001364 00756.		
	001172 000744 00484. MANAGE		HEAN.OBJ?10
	002136 000412 00266. FORBLO		FORBLO.OBJ?2
	002550 000006 00006. FILEOP		FILEOP.OBJ?2
FCS : (RW,I,LCL,REL,CON)	002556 001506 00838.		
	002556 001506 00838. FILEOP		FILEOP.OBJ?2
NAME : (RW,I,LCL,REL,CON)	004264 001124 00596.		
	004264 001124 00596. SKIP		NAME.OBJ?2
PASRUN: (RW,I,LCL,REL,CON)	005410 004342 02274.		
REANAM: (RW,I,LCL,REL,CON)	011752 002470 01336.		
	011752 002470 01336. SKIP		NAME.OBJ?2
SKIP : (RW,I,LCL,REL,CON)	014442 000166 00118.		
	014442 000166 00118. SKIP		NAME.OBJ?2
##FSR1: (RW,D,GBL,REL,UVR)	014630 003060 01584.		
	014630 003060 01584. FILEOP		FILEOP.OBJ?2
	014630 000000 00000. SKIP		NAME.OBJ?2
##FSR2: (RW,D,GBL,REL,CON)	017710 000104 00068.		
##RESL: (RW,I,LCL,REL,CON)	020014 013176 05758.		
999999: (RW,I,LCL,REL,CON)	033212 001526 00854.		
	033212 001526 00854. SKIP		NAME.OBJ?2

GLOBAL SYMBOLS:

BUHO 002434-R FORBLO 002136-R MANAGE 001172-R READ 002556-R SKIP 014442-R ##HEAP 033212-R
 CLOSE 002766-R INIT 003014-R OPENE 002724-R REANAM 011752-R UR11E 003044-R

HUX.1SK11 MEMORY ALLOCATION MAP TKB
MEAN 5-SEP-83 09:06

PAGE 2

STRONA 46
STRON 46
N=REF. 5081

*** TASK BUILDER STATISTICS:

TOTAL WORK FILE REFERENCES: 15116.
WORK FILE READS: 0.
WORK FILE WRITES: 0.
SIZE OF CORE POOL: 6280. WORDS (24. PAGES)
SIZE OF WORK FILE: 3072. WORDS (12. PAGES)

ELAPSED TIME: 00:01:22

R W iA@Z / D P 7H J'o H ^ vQI H z / uR v s: 3H