

Klicova slova jazyka BASIC-G pro CONSUL 2717:

ABS - absolutni hodnota promenne; PRINT ABS(-2) ... =2
ADR - adresa umistení promenne v pameti;
PRINT ADR(X);ADR(DD) ... =9255 9249
AND - logicky soucin binarni; 14 AND 7=0;
APEEK - obsah dvou po sobe jdoucich bytu v pameti;
PRINT APEEK(000) ... =32708
ASC - dekadicka hodnota 1.znaku retezce;
X\$="Abc";PRINT ASC(X\$) ... =65
AT - urcuje radek(0-24) a sloupec(0-47) v PRINT AT;
PRINT AT 4,10;AT 8,10;AT 8,20;AT 8,20;VI
ATN - autangens cisla; ATN(1)=.785398;
AUTO - automaticke cislovani radku pri psani programu;
AUTO200,3 - od radku 200 s krokem 3: 203,206, ...
AXIS - vykresleni os v pruseciku souradnic;
AXIS0,0:AXIS255,242 - vykresli ramecek;
BEEP - zvukovy signal (pipnutí);
BIT - hodnota bitu B(0-7) promenne X(0-255);
X=5:BIT X,0:1:BIT X,1:0:BIT X,2:1: ...
BMOVE - presun pera na sloupec S(0-47) a radek R(0-242);
BMOVE S,R: ... BMOVE 0,0 - pozice vlevo nahore;
BPLOT - vykresleni obrazce zadaného retezce bytu;
T\$=CHR\$(A)+CHR\$(B)+CHR\$(C)+CHR\$(D)+ ...
BPLOT T\$,2 - vykresli A\*B, pod nimi C\*D, atd.
CLEAR - vymulovani vseh promennych v programu;
CODE - jako retezec sestnactkovech bytu,konci C9(=RET);
P\$="3E41C9324C1C0085C9";CODE P\$ - vypise A;
CONTROL - nastaveni vstup/vystupního kanalu K(0-7), jeho
registru R(0-x)/vystupními daty: CONTROL,4,3,132,5
nastavi: kanal 4, port B - vystup, bit PC2=1
-pokracovani programu zastaveného povelem STOP;
COS - hodnota kosinu uhlu v radiánech/stupních;
RAD: COS(3.14) ... =-1; DEG: COS(180) ... =-1;
DATA - seznam dat pro cteni povelem READ;
DEF FNC - definice uzivatelske funkce;
DEF FNC A(X)=(X-2):Y=FNC A(2) ... =4
DEG - uhly pro goniometrické funkce jsou ve stupních;
DIM - vymizení ciselného/retezceho pole v pameti;
DIM A(2,3)-dvourozmerne pole o 3\*4 prvcích;
DIM A\$(9)-jednorozmerne pole o 10 prvcích;
DISP - obrazeni vyrazu/textu v dialogovem radku;
DISP 2\*5:55: DISP K\$: DISP TAB(10)/A/B
DLOAD - cteni dat z magnetofonu do dimenzovaného pole;
DIM A\$(0):DLOAD 0;A\$(0) -soubor 01 do A\$(0);
DSAVE - zaznam souboru dat na magnetofon z daného pole;
DSAVE 02;CU(2) -vysledky ... Nazev souboru;
END - ukončení programu;
ENTER - vstup dat z kanalu a registru KR do seznamu S;
ENTER 401;X - vstup dat do X z reg.01 4.Kanalu;
EXP - mocnina zakladu prirodzenych logaritmu;
E=2.71828 ... EXP(2)-E=2=7.38906
FILL - vyplnení oblasti o sirce(0-255) a vysce(0-242)
binarnim profilem N(0-63): FILL S,V;M
FILL 10,20:5 -profil 0101 sirky 10 a vysky 20;
FNC - vyvolani definovane uzivatelske funkce; Y=FNC A(X)
FRE - velikost volne pameti pro program nebo retezce
promenne: FRE(X)(<=)14630, FRE(X)(<=)3841;
FOR...TO - zahlaví program.cyklu; provede opakovane
povele mezi FOR a NEXT, zvsujsje promennou o STEP
FOR I=0:109:PRINT I:NEXT I-vypise hodnoty promenne;
neni-li uveden STEP, zvsujsje se promenna o +1;

PEN - ridi mod vykreslovani grafiky; PEN 1-polojka;
2-blikani,4-inverse,8-negovani,16-mazani
PEEK - zjistí hodnotu bytu na adrese pameti;
B=PEEK(A):PEEK(8000)=C3
PAUSE - casova prodleva v programu v nasobcich 0.1sec;
PAUSE 20 ... 2 sec: PAUSE0 ... 25.5 sec
PLOT - vykresli spojnicí do bodu daného souradnicemi;
z predchazejici pozice pera, nebo pouze bod (P=1);
PLOT X,Y; PLOT 20,20;50,50;90,10;70,20,1
POKE - zapise na adresu pameti byte (0-255);
POKE 2E,00 - zmeni vypis na 80 znaku/radek;
PRINT - vypis vyrazu ze seznamu na obrazovku;
PRINT I;2;3 -vypise cisla vedle sebe s mezerou;
PRINT I,2,3 -vypise cisla s tabelacii:1,15,29
RAD - uhly dalsich goniom. funkci jsou v radiánech;
READ - cteni polozek na radku DATA do seznamu v READ;
DATA 1,a:READ X,10 ... X=1,10;"a"
REM - komentár v programu; povele za REM jsou neucinné;
RETURN -nastavi ukazatel na radek DATA urceny pro READ
RIGHT -povel retezce z programu volaneho GOSUB;
ABS="abcde";RIGHT(A\$,2)='de'
ROM - presun programu z pameti typu ROM do pameti RAM
na adresu /7000 a jeho spuštění: ROM 0
RND -generátor pseudonahodného cisla z intervalu(0-1);
RUN -spusti program od zacatku nebo daného radku: RUN90
SAVE -zapis programu daného cisla (<99) na mg.kazetu;
SAVE 99,NAZEY...nazev muze mit 0-8 znaku;
SCALE -nastavi meritko souradnic pro grafické prikazy;
SCALE0,255,0,242 nebo: SCALE -1,1,-10,10
SGN -hodnota znaménka vyrazu z zavorce;
SGN(-5)=-1:SGN(0/2)=0:SGN(3\*4)=1;
SIN -hodnota funkce sinus uhlu v radiánech/stupních;
DEG:SIN(90)=1:RAD:SIN(3.14159/2)=1
SOR -druha odmocnina z vyrazu; SOR(500+29)=23
SPC -vlozeni uvedeneho poctu mezer do textu;
PRINT 1;SPC(5);2;SPC(5);3; ...
STATUS-nacte z kanalu(0-7) hodnotu registru(0-X)
S-STATUS 4,X ... X(0-3) registru portu 8255;
STOP -cast zahlaví programového cyklu FOR...TO \_STEP\_
\*\* Stop at line 95 \*\* ...Zastaveni na radku 95;
STRU -premeni cislo na retezec znaku;
X=-1.234:STR\$(X)='-1.234'
TAB -odsok na zadanou pozici v radku textu;
PRINT TAB(8);a: DISP TAB(10);10
TAN -hodnota tangens uhlu v radiánech/stupních;
RAD:TAN(3.14159/4)=1:DEG:TAN(45)=1
THEN -soucast podminkového povelu IF...THEN
TO -soucast zahlaví programového cyklu FOR...TO
USR -volani uzivatelského podprogramu ve strukturním
X=USR(7000) ... v promenne je vložena strážná
VAL -premeni retezec cisel na cislo; opak strn;
X=8:VAL(STR\$(7))=8
WAIT -zastavení programu az do splnení podmínky cteni
hodnoty z portu a bity danými maskou R(0-255)
a neekvivalenci se zadanou hodnotou X(0-255);
WAIT P,M,K ... WAIT 4,15,11
? -zkraceny prikaz vypisu zpravy do dialog. radku;
program pokrácuje po stlacení lib.klavesy;
; -symbol retezceve promenne (dolar).

GCLFAR -vymazani stinitka obrazovky;
GOSUB -volani podprogramu na radku cislo/vyraz;
GOSUB 120 nebo: V=5\*X:GOSUB V
GOTO -podmineny skok na radek cislo/vyraz;
GOTO 240 nebo: R=10:GOTO R
HEX\$ -prevod desitkového vyrazu na sestnactkove cislo;
HEX\$(256) ... =100; HEX\$(1) ... =FFFF
CHECK -overeni kontrolního souctu nahraného programu
prikazem SAVE: CHECK 02 ... 02-cislo nahravky
CHR\$ -prevod desitkového vyrazu (0-255) na znak ASCII;
CHR\$(65) ... =A, nezname znaky se zobrazí jako
IF...THEN -povel podmíny splnením vyrazu za IF; není-li
vyraz splnen pokracuje program na dalsim radku;
IF 0=0 OR 0>7 THEN PRINT 0:GOTO350
INK -ridi mod vypisu v PRINT: INK(1)-polojas,
INK(2)-blikani,INK(4)-inverse,INK(5)-inv.tpolojas
INKEY -nactení hodnoty (0-11) klavesy klíče F0-F11;
INP -přirazení bytu od portu (0-255); INP(7);
INPUT -cte znaky z klavesnice do seznamu promennych;
INPUT A,ZH; ...-kontroluje syntax u cis.promenne;
INT -zaokrouhli dolu desetinné cislo; INT(-2.0)=-3;
LABEL -rozkresli vyrazu v nasobcich sirky S(1-51)
a vysky V(1-34): LABELS,V;T\$:LABEL2,4,9;X;'E'
LEFT\$ -omezí retezec na urceny počet znaku zleva;
R\$="abcde";LEFT\$(R\$,2)='ab'
LEN -urceni delky (poctu znaku) retezceve promenne;
X\$="qwert";LEN(X\$)=5
LEFT -nepovinný prikaz přirazení hodnoty promenne;
LIST -vypis programu na obrazovku od zacatku nebo
od urcenoého radku: LIST 130
LIST# -vypis programu na tiskarnu pripojenu na kanal
K(0-7) a jeho registru R(0-X): LIST#404;130
LLIST -vypis programového radku do dialogového radku
k editaci; LLIST -od zacatku; LLIST 120 -od radku;
LOAD -nactení programu uvedeneho cisla z magnetofonu;
LOAD 05 -jine nahravky necte,zobrazí jejich nazvy;
LOG -přirozeny logaritmus kladného vyrazu;
LOG(1+1.71829)=1
MID\$ -omezí retezec od urcene pozice na urceny
pocet znaku. X\$="12345";MID\$(X\$,3,2)='34'
MOVE -presun grafického pera do bodu X,Y urcenoého
merikem: SCALE Xmin,Xmax,Ymin,Ymax
NEXT -ukončení programového cyklu,zacinajícího FOR...TO;
NEW -zrusení programu i jeho promennych v pameti;
NOT -logicka negace vyrazu (dvojkovy doplněk);
A=5:NOT A=-6:NOT (A-1)=-A
NULL -vlození (0-7) prázdných znaku po kazdem
radku v pameti programu: NULL(8)
ON ERR-potlaci nejbliži vypis chyby a program
pokracuje uvedeným prikazem: ON ERR GOTO20;
ON GOSUB -programovy prepínac podle hodnoty promenne;
ON N=5:ON N GOSUB10,20,30,40,50
ON\_GOTO -programovy prepínac skoku podle promenne;
seznam radku skoku musí mít N hodnot;
je-li N mimo rozsah,provede se další radek;
OR -logicky binární součet vyrazu;
7 OR 14=15; 2 OR 7=7
OUT -vystup promenne B(0-255);
OUT I,B: OUT 4,132
OUTPUT -vystup dat ze seznamu S na kanal K(0-7)
a registr R(0-X): OUTPUT 404,9,A,10

Znaky v kodu ASCII (ISO-7):

00 NUL	20	40 @	60 `
01 SOH	21 !	41 A	61 a
02 STX	22 "	42 B	62 b
03 ETX	23 #	43 C	63 c
04 EOT	24 \$	44 D	64 d
05 ENG	25 %	45 E	65 e
06 ACK	26 &	46 F	66 f
07 BEL	27 '	47 G	67 g
08 BS	28 (	48 H	68 h
09 HT	29 )	49 I	69 i
0A LF	2A *	4A J	6A j
0B VT	2B +	4B K	6B k
0C FF	2C ,	4C L	6C l
0D CR	2D -	4D M	6D m
0E SO	2E .	4E N	6E n
0F SI	2F /	4F O	6F o
10 DLE	30 0	50 P	70 p
11 DC1(X-ON)	31 1	51 Q	71 q
12 DC2	32 2	52 R	72 r
13 DC3	33 3	53 S	73 s
14 DC4	34 4	54 T	74 t
15 NAK	35 5	55 U	75 u
16 SYN	36 6	56 V	76 v
17 ETB	37 7	57 W	77 w
18 CAN	38 8	58 X	78 x
19 EM	39 9	59 Y	79 y
1A SUB	3A :	5A Z	7A z
1B ESC	3B ;	5B [	7B {
1C FS	3C <	5C \	7C
1D GS	3D =	5D ]	7D }
1E RS	3E >	5E ^	7E ~
1F US	3F ?	5F _	7F `

Vyznam nekterych ridicich znaku:

- NUL -prazdny znak;
- SOH (Start Of Header)-zahlaví dat;
- STX (Start of Text)-zacatek textu;
- ETX (End of Text)-konec textu;
- ENG (ENquiry)-pozadavek spojeni;
- ACK (ACKnowledge)-potvrzeni zpravy;
- BEL -zvukovy signal;
- LF (Line Feed)-novy radek;
- CR (Carriage Return)-navrat voziku;
- NAK (No Acknowledge)-nepotvrzeni;

Prvovodni tabulka cisel desitkove/dvojkove/sestnactkove:

0 = 00000000 = 00	0 = 00000000 = 00
1 = 00000001 = 01	16 = 00010000 = 10
2 = 00000010 = 02	32 = 00100000 = 20
3 = 00000011 = 03	48 = 00110000 = 30
4 = 00000100 = 04	64 = 01000000 = 40
5 = 00000101 = 05	80 = 01010000 = 50
6 = 00000110 = 06	96 = 01100000 = 60
7 = 00000111 = 07	112 = 01110000 = 70
8 = 00001000 = 08	128 = 10000000 = 80
9 = 00001001 = 09	144 = 10010000 = 90
10 = 00001010 = 0A	160 = 10100000 = A0
11 = 00001011 = 0B	176 = 10110000 = B0
12 = 00001100 = 0C	192 = 11000000 = C0
13 = 00001101 = 0D	208 = 11010000 = D0
14 = 00001110 = 0E	224 = 11100000 = E0
15 = 00001111 = 0F	240 = 11110000 = F0

Chybova hlaseeni BASIC-G:

- Arr Alleg -chybne nebo dvakrat dimenzovane pole;
- Can't cont -nespravne pouziti prikaz CONT;
- Data exhan -chybejici radek DATA pro povel READ;
- Dv by zero -pokus o deleni nulou;
- Field lost -prekrocen licet vstupnich dat v INPUT;
- File bound -prekrocen maximalni pocet zaznamu;
- File error -chyba pri nahravani (premosu) dat;
- File small -data z mpp neodpovidaji dimenzovanemu poli;
- Fnc param -nespravny parametr funkce;
- Input err. -chyba pri vstupu dat (znak misto cislice);
- No for stm -chybne zadana program.smycka FOR\_TO\_NEXT;
- No str.spc -preplnena pamet pro retezcove promenne;
- Numb.nonex -neexistujici program.radek v GOTO;GOSUB;
- Only in pg -povel lze pouzít jen v programovem modu;
- Overflow -prekrocen rozsah vypoctu (vysledek >10^38);
- Pg too big -neni jiz volna pamet pro program;
- Return err -pouzit RETURN bez volani GOSUB;
- Stop at ln -povel STOP na radku ln zastavil program;
- Str algrt. -vyraz v reteci je slozity; rozdelit;
- Str.long -delka retece presahuje 255 znaku;
- Subscr.rng -index promenne mimo rozsah urceny DIM;
- Syntax err -chybne zadany povel (preklep);
- Type conv -chybny prevod cisla na retezec a opacne.

Povely pro praci v rezimu monitor:

- DUMP -vypis obsahu pameti na obrazovku; DUMP 8427 8427 4F 73 20 72 65 61 64 79 0s ready (lze pozastavit pomoci SHIFT a zastavit STOP);
- JOB -prenos programu z adresy AROM v modulu ROM do RAM na adresu APOC v delce LONG; JOB AROMLONGAPOC JOB 000025000000...zavede interpret BASIC-G z ROM;
- JUMP -odstartovani programu na zadane adrese; Napr.: JUMP 0000...odstartuje interpret BASIC-G;
- MEM -vypis 16 bytu z pameti od zadane adresy do dialog.radeku v sestnactkovem tvaru; MEM C030
- MCSV -zapis programu z pameti na magnetofon pod uvedenym cislem(99); od adresy po adresu; s nazvem (0-8 znaku) MCSV 15,0000-2500,BASIC G
- MCEND kontrola nahraného programu na magnetofonu; je-li spravny kontrolni soucet bytu zaznamu,vypise: OK
- MGLD -nacteni programu uvedeného cisla z magnetofonu do pameti pocitace; MGLD 15 (jina cisla nenahrava);
- SUB -SUBSTITUTE-zmena obsahu pameti na uvedene adrese;
- ↑RST-prechod do rezimu BASIC-G pres RESTART;
- ↑DEL-odstartuje program v pameti od adresy 0000; napr. pokud byl jiz v pameti BASIC-G, spusti jej.

Hlaseeni monitoru:

- Error in adres -chyba v adrese (A-mistna;sestnactkova);
- Error in data -chyba v datech SUB (sestnactkova);
- Executive -oznamuje spuštění programu ve stroj.kodu;
- File error -cteny soubor je chybný nebo nebyl nahrazen v rezimu MONITOR;
- Memory overflow-preplnena pamet klicu F0-F11 ('7000-'7F00;
- No command -pouzit povel neznamy monitoru;
- No key -stisknutý klic nema zadan obsah;
- Mg stop ! -ukonceno nahravani programu.

Vyznam ridicich a editacnich klaves:

- C-D -zobrazeni posledni zpravy pocitace;
- CLR -vymazani celého dialogoveho radku;
- DEL -vymazani znaku na miste kursoru a posunutí zbylého textu o znak vlevo;
- END -posun kursoru za posledni znak textu;
- EOL -odeslání textu ke zpracovani pocitace (provedeni prikazu nebo zazaceni radku);
- F0-F11-programove kliese;po predchozim stlaceni WRK se do nich uchova text dialogoveho radku;
- INS -na miste kursoru vytvori mezeru pro vlozeni dalšio znaku;
- RCL -vypolani posledního textu odeslaného EOL;
- STOP -přeruší provedeni programu po ukonceni vsech povelu programoveho radku; pokračovat lze povelam CONT (a EOL);
- WRK -umožňuje zapis do programovych klicu F0-F11; ← → -posun kursoru o jeden znak vlevo/vpravo; ↵ -navratí kursor na zacatek dialogoveho radku; ← → -posun dialogoveho radku o 1 znak vuci kursoru; -prnykac (shift) umožňuje psani hornich znaku klaves a malých písmen; po dobu stlaceni pozastavuje spuštění programu; ↑C-D -prechod do rezimu terminalu; ↑CLR -vymaze text od kursoru vpravo; ↑DEL -odstartuje program od adresy 0000; byl-li tam jiz interpret BASIC-G, tak jej spusti; ↑END -vypne/zapne zvukovou signalizaci (pipani); ↑F0-F11-programove kliese 'horni' pro uchovani textu; ↑PRINT-vypnuti/zapnuti opisu dialogoveho radku do pracovni casti stinitka; ↑RCL -prepnuti do rezimu MONITOR (navrat ↑DEL); ↑RST -reset systemu; nouzova funkce pro pripad havarie programu; spusti program od adr.0000; ↑WRK -vypolá znak nasledujici v tabulce ASCII za znaka, na kterem stojí kursor; ↑↑ -vymaze obrazovku; ↑↑K -prepne zobrazeni na inverzni (cerne na bilem); ↑↑ -prepne dialogovy radek na polojas (blikani).

Rozkresleni znaku na hornim radku:

Adr:	Bity	Obsah adresy:
HEX	012345	HEX DEK
'C000		'1F = 31
'D940		'01 = 1
'D980		'01 = 1
'C0C0		'07 = 7
'C100		'01 = 1
'C140		'01 = 1
'C180		'01 = 1
'C1C0		'00 = 0
'C200		'00 = 0

Instrukce strojového kodu MHB 2080:  
(d-data 8b, w-data 16b, a-adresa 16b, a8-adresa 8b)

00 NOP	40 MOV B,B	80 ADD B	C0 RNZ
01 LXI B,w	41 MOV B,C	81 ADD C	C1 POP B
02 STAX B	42 MOV B,D	82 ADD D	C2 JMP a
03 INX B	43 MOV B,E	83 ADD E	C3 JMP a
04 INR B	44 MOV B,H	84 ADD H	C4 CNZ a
05 DCR B	45 MOV B,L	85 ADD L	C5 PUSH B
06 MVI B,d	46 MOV B,M	86 ADD M	C6 ADI d
07 RLC	47 MOV B,A	87 ADD A	C7 RST 0
08 ...	48 MOV C,B	88 ADC B	C8 RZ
09 DAD B	49 MOV C,C	89 ADC C	C9 RET
0A LDAX B	4A MOV C,D	8A ADC D	CA JZ a
0B DCX B	4B MOV C,E	8B ADC E	CB ...
0C INR C	4C MOV C,H	8C ADC H	CC CZ a
0D DCR C	4D MOV C,L	8D ADC L	CD CALL a
0E MVI C,d	4E MOV C,M	8E ADC M	CE ACI d
0F RRC	4F MOV C,A	8F ADC A	CF RST 1
10 ...	50 MOV D,B	90 SUB B	D0 RNC
11 LXI D,w	51 MOV D,C	91 SUB C	D1 POP D
12 STAX D	52 MOV D,D	92 SUB D	D2 JMP a
13 INX D	53 MOV D,E	93 SUB E	D3 OUT a8
14 INR D	54 MOV D,H	94 SUB H	D4 CNC a
15 DCR D	55 MOV D,L	95 SUB L	D5 PUSH D
16 MVI D,d	56 MOV D,M	96 SUB M	D6 SUI d
17 RAL	57 MOV D,A	97 SUB A	D7 RST 2
18 ...	58 MOV E,B	98 SBB B	D8 RC
19 DAD D	59 MOV E,C	99 SBB C	D9 ...
1A LDAX D	5A MOV E,D	9A SBB D	DA JC a
1B DCX D	5B MOV E,E	9B SBB E	DB IN a8
1C INR E	5C MOV E,H	9C SBB H	DC CC a
1D DCR E	5D MOV E,L	9D SBB L	DD ...
1E MVI E,d	5E MOV E,M	9E SBB M	DE SBI d
1F RAR	5F MOV E,A	9F SBB A	DF RST 3
20 ...	60 MOV H,B	A0 ANA B	E0 RPO
21 LXI H,w	61 MOV H,C	A1 ANA C	E1 POP H
22 SHLD a	62 MOV H,D	A2 ANA D	E2 JPO a
23 INX H	63 MOV H,E	A3 ANA E	E3 XTHL
24 INR H	64 MOV H,H	A4 ANA H	E4 CPO a
25 DCR H	65 MOV H,L	A5 ANA L	E5 PUSH H
26 MVI H,d	66 MOV H,M	A6 ANA M	E6 ANI d
27 RAL	67 MOV H,A	A7 ANA A	E7 RST 4
28 ...	68 MOV L,B	A8 XRA B	E8 RPE
29 DAD H	69 MOV L,C	A9 XRA C	E9 PCHL
2A LHLD a	6A MOV L,D	AA XRA D	EA JPE a
2B DCX H	6B MOV L,E	AB XRA E	EB XCHG
2C INR L	6C MOV L,H	AC XRA H	EC CPE a
2D DCR L	6D MOV L,L	AD XRA L	ED ...
2E MVI L,d	6E MOV L,M	AE XRA M	EE XRI d
2F RAR	6F MOV L,A	AF XRA A	EF RST 5
30 ...	70 MOV M,B	B0 ORA B	F0 POP
31 LXI SP,w	71 MOV M,C	B1 ORA C	F1 POP PSM
32 STAX a	72 MOV M,D	B2 ORA D	F2 JPI a
33 INX SP	73 MOV M,E	B3 ORA E	F3 DI
34 INR M	74 MOV M,H	B4 ORA H	F4 CP a
35 DCR M	75 MOV M,L	B5 ORA L	F5 PUSH PSH
36 MVI M,d	76 HLT	B6 ORA M	F6 ORI d
37 STC	77 MOV M,A	B7 ORA A	F7 HS 6
38 ...	78 MOV A,B	B8 ORA B	F8 RM
39 DAD SP	79 MOV A,C	B9 ORA C	F9 SHL
3A LDA a	7A MOV A,D	BA ORA D	FA JM a
3B DCX SP	7B MOV A,E	BB ORA E	FB EI
3C INR A	7C MOV A,H	BC ORA H	FC CM a
3D DCR A	7D MOV A,L	BD ORA L	FD ...
3E MVI A,d	7E MOV A,M	BE ORA M	FE CPI d
3F CMC	7F MOV A,A	BF ORA A	FF RST 7

... instrukce není definována.

Naplnění registru konstantou d=8bitů:

06 MVI B,d 16 MVI D,d 26 MVI H,d 36 MVI M,d  
0E MVI C,d 1E MVI E,d 2E MVI L,d 3E MVI A,d

Naplnění reg\_paru konstantou w=16bitů:  
01 LXI B,w 11 LXI D,w 21 LXI H,w 31 LXI SP,w

Presuny mezi registry: 40 MOV B,B - 7F MOV A,A

Presuny mezi reg\_pary:  
E3 XTHL E9 PCHL EB XCHG F9 SPHL

Plnění/uložení strádace A a reg\_paru HL:  
0A LDAX B 1A LDAX D 2A LHLD a 3A LDA a  
02 STAX B 12 STAX D 22 SHLD a 32 STA a

Uložení reg\_paru do zásobníku/naplnění ze zásobníku:  
C5 PUSH B D5 PUSH D E5 PUSH H F5 PUSH PSM  
C1 POP B D1 POP D E1 POP H F1 POP PSM

Skoky podmíněné a nepodmíněné na adresu a:  
C2 JNZ a D2 JNC a E2 JPO a F2 JP a  
CA JZ a DA JC a EA JPE a FA JM a  
C3 JMP a (nepodmíněný)  
Volání podprogramu podmíněné i nepodmíněné:  
C4 CNZ a D4 CNC a E4 CPO a F4 CP a  
CC CZ a DC CC a EC CPE a FC CM a  
CD CALL a

Navraty z podprogramu podmíněné i nepodmíněné:  
C0 RNZ D0 RNC E0 RPO F0 RP  
C8 RZ D8 RC E8 RPE F8 RM  
C9 RET

Restarty (obdoba volání podprogramu) na adr.X\*8:  
C7 RST 0 D7 RST 2 E7 RST 4 F7 RST 6  
CF RST 1 DF RST 3 EF RST 5 FF RST 7

Zvětšení (INX)/zmenšení (DCX) obsahu reg\_paru o 1:  
03 INX B 13 INX D 23 INX H 33 INX SP  
0B DCX B 1B DCX D 2B DCX H 3B DCX SP

Zvětšení/zmenšení obsahu reg\_o 1 (meni F mimo CY):  
04 INR B 14 INR D 24 INR H 34 INR M  
0C INR C 1C INR E 2C INR L 3C INR A  
05 DCR B 15 DCR D 25 DCR H 35 DCR M  
0D DCR C 1D DCR E 2D DCR L 3D DCR A

Součet reg\_paru a IHLJ (výsledek v H,meni jen CY):  
09 DAD B 19 DAD D 29 DAD H 39 DAD SP  
Součet A s konstantou d nebo registrem (meni F):  
C6 ADI d 80 ADD B 81 ADD C... (viz tab.)  
CE ACI d 88 ADC B 89 ADC C... (přičte přenos CY)

Odečtení konstanty/registru od A (meni F):  
D6 SUI d 90 SUB B 91 SUB C... (viz tab.)  
DE SBI d 98 SBB B 99 SBB C... (pujci si 1 z CY)

Logické funkce konstanty/registru s A (meni F):  
E6 ANI d A0 ANA B A1 ANA C...  
EE XRI d A8 XRA B A9 XRA C...  
FE ORI d B0 ORA B B1 ORA C...  
FE CPI d B8 CMP B B9 CMP C...  
Rotace obsahu A/A+CY vlevo L/vpravo R:  
07 RLC 0F RRC 17 RAL 1F RAR

Nastavení přenosu CY: 3F STC  
Doplněk přenosu -CY: 3F CMC  
Doplněk strádace -A: 2F CMA  
Dekadické nastavení A: 27 DAA  
Prerušeni zakázáno: F3 DI  
Prerušeni povoleno: FB EI (po další instrukci)  
Výstup dat strádace: D3 OUT a8 (adresa 8bitu)  
Výstup dat do strádace: DB IN a8 (adresa 8bitu)  
Prázdná operace: 00 NOP (přičte +1 k PC)  
Zastavení procesoru: 76 HLT (start po RESET,INT)

Prevodni tabulka cisel desitkove/sestnactkove:

0	00	64	40	128	80	192	C0
1	01	65	41	129	81	193	C1
2	02	66	42	130	82	194	C2
3	03	67	43	131	83	195	C3
4	04	68	44	132	84	196	C4
5	05	69	45	133	85	197	C5
6	06	70	46	134	86	198	C6
7	07	71	47	135	87	199	C7
8	08	72	48	136	88	200	C8
9	09	73	49	137	89	201	C9
10	0A	74	4A	138	8A	202	CA
11	0B	75	4B	139	8B	203	CB
12	0C	76	4C	140	8C	204	CC
13	0D	77	4D	141	8D	205	CD
14	0E	78	4E	142	8E	206	CE
15	0F	79	4F	143	8F	207	CF
16	10	80	50	144	90	208	D0
17	11	81	51	145	91	209	D1
18	12	82	52	146	92	210	D2
19	13	83	53	147	93	211	D3
20	14	84	54	148	94	212	D4
21	15	85	55	149	95	213	D5
22	16	86	56	150	96	214	D6
23	17	87	57	151	97	215	D7
24	18	88	58	152	98	216	D8
25	19	89	59	153	99	217	D9
26	1A	90	5A	154	9A	218	DA
27	1B	91	5B	155	9B	219	DB
28	1C	92	5C	156	9C	220	DC
29	1D	93	5D	157	9D	221	DD
30	1E	94	5E	158	9E	222	DE
31	1F	95	5F	159	9F	223	DF
32	20	96	60	160	A0	224	E0
33	21	97	61	161	A1	225	E1
34	22	98	62	162	A2	226	E2
35	23	99	63	163	A3	227	E3
36	24	100	64	164	A4	228	E4
37	25	101	65	165	A5	229	E5
38	26	102	66	166	A6	230	E6
39	27	103	67	167	A7	231	E7
40	28	104	68	168	A8	232	E8
41	29	105	69	169	A9	233	E9
42	2A	106	6A	170	AA	234	EA
43	2B	107	6B	171	AB	235	EB
44	2C	108	6C	172	AC	236	EC
45	2D	109	6D	173	AD	237	ED
46	2E	110	6E	174	AE	238	EE
47	2F	111	6F	175	AF	239	EF
48	30	112	70	176	B0	240	F0
49	31	113	71	177	B1	241	F1
50	32	114	72	178	B2	242	F2
51	33	115	73	179	B3	243	F3
52	34	116	74	180	B4	244	F4
53	35	117	75	181	B5	245	F5
54	36	118	76	182	B6	246	F6
55	37	119	77	183	B7	247	F7
56	38	120	78	184	B8	248	F8
57	39	121	79	185	B9	249	F9
58	3A	122	7A	186	BA	250	FA
59	3B	123	7B	187	BB	251	FB
60	3C	124	7C	188	BC	252	FC
61	3D	125	7D	189	BD	253	FD
62	3E	126	7E	190	BE	254	FE
63	3F	127	7F	191	BF	255	FF



ASM (d.) jméno souboru (.plp2b3)  
 --- Absolutní překladář JSI. Překládá uloží ve formě HEX na označený disk. Protokol o překladu, uloží do určeného souboru nebo potlačí.  
 d: - označení diskové jednotky, má přednost před p1, p2 a p3  
 jméno souboru - pouze jméno bez přípony  
 p1 - určuje jednotku se zdrojovým programem jméno souboru, ASM  
 p2 - určuje umístění generovaného souboru jméno souboru, HEX  
 Z - potlačí generování HEXa kódu  
 p3 - určuje umístění výpisu jméno souboru, PRN  
 X - výpis na konzole  
 Z - potlačení výpisu

Zpráva o chybách při překladu  
 D - chyba dat; prvek nemůže být umístěn v oblasti  
 E - chyba ve výrazu, je chybně konstruován  
 L - chyba návěští, vícenásobná definice  
 N - příkaz není příkazem této verze překladáře  
 O - přeplnění, složitý výraz nelze vyhodnotit  
 P - návěští nemá v průchodech stejnou hodnotu  
 R - chyba registrové operace ( neexistuje )  
 S - chybná syntaxe, neznámý operand  
 V - chyba hodnoty, nesprávně zapsaný operand

Závažné chyby  
 NO SOURCE FILE PRESENT soubor ASM neexistuje  
 NO DIRECTOR SPACE adresář disku je zaplněn  
 SOURCE FILE NAME ERROR nesprávně zapsaná jméno souboru  
 SOURCE FILE READ ERROR chyba čtení zdrojového souboru  
 OUTPUT FILE WRITE ERROR chyba při zápisu souboru  
 CANNOT CLOSE FILE soubor nelze uzavřít

DIR (d.) [jméno]  
 --- Zobrazí obsah adresáře souborů mimo označené SYS  
 d: - vypíše adresář zadané jednotky

jméno - vypíše pouze jméno, nebo NO FILE, pokud neexistuje  
 Je možno použít ve jménu znak ? místo libovolného znaku a znak x, který znamená cokoli.

DUMP jméno  
 --- Vypíše na obrazovku obsah souboru hexadecimálně. První je absolutní adresa začátku skupiny, potom následuje samotný hexadecimální výpis souboru.

ERA (d.) jméno  
 --- Zruší na disku specifikovaný soubor. Je možné použít znaky ? a x.

FORMAT  
 --- Formátuje disk. Mnejdříve generuje dotaz na určení jednotky, ve které budeme formátovat. Potom žádá o vložení disku do zadané jednotky. Je je oznámeno pomocí RETURN. Po naformátování žádá určení jednotky pro další FORMAT.

LOAD (d.) jméno  
 --- Příkaz provede převedení hexformátu na absolutní kód. Používá se pro převod překladu .HEX na spustitelný .COM soubor od adresy 100 h. U jména se předpokládá HEX.

PIP kam=odkud (přepínače)  
 --- Univerzální kopírovač. Program přenosu mezi logickými zařízeními. Bez udání parametrů se hlásí "x" a může pracovat až do zadání RETURN na dotaz "x".  
 kam - povolené označení výstupu  
 d: jméno souboru na disku  
 CON: logická systémová konzola  
 CRT: rychlá zobrazovací jednotka  
 UCI: uživatelské definovaná konzola  
 LST: logická tiskárna  
 LPT: fyzická tiskárna  
 ULI: uživatelská tiskárna  
 ITY: dálnopis  
 PRN: jako LST, přenáší tabelátor, čísluje řádky a a stránkuje po 60 řádcích  
 PUN: logické zařízení výstupu znaků  
 PIP: děrovač  
 UP1: uživatelský děrovač  
 UP2: uživatelský děrovač  
 OUT: speciální zařízení, nutno připojit programem  
 MUL: pseudozařízení, 40 prázdných znaků  
 EOF: pseudozařízení, generuje znak konec souboru

odkud - označení vstupu souboru  
 d: jméno souboru na disku  
 CON: logická systémová konzola  
 ITY: dálnopis  
 CRT: rychlá konzola  
 UCI: uživatelská konzola  
 RDR: logické zařízení znakového vstupu  
 PTR: snímač dřené pásky  
 UR1: uživatelský snímač  
 UR2: uživatelský snímač  
 INP: speciální vstup, nutno připojit program

přepínače - mezi odkud a začátkem přepínačů nesmí být mezer.  
 Pole přepínačů začíná znakem "\*" a končí "j".  
 B čte spojitě do paměti až do znaku (CTRL S) poté zavede obsah paměti na výstup (vstup z mg).  
 Dn kopíruje pouze n znaků z řádku, ostatní ignoruje.  
 E echo, kopíruje zároveň na obrazovku.  
 F vynechá ze vstupu hodnotu znaku 0ch a FF.  
 Gn povolí čtení oblasti nastavené USIRn.  
 H je provedena kontrola, zda se jedná o znaky HEX.  
 I vynechá záznamy "00" formátu HEX a nastaví H.  
 L převádě velké znaky na malé.  
 N čísluje řádky s potlačením nevýznamných nul.  
 N2 jako N, ale nuly nejsou potlačeny.  
 O ignoruje označení konce u netextových souborů.  
 P stránkuje po 60 řádcích.  
 Pn stránkuje po n řádcích.  
 Of-Z zastaví kopírování při výskytu řetězce ř, při přímém volání nelze zadat malá písmena.  
 R povolí kopírování souboru SYS.  
 SF-Z obnoví kopírování při výskytu řetězce ř.  
 In expanduje tabelátor na každý n ty sloupec.  
 U převádě malá písmena na velká.  
 V ověřuje zapsané informace kontrolním čtením.  
 W povolí přepsání souboru chráněného R/O.  
 X vypne kontrolu ASCII.  
 Z nuluje ve všech znacích paritní bit.

REN d: jméno= jménoold  
 --- přejmenuje soubor jménoold na jméno=.  
 SAVE n d: jméno  
 --- vytvoří kopii n 256 bytových stránek od adresy 100h.  
 STAT parametry  
 --- Slouží k zobrazení údajů o obsazení disků, rozsahu souborů, přiřazení logických a fyzických zařízení.  
 VAL: - informuje komplexní informací o systému, disku a přiřazených zařízeních.

bez parametru - zobrazí statut a rozsah volného prostoru pro všechny aktivní jednotky.  
 d: - zobrazí volný prostor na disku.  
 jméno-zobrazí údaje o počtu sektorů, délce souboru, počtu rozšíření, zprávu R/O nebo R/W.  
 jméno DR/O soubor je možné pouze číst.  
 jméno DR/W soubor je možné přepisovat.  
 jméno MSYS soubor nelze kopírovat ani zobrazit DIR.  
 jméno MDIR soubor je zobrazen DIR a lze jej kopírovat.  
 d: R/O nastaví jednotku pouze pro čtení, ale pouze do výmítky! po opětovném zavedení disku je normálně přístupný.  
 DEV: zobrazí aktuální přiřazení fyzických zařízení zařízením logickým (pro I/O operace).  
 log:=fyz: nastavení logických zařízení fyzickým.

Seznam logických a fyzických zařízení je u příkazu PIP.  
 d: DSK: zobrazí všechny informace o disku.  
 USR: informuje o uživatelských aktuálních disku.

SUBMIT jméno parametry  
 --- Slouží ke zpracování dávkových příkazů. Do dávků dává na místo formálních parametrů 00. U skutečné parametry zadané za jménem dávků. Jméno se očekává implicitně .SUB.

SYSGEN  
 --- Překopíruje systém CP/M na zadaný disk. Na dotaz se určí jednotka ze které má být systém natežen a potom jednotka, na kterou má být zaveden.

TYPE d: jméno  
 --- Předpokládá soubor ve formě ASCII. Překopíruje obsah na obrazovku.

USER n  
 --- Nastaví disk pro uživatele n (max 15).

XSUB  
 --- Modifikuje čtení znaků na soubor 000.SUB. Je tak možné vkládat do dávků i vstupní údaje pro programy.

UM d: jméno  
 Textový editor WORD MASTER Jméno je jméno souboru  
 Přehled příkazů obrazovkového režimu (<CTRL )  
 ~ V vložení textu z/v H mazání znaků zleva  
 ~ D VPRÁVO o znak G mazání znaků zprava  
 ~ T VPRÁVO o slovo L mazání slova zprava  
 ~ I VPRÁVO o sovo U mazání řádku dolů  
 ~ E nahoru o řádek Z mazání celého řádku  
 ~ X dolů o jeden řádek T vložení tabulátoru  
 ~ -- zač/konec stránky N vložení prázdného řádku  
 ~ TO zač/konec řádku P zápis řídicího znaku  
 ~ Y rolování nahoru  
 ~ Y rolování dolů DEL mazání znaku zleva  
 ~ R předcház. stránka ESC konec VIDEO MODE  
 ~ C násled. stránka

Program umožňující přímé prohlázení libovolného sektoru na zvolené jednotce v HEX a ASCII tvaru s možností jeho přepsání nebo opravy Na dotaz  
 SELECT DISK DRIVE (A-P)  
 odpovíme volbou jednotky Na volbu program odpoví  
 COMMAND ? (B,D,L,P,T,C,H,DUMP NEXT SECTOR>)  
 Zde je už možné zadávat jednotlivé příkazy  
 B - volí se číslo úseku , který se bude zkoumat  
 SELECT (HEX) BLOCK NO. zadáme číslo úseku  
 Číslo úseku je přímo v adresáři prohlášeného souboru  
 D - po dotazu SELECT DISK DRIVE (A-P) : volíme procesní disk  
 H - HELP Na konzolu se vypíše seznam příkazů s nápovědou  
 L - vypíše sektor přítomný v paměti na tiskárnu  
 P - nastavuje režim opravy sektorů přítomného v paměti  
 PATCHING ? (A,X,H,<QUIT>)  
 Volba režimů  
 A - znakové opravy Vhodné pro opravu ASCII souborů  
 Na dotaz ASCII PATCH - (HEX) ADDRESS : zadáme adr.  
 slabiky (bytu) v rámci sektoru . Zadáním RETURN se režim ukončí . Je vypsaná adresa a její obsah , který lze změnit . Mechceme-li obsah měnit , volbou RETURN .CR a LF nelze přepsat . Ukončení pomocí "C režim A  
 X - nastavuje hexsdecimální režim oprav.Ovládá se jako režim A  
 H - výpis pomocné zprávy  
 RETURN konec režimu oprav  
 TO - ukončí režim opravování  
 R - opakovaný výpis obsahu aktuálního sektoru v paměti  
 T volí se přístup k sektoru pomocí čísla stopy a sektoru  
 SELECT (DECIMAL) TRACK NO. ;  
 zadá se dekadicky číslo stopy (0 - 73)  
 SELECT (DECIMAL) SECTOR NO. ;  
 zadá se dekadicky číslo sektoru (0 - 26)  
 Stopy se zadávají od 0 . Sektory jsou číslovány od 1 .  
 Ja provedena kontrola zda zadané číslo sektoru nebo stopy není větší než skutečný počet . Pomocí RETURN je nastavena počáteční hodnota , nebo se pokračuje krokem na další

Přehled příkazů příkazového režimu  
 + - směr na začátek + směr ke konci souboru  
 @ označuje znak <CR> nebo <LF>  
 # označuje znak <ALT> <CR> <LF> nebo "Z  
 n představuje číslo, implicitně je 1, # = 65535  
 +-nC ukazatel o n znaků +nD vymazat n znaků  
 +-nL ukazatel o n řádků +-nK vymazat n řádků  
 +-nI zobrazit n řádků +-nO posun o n řádků  
 +-nP posun o n stránek +-NB zač/konec souboru  
 n!text# vložení 'textu' n-krát  
 !@ přechod do vkladacího režimu  
 A@ n!text# připojení 'textu' za konec řádku  
 n<...># opakování příkazů n-krát  
 +-n!key# vzhledání n-tého opakování řetězce  
 +-n!mkey# jako F, ale v celém souboru  
 +-n!Skey#text# zamění n-krát řetězec key# za text#  
 +-n!Rkey#text# jako S, ale v celém souboru  
 Práce se soubory  
 Y!d: name, type čtení souboru za ukazatel  
 n!f(d: name, type) zápis n řádků do souboru  
 Změňování znaků  
 ~ N zameňuje <CR> => <LF> ~ Y zameňuje <ALT>  
 ~ A nahrazuje libovolný znak  
 ~ OX zameňuje libovolný znak kromě znaku X  
 Práce s O-bufferem  
 n!P přesune n řádků do O-BUFFERU a zruší v textu  
 n!OP připojení n řádků k O-BUFFERU  
 n!OC kopíruje n-krát obsah O-BUFFERU do souboru  
 OI zobrazení obsahu bufferu  
 OX vymaže obsah bufferu  
 O!X vypočítá příkazy v bufferu  
 n!O!text# příkaz zápis do bufferu  
 n/O!text# příkaz příkazy textu opakuje n-krát

UM d: jméno  
 Textový editor WORD MASTER Jméno je jméno souboru  
 Přehled příkazů obrazovkového režimu (<CTRL )  
 ~ V vložení textu z/v H mazání znaků zleva  
 ~ D VPRÁVO o znak G mazání znaků zprava  
 ~ T VPRÁVO o slovo L mazání slova zprava  
 ~ I VPRÁVO o sovo U mazání řádku dolů  
 ~ E nahoru o řádek Z mazání celého řádku  
 ~ X dolů o jeden řádek T vložení tabulátoru  
 ~ -- zač/konec stránky N vložení prázdného řádku  
 ~ TO zač/konec řádku P zápis řídicího znaku  
 ~ Y rolování nahoru  
 ~ Y rolování dolů DEL mazání znaku zleva  
 ~ R předcház. stránka ESC konec VIDEO MODE  
 ~ C násled. stránka

ERROR: NON -EXISTENT !  
 jméno disku , který není připojen , nebo neexistuje číslo stopy , úseku nebo sektoru  
 UNRECOVERABLE DISK ERROR !  
 nalezena neopravitelná chyba na disku  
 WARNING: CANNOT CALCULATE NO. OF TRACKS ON SELECT DISK  
 je připojen disk s velkou kapacitou a DUMP nezaručuje dále kontrolu zadání čísla stopy  
 ERROR: CANNOT CALCULATE BLOCK NO.  
 jako předcházi, DUMP není schopen vypočítat číslo blok

Program umožňující přímé prohlázení libovolného sektoru na zvolené jednotce v HEX a ASCII tvaru s možností jeho přepsání nebo opravy Na dotaz  
 SELECT DISK DRIVE (A-P)  
 odpovíme volbou jednotky Na volbu program odpoví  
 COMMAND ? (B,D,L,P,T,C,H,DUMP NEXT SECTOR>)  
 Zde je už možné zadávat jednotlivé příkazy  
 B - volí se číslo úseku , který se bude zkoumat  
 SELECT (HEX) BLOCK NO. zadáme číslo úseku  
 Číslo úseku je přímo v adresáři prohlášeného souboru  
 D - po dotazu SELECT DISK DRIVE (A-P) : volíme procesní disk  
 H - HELP Na konzolu se vypíše seznam příkazů s nápovědou  
 L - vypíše sektor přítomný v paměti na tiskárnu  
 P - nastavuje režim opravy sektorů přítomného v paměti  
 PATCHING ? (A,X,H,<QUIT>)  
 Volba režimů  
 A - znakové opravy Vhodné pro opravu ASCII souborů  
 Na dotaz ASCII PATCH - (HEX) ADDRESS : zadáme adr.  
 slabiky (bytu) v rámci sektoru . Zadáním RETURN se režim ukončí . Je vypsaná adresa a její obsah , který lze změnit . Mechceme-li obsah měnit , volbou RETURN .CR a LF nelze přepsat . Ukončení pomocí "C režim A  
 X - nastavuje hexsdecimální režim oprav.Ovládá se jako režim A  
 H - výpis pomocné zprávy  
 RETURN konec režimu oprav  
 TO - ukončí režim opravování  
 R - opakovaný výpis obsahu aktuálního sektoru v paměti  
 T volí se přístup k sektoru pomocí čísla stopy a sektoru  
 SELECT (DECIMAL) TRACK NO. ;  
 zadá se dekadicky číslo stopy (0 - 73)  
 SELECT (DECIMAL) SECTOR NO. ;  
 zadá se dekadicky číslo sektoru (0 - 26)  
 Stopy se zadávají od 0 . Sektory jsou číslovány od 1 .  
 Ja provedena kontrola zda zadané číslo sektoru nebo stopy není větší než skutečný počet . Pomocí RETURN je nastavena počáteční hodnota , nebo se pokračuje krokem na další

Přehled příkazů příkazového režimu  
 + - směr na začátek + směr ke konci souboru  
 @ označuje znak <CR> nebo <LF>  
 # označuje znak <ALT> <CR> <LF> nebo "Z  
 n představuje číslo, implicitně je 1, # = 65535  
 +-nC ukazatel o n znaků +-nD vymazat n znaků  
 +-nL ukazatel o n řádků +-nK vymazat n řádků  
 +-nI zobrazit n řádků +-nO posun o n řádků  
 +-nP posun o n stránek +-NB zač/konec souboru  
 n!text# vložení 'textu' n-krát  
 !@ přechod do vkladacího režimu  
 A@ n!text# připojení 'textu' za konec řádku  
 n<...># opakování příkazů n-krát  
 +-n!key# vzhledání n-tého opakování řetězce  
 +-n!mkey# jako F, ale v celém souboru  
 +-n!Skey#text# zamění n-krát řetězec key# za text#  
 +-n!Rkey#text# jako S, ale v celém souboru  
 Práce se soubory  
 Y!d: name, type čtení souboru za ukazatel  
 n!f(d: name, type) zápis n řádků do souboru  
 Změňování znaků  
 ~ N zameňuje <CR> => <LF> ~ Y zameňuje <ALT>  
 ~ A nahrazuje libovolný znak  
 ~ OX zameňuje libovolný znak kromě znaku X  
 Práce s O-bufferem  
 n!P přesune n řádků do O-BUFFERU a zruší v textu  
 n!OP připojení n řádků k O-BUFFERU  
 n!OC kopíruje n-krát obsah O-BUFFERU do souboru  
 OI zobrazení obsahu bufferu  
 OX vymaže obsah bufferu  
 O!X vypočítá příkazy v bufferu  
 n!O!text# příkaz zápis do bufferu  
 n/O!text# příkaz příkazy textu opakuje n-krát

UM d: jméno  
 Textový editor WORD MASTER Jméno je jméno souboru  
 Přehled příkazů obrazovkového režimu (<CTRL )  
 ~ V vložení textu z/v H mazání znaků zleva  
 ~ D VPRÁVO o znak G mazání znaků zprava  
 ~ T VPRÁVO o slovo L mazání slova zprava  
 ~ I VPRÁVO o sovo U mazání řádku dolů  
 ~ E nahoru o řádek Z mazání celého řádku  
 ~ X dolů o jeden řádek T vložení tabulátoru  
 ~ -- zač/konec stránky N vložení prázdného řádku  
 ~ TO zač/konec řádku P zápis řídicího znaku  
 ~ Y rolování nahoru  
 ~ Y rolování dolů DEL mazání znaku zleva  
 ~ R předcház. stránka ESC konec VIDEO MODE  
 ~ C násled. stránka

ERROR: NON -EXISTENT !  
 jméno disku , který není připojen , nebo neexistuje číslo stopy , úseku nebo sektoru  
 UNRECOVERABLE DISK ERROR !  
 nalezena neopravitelná chyba na disku  
 WARNING: CANNOT CALCULATE NO. OF TRACKS ON SELECT DISK  
 je připojen disk s velkou kapacitou a DUMP nezaručuje dále kontrolu zadání čísla stopy  
 ERROR: CANNOT CALCULATE BLOCK NO.  
 jako předcházi, DUMP není schopen vypočítat číslo blok

Program umožňující přímé prohlázení libovolného sektoru na zvolené jednotce v HEX a ASCII tvaru s možností jeho přepsání nebo opravy Na dotaz  
 SELECT DISK DRIVE (A-P)  
 odpovíme volbou jednotky Na volbu program odpoví  
 COMMAND ? (B,D,L,P,T,C,H,DUMP NEXT SECTOR>)  
 Zde je už možné zadávat jednotlivé příkazy  
 B - volí se číslo úseku , který se bude zkoumat  
 SELECT (HEX) BLOCK NO. zadáme číslo úseku  
 Číslo úseku je přímo v adresáři prohlášeného souboru  
 D - po dotazu SELECT DISK DRIVE (A-P) : volíme procesní disk  
 H - HELP Na konzolu se vypíše seznam příkazů s nápovědou  
 L - vypíše sektor přítomný v paměti na tiskárnu  
 P - nastavuje režim opravy sektorů přítomného v paměti  
 PATCHING ? (A,X,H,<QUIT>)  
 Volba režimů  
 A - znakové opravy Vhodné pro opravu ASCII souborů  
 Na dotaz ASCII PATCH - (HEX) ADDRESS : zadáme adr.  
 slabiky (bytu) v rámci sektoru . Zadáním RETURN se režim ukončí . Je vypsaná adresa a její obsah , který lze změnit . Mechceme-li obsah měnit , volbou RETURN .CR a LF nelze přepsat . Ukončení pomocí "C režim A  
 X - nastavuje hexsdecimální režim oprav.Ovládá se jako režim A  
 H - výpis pomocné zprávy  
 RETURN konec režimu oprav  
 TO - ukončí režim opravování  
 R - opakovaný výpis obsahu aktuálního sektoru v paměti  
 T volí se přístup k sektoru pomocí čísla stopy a sektoru  
 SELECT (DECIMAL) TRACK NO. ;  
 zadá se dekadicky číslo stopy (0 - 73)  
 SELECT (DECIMAL) SECTOR NO. ;  
 zadá se dekadicky číslo sektoru (0 - 26)  
 Stopy se zadávají od 0 . Sektory jsou číslovány od 1 .  
 Ja provedena kontrola zda zadané číslo sektoru nebo stopy není větší než skutečný počet . Pomocí RETURN je nastavena počáteční hodnota , nebo se pokračuje krokem na další

Přehled příkazů příkazového režimu  
 + - směr na začátek + směr ke konci souboru  
 @ označuje znak <CR> nebo <LF>  
 # označuje znak <ALT> <CR> <LF> nebo "Z  
 n představuje číslo, implicitně je 1, # = 65535  
 +-nC ukazatel o n znaků +-nD vymazat n znaků  
 +-nL ukazatel o n řádků +-nK vymazat n řádků  
 +-nI zobrazit n řádků +-nO posun o n řádků  
 +-nP posun o n stránek +-NB zač/konec souboru  
 n!text# vložení 'textu' n-krát  
 !@ přechod do vkladacího režimu  
 A@ n!text# připojení 'textu' za konec řádku  
 n<...># opakování příkazů n-krát  
 +-n!key# vzhledání n-tého opakování řetězce  
 +-n!mkey# jako F, ale v celém souboru  
 +-n!Skey#text# zamění n-krát řetězec key# za text#  
 +-n!Rkey#text# jako S, ale v celém souboru  
 Práce se soubory  
 Y!d: name, type čtení souboru za ukazatel  
 n!f(d: name, type) zápis n řádků do souboru  
 Změňování znaků  
 ~ N zameňuje <CR> => <LF> ~ Y zameňuje <ALT>  
 ~ A nahrazuje libovolný znak  
 ~ OX zameňuje libovolný znak kromě znaku X  
 Práce s O-bufferem  
 n!P přesune n řádků do O-BUFFERU a zruší v textu  
 n!OP připojení n řádků k O-BUFFERU  
 n!OC kopíruje n-krát obsah O-BUFFERU do souboru  
 OI zobrazení obsahu bufferu  
 OX vymaže obsah bufferu  
 O!X vypočítá příkazy v bufferu  
 n!O!text# příkaz zápis do bufferu  
 n/O!text# příkaz příkazy textu opakuje n-krát

UM d: jméno  
 Textový editor WORD MASTER Jméno je jméno souboru  
 Přehled příkazů obrazovkového režimu (<CTRL )  
 ~ V vložení textu z/v H mazání znaků zleva  
 ~ D VPRÁVO o znak G mazání znaků zprava  
 ~ T VPRÁVO o slovo L mazání slova zprava  
 ~ I VPRÁVO o sovo U mazání řádku dolů  
 ~ E nahoru o řádek Z mazání celého řádku  
 ~ X dolů o jeden řádek T vložení tabulátoru  
 ~ -- zač/konec stránky N vložení prázdného řádku  
 ~ TO zač/konec řádku P zápis řídicího znaku  
 ~ Y rolování nahoru  
 ~ Y rolování dolů DEL mazání znaku zleva  
 ~ R předcház. stránka ESC konec VIDEO MODE  
 ~ C násled. stránka

M88255 - paralelní vstup/výstupní brány PA, PB, PC.

Profil řídícího slova pro různé režimy práce:

76543210 - bity výstupního (řídícího) bytu posílaného do CMR

0000011X - režim nastavení bitu PC (bit) do hodnoty X (0/1);
0000010X = '02-PC0=0; = '01-PC0=1
0000011X = '02-PC1=0; = '03-PC1=1
0000010X = '04-PC2=0; = '05-PC2=1
0000011X = '06-PC3=0; = '07-PC3=1
0000010X = '08-PC4=0; = '09-PC4=1
0000011X = '0A-PC5=0; = '0B-PC5=1
0000010X = '0C-PC6=0; = '0D-PC6=1
0000011X = '0E-PC7=0; = '0F-PC7=1

1 - aktivní druh provozu (mody 0,1,2)
21X - 00-Skupina A mod 0 (nebo: 01-mod 1; 10-mod 2)
1 X - 0-PA OUT/výstup (1-PA IN/vstup); OUT 29,144
1 X - 0-PCH OUT (1 PCH IN); PCH=PC4/7)
1 X - 0-Skupina B mod 0 (1-skupina B mod 1); CONTROL 4,3,132
1 X - 0-PB OUT/výstup (1-PB IN/vstup)
1 X - 0-PCL OUT/výstup (1-PCL IN); PCL=PC0/3)

Mod 0 ve všech branách (zápis nebo čtení dat):

76543210 K PA PB PCH PCL
10000000 = '08=128 out out out out
10000001 = '81=129 out out out in
10000010 = '82=130 out in out out
10000011 = '83=131 out in out in
10001000 = '88=136 out out in out
10001001 = '89=137 out out in in
10001010 = '8A=138 out in in in
10001011 = '8B=139 out in in in
10010000 = '90=144 in out out out
10010001 = '91=145 in out out out
10010010 = '92=146 in in out out
10010011 = '93=147 in in out in
10011000 = '98=152 in out in out
10011001 = '99=153 in out in in
10011010 = '9A=154 in in in in
10011011 = '9B=155 in in in in

Mod 1 (HANDSHAKE) pouze v PA (+PC3/7) a PB (+PC0/2):

76543210 K PA PB PCi (např: 7=PC7; ...=0BF)
10100000 = 'A6=160 out PC7=0BF, PC6=ACK, PC3=INTRA
10100001 = 'B0=176 in PC4=STB, PC5=IBF, PC3=INTRA
10000100 = '84=132 out PC1=0BF, PC2=ACK, PC0=INTRB
10000110 = '86=134 in PC2=STB, PC1=IBF, PC0=INTRB
a jejich kombinace, nebo kombinace s modem 0.

Mod 2 pouze v PA (+PC3/7); PB může mít mody 0 nebo 1:

76543210 K PA PB (PC3/7 viz u PA mod 1)
11000000 = 'C0=192 i/o out(mod 0)
11000011 = 'C2=194 i/o in (mod 0)
11000101 = 'C4=196 i/o out(mod 1)
11000111 = 'C6=198 i/o in (mod 1)

M88254A - seriový synchronní/asynchronní vysílač/přijímač

Formát instrukce pro synchronní provoz:

76543210 - bity řídícího slova CMR (Control Word Register)

00 - synchronní provoz;
xx00 - počet bitů: xx=00-5bitů; 01-6bitů; 10-7bitů; 11-8bitů
x 00 - PEN (Parity Enable)-kontrola parity: x=0-ne, x=1-ano;
x 00 - EP (Even Parity)-druh parity: x=0-lichá; x=1-sudá;
x 00 - synchronizace x=0-externí; x=1-interní;
x 00 - počet synchronizačních znaků: xx=2-2s; x=1-1s;
00001100 = '0C=12 CONTROL 1,1;12 -provoz:sync+8b-PEN-EP+int+2sz
zápis synchronizace: CONTROL 1,1;170,85 -2sz; 170=AA, 85=55.
Páre musí následovat povel zahájení přenosu, tj. celkem 4 byty.

Formát instrukce pro asynchronní provoz:

76543210 - bity řídícího slova CMR;
xx - rychlost (baud): xx=01-1s; 10-16s; 11-64s;
xx - počet bitů: xx=00-5bitů; 01-6bitů; 10-7bitů; 11-8bitů;
x - PEN-kontrola parity: x=0-ne; x=1-ano;
x - EP-druh parity: x=0-lichá; x=1-sudá;
xx - počet stop-bitů: xx=01-1sb; xx=10-1.5sb; xx=11-2sb;
11001110 = 'CE=206 -provoz:as+16+8b-PEN-EP+2sb;
a musí následovat povel zahájení přenosu.

Formát povelu pro zahájení nebo změnu přenosu:

76543210 - bity řídícího povelu vyslaného do CMR;
1 - TXEN-povoleno vysílání; (0-vysílání nepovoleno);
1 - DTR-Data Terminal Ready; 1-zápnout modem; (0-vypnout);
1 - RXEN-povoleno příjem dat; (0-nepovoleno);
1 - SBREAK-přerušení provozu; (0-normální provoz);
1 - ER-Error Reset - vynulování příznaků chyb: PE, OE, FE
1 - RTS-(Request To Send) - výzva k vysílání (pro modem);
1 - IR-(Internal Reset) - vnitřní nulování;
1 - EH-(Enable Hunt) - povoleno vyhledávání synchron. znaků;
00110111 = '37=55 - CONTROL 1,1;55 -povel zahájení přenosu.

Stavové slovo M88254A: (S=STATUS1,1; S=INP(31))

76543210 -status bity;
x - TXRDY: x=1-nezá data k vysílání; x=0-vysílá data;
x - RRDY: x=1-přijímač přijal data; x=0-čeká data;
x - TXE: x=1-vysílání ukončeno; x=0-neukončeno;
x - FE-Parity Error: x=1-chyba parity; x=0-parita správná;
x - FE-Frame Error: chyba rámce: x=1-nepřijel STOP bit;
x - OE-OverRun Error: x=1-nepřevzata včas přijatá data;
x - SYNDET-SYNchronizace DETekována -přijaty synchronizaky;
x - DSR: x=0-modem připraven (odpověď na signál DTR);
00001010 -S=0A=10 - byte dat přijat s chybou parity.

CTC 8253-16 bitové čítače/časovate: CT0, CT1, CT2.

Formát řídícího slova CW (Control Word) pro CMR: CONTROL 5,3,CM

76543210 - bity řídícího slova (bytu):

00 - CT0-čítat 0;
01 - CT1-čítat 1;
10 - CT2-čítat 2;
11 - Neplatné;
00 - čtení se vzorkováním;
01 - R/W LB - čtení/zápis vyššího bytu
10 - R/W LB - čtení/zápis nižšího bytu
11 - R/W LB, HB - čtení/zápis obou bytů
0000 - Mod 0: čítá dolů, nahodí OUT, čítá dál;
0010 - Mod 1: čítá dolů, nahodí OUT; start po GATE=1
0100 - Mod 2: dělička CLK/N, výstup: impuls -OUT
0110 - Mod 3: dělička CLK/N, výstup: +OUT=N/2, -OUT=N/2
1000 - Mod 4: čítá dolů, výstup: -OUT; start po zápisu dat
1010 - Mod 5: čítá dolů, výstup: -OUT; start po GATE=1
00110000 - X - druh čítání: 0-binární; 1-decimální;
00110000 - 30=48 - CONTROL 5,3,48 OUT 95,48

Tabulka kódů řídících slov čítačů pro mody:

Table with columns: CT mod (0, 1, 2, 3, 4, 5), Vzorokuj, R/W LB, R/W HB, R/W LBHB for modes 1 and 2.

Konstanty nastavení čítačů při kmitočtu hodin CLK=2.048 MHz

Table with columns: Rychlost, Dělící poměr, Sestřnáctkové Dvořádkové, CONTROL 5,X, Baudbit/s, HB LB, LB, HB.



---  
 tvar příkazu (cill), protokol o překladu: zdroj/dl/p2...  
 Překladač makroassembleru pro jazyk symbolických  
 instrukcí mikroprocesoru I 8080 a Z 80. Cílem překladu  
 je relativní a přemístitelný modul strojového kódu.  
 Protokol o překladu může být dále zpracován programem  
 CGREF80. Získáme tak výpis opatřený přířovými odkazy.

**Spuštění překladače**

cil alternativní jméno cílového souboru s příponou  
 .REL. Není nutné .REL uvádět.  
 zdroj zdrojový text v JSA I8080 nebo Z80. Implicitní  
 přípona je .MAC.

přepínače /dl/p2  
 /C požadavek na vygenerování výpisu o překladu pro  
 CGREF80 s příponou .CRF.  
 /L požadavek na provedení protokolu o překladu do  
 souboru .PRN.  
 /O všechny číselná návěští a konstanty jsou vyjád-  
 řena v oktálové soustavě.  
 /R požadavek na vytvoření relativního modulu .REL.  
 /Z přepne se do režimu zpracování instrukcí Z80.

Při zadání jména M00 bez parametrů se přejde do opako-  
 vacího režimu indikovaného znakem \*.  
 \*-jméno provede překlad souboru jméno.MAC a  
 vygeneruje soubor jméno.REL

\*.jméno soubor .REL se negeneruje.  
 \*jméno1.jméno2=jméno2 provede překlad souboru jméno2.MAC a  
 generuje cílový soubor jméno1.REL s  
 protokolem o překladu jméno2.PRN.

\*.jméno pro výpis=jméno pořídí pouze protokol o překladu do  
 výstupního souboru na disku nebo na  
 zařízení: IY: = obrazovka  
 LSI: = tiskárna

\*jméno1=jméno2 přeloží soubor jméno2.MAC na soubor  
 jméno1.MAC.

Zprávy o chybách

- A chyba argumentu direktivy ( nadř. PAGE i aj. )
  - C chyba konstanty podmíněných bloků
  - D odkaz na vícenásobně definované jméno
  - E chyba vnějšího symbolu
  - M vícenásobně definované jméno
  - N chyba v číselné konstantě
  - O chybný typ operace nebo chybná syntaxe
  - P chyba průchodu (nadř. hodnota definována EOU je při  
 druhém průchodu lina než při prvním průchodu).
  - Q problematická chyba, význam je pouze upozorňující
  - R chyba přemístitelnosti u relativních bloků
  - U nedefinované jméno
  - V chyba hodnoty
- Zprávy o průběhu překladu
- NO END STATEMENT ENCOUNTERED ON INPUT FILE  
 chybný příkaz END na konci zdrojového souboru  
 UNTERMINATED CONDITIONAL  
 minimálně jeden podmíněný blok nebyl ukončen  
 UNTERMINATED REPT/IRP/IRPC/MACRO  
 neukončené makro  
 NO/XX FATAL ERROR(S) ( YY WARNINGS)  
 indikuje XX chyb, YY upozornění

?COMMAND ERROR  
 v příkazovém řádku je neznámý parametr  
 ?FILE NOT FOUND  
 nenašel se vstupní soubor  
 ?CAN'T ENTER FILE  
 nelze ukončit výstupní soubor

**Protokol o překladu**

implicitně je typu .PRN. Pro přepínač /C je přípona .CRF.  
 Řádek překladu strojové instrukce má následující tvar  
 (nnnn) (o) aaaaam xx (vv/zzzzm) (C)(+) zdrojový řádek  
 (nnnn) číslo řádku generovaného CGRAF80  
 (e) kód chyby

relativní ukládací adresa  
 aaam kód instrukce  
 xx kód instrukce  
 (vv/zzzzm) 2. a 3. slabika instrukce  
 znak -m- má tyto významy  
 \* relativní kódový úsek (LSEG)  
 \* relativní datový úsek (DSEG)  
 \* relativní datový úsek COMMON  
 \* mezera absolutní úsek (ASEG)  
 \* vnější hodnota (EXRN)  
 (C) příznak, že řádek pochází z vloženého zdrojového  
 souboru (INCLUDE)  
 (+) příznak, že řádek pochází z makroinstrukce

**Zpracovávání jazyků**

Jedná se o JSA mikroprocesoru I8080 nebo po přepnutí Z80.  
 Formát příkazového řádku je následující:  
 (návěští: |) (operace) (operand) (komentař)  
 OPERANDY

Rezervovaná jména: A, B, C, D, E, H, L, M, SP, PSW.  
 Počítadlo adres: a  
 Obsahuje aktuální stav počítadla adres  
 Definovaná jména

Jedná se o jména návěští, konstanty definované pomocí EOU a SEL.  
 Jména jsou libovolně dlouhá, významných je pouze prvních 8 znaků.  
 Jména mohou obsahovat pouze následující znaky  
 písmena A-Z a a-z, čísla 0-9 (jméno nemůže začít číslem) a znaky  
 '.', '?', '@', a

Typy jmen

- Pevná jména - jsou to návěští ne konstanty definovaná EOU
  - Proměnná jména - jsou to jména definovaná pomocí SEL
  - Vnější jména - jsou to externě definovaná jména, znamená to,  
 že jsou v daném modulu používána, ale definice  
 je v jiném modulu
  - Veřejná jména - jsou jména v modulu nadefinována, ale pří-  
 tupná jsou i z jiných modulů (zde jsou vnější)
- Číselné soustavy

Různé indikátory za číselnou konstantou udávají základ.  
 B = dvojková soustava, O nebo Q = oktálová soustava, bez znaku  
 nebo D = dekadická soustava, H = hexadecimální soustava.

- Výrazy
- \* aritmetický součet
  - '/' dělení
  - \* násobení
  - 'mod' zbytek po dělení
  - 'high' vyšší slabika hodnoty
  - x shl v posun x o v bitů vlevo
  - x shr v posun x o v bitů vpravo
  - x and v logické součin
  - not x negace x
  - x or v logický součet
  - x xor v konkvivalence
  - x eq v x rovnó v => pravda (i); jinak nepravda (0)

x ne v x nerovno v => pravda (1), jinak nepravda (0)  
 x lt v x menší než v => pravda (1), jinak nepravda (0)  
 x le v x menší nebo rovno v => pravda (1), jinak nepravda (0)  
 x gt v x větší než v => pravda (1), jinak nepravda (0)  
 x ge v x větší nebo rovno v => pravda (1), jinak nepravda (0)  
 nul x je-li parametr prázdný, vrací pravda (1)  
 Úseky programu

příkaz|typ úseku  
 absolutní úsek 'aseg' |relativní úsek dat 'dseg'  
 relativní úsek kódu 'cseg' |společný relativní úsek 'common'  
 Definování hodnot v paměti

význam  
 db definuje jeden byt nebo řetězec uložený mezi  
 znaky, pro oddělení konstant slouží znak  
 'dv' definuje obsah slova, oddělovac je  
 'dc' pro definici řetězců, poslední znak má nastaven  
 nejvyšší bit na hodnotu 1;  
 'ds' rezervuje úsek paměti v počtu x bytů  
 Podmíněné příkazy

Umožňuje použít zdrojový text pro různé varianty programu.  
 Na počátku zdrojového textu se pomocí direktivy SEI nastavují  
 příslušné konstanty, které při překladu přepínají příslušné  
 bloky. Nejčastěji se podmíněného překladu užívá při rozvoji  
 makroinstrukcí.

Struktura:  
 IF podmínka  
 .....překlad při splnění podmínky  
 [ELSE  
 .....překlad pro nepravdu podmínky  
 ENDIF konec podmíněného překladu  
 podmínka

direktiva podmínka  
 IF p p <= 0 IFI p p <= 0  
 IFE p p = 0 IFF p p = 0  
 IFI první průchod překladu IF2 druhý průchod překladu  
 IFDEF jméno je definováno IFNDEF jméno není definováno  
 IFB parametr je prázdný IFNB parametr není prázdný  
 Rizení výpisu překladu

titulek každé stránky v protokolu  
 jako předchozí  
 podtitulek stránky  
 nastává kolik je na stránce řádků  
 nová stránka  
 potlačení výpisu  
 obnovení výpisu  
 potlačení křížových odkazů  
 obnovení křížových odkazů

vypisuje se úplný rozvoj makroinstrukce  
 při rozvoji makroinstrukce se generuje  
 pouze cílový kód  
 vepisují se jenom řádky generující kód  
 umístuje do zdrojového textu rozsáhlé  
 komentáře

komentář  
 komentář  
 vypsání pro Z80

Direktiva význam  
 CONT zahájení překladu podmíněného bloku  
 ENDC konec překladu podmíněného bloku  
 \*EJECI přechod na novou stránku  
 DEFB ekvivalent DB  
 DEFS ekvivalent DS  
 DEFU ekvivalent DW

DEFM umísťuje do paměti řetězec  
GLOBAL definuje veřejné jméno  
EXTERNAL definuje vnější jméno  
Výběr možných instrukcí  
.9880 zapne instrukce pro 10880  
.280 zapne instrukce pro 280  
Modulární filozofie  
-----  
Modulární filozofie je vystavěna na možnosti pomoci PUBLIC  
definovat jména jako veřejná, tedy dostupná i v jiných modulech  
a pomocí EXTRN zpřístupnit jména definovaná v jiných modulech.  
Veřejné jméno MERRY  
-----  
Veřejné jméno MERRY obsahuje poslední adresu programu.  
Definice:  
MERRY: ds  
Makroinstrukce  
-----  
jméno MACRO parametr1.parametr2.parametr3...  
tělo makroinstrukce  
ENDM  
-----  
EXITH vystoupí z těla makroinstrukce před nalezením ENDM  
: komentář uvnitř makroinstrukce není její součástí  
: následující znak v parametru ztrácí řídicí význam  
<...> skutečný parametr obsahující řídicí znaky uložit do < >  
LOCAL definuje návěští lokální uvnitř makroinstrukce  
& spojuje text a formální parametry:  
přeloží pro x v jako: push b  
chvba: mvi b, v  
: mvi b, &x  
Makroinstrukce rozvoje:  
IPR fpar, <spar1, spar2, spar3, >  
fpar formální parametr  
spar seznam skutečných parametrů  
-----  
příklad:  
ipr x, <1.2.3.4.5.6.7.8.9>  
db x  
endm  
-----  
vtvoří devět definic db 1 až db 9  
IRPC fpar, <spar>  
formální parametr je nahrazen postupně znaky z řetězce  
spar nebo <spar> v případě, že obsahuje řídicí znaky.  
par udává kolikrát se má tělo opakovat  
REPT par udává kolikrát se má tělo opakovat  
ENDM konec těla makroinstrukci opakování  
-----  
LIB univerzální spojovací program - linker, spojuje rela-  
tivní programové moduly a knihovny v soustředěné soubo-  
ru typu .COM  
-----  
Zápis: lib jméno/p.jméno2/p...  
jméno specifikace jména relativního modulu nebo knihovny  
/p parametr řízení linkeru:  
/d:address nastavuje počátek datové oblasti dseg a common  
/h všechny číselné údaje jsou od této chvíle hexa-  
decimální  
decimální všechny číselné údaje jsou od této chvíle octa-  
lové  
/o všechny číselné údaje jsou od této chvíle octa-  
lové  
/e: jméno ukončení spojování a předání řízení systému =>  
při zadání jména vstupního souboru s /n a /x  
uloží cílový kód do tohoto souboru  
vynší se - nedefinovaná jména  
- startovací adresa programu  
- umístění programu v paměti  
při zadání jméno se toto chápá jako startovací  
adresu. jméno je v modulu definováno jako

veřejné stejné jako pro /e a předá řízení spojenému programu  
/vpis informací o programu  
/n označení vstupního souboru  
/r nastavuje linker do počátečního stavu  
/s označuje, že soubor je knihovna  
/u jako /m, ale nezveřejňuje veřejná jména  
/x vstupní soubor je typu .HEX  
/v vytvoření tabulky veřejných jmen  
/p:address adresa počátku uložení kódu  
-----  
Příklad:  
lib /p:300/d:600/test/n:test knihovna s/e  
data jsou uložena od adresy 600, kód od adresy 300  
je vytvořen cílový soubor test.com, modul test.rel se spoji  
s moduly knihovny knihovna.rel  
Zoráv linkeru  
-----  
po každém sestaveném příkazovém řádku  
DATA xxxx vvvv  
xxxx počátek programu  
vvvv první volná slabika v paměti  
pro parametr /d  
DATA uuuu vvvv  
PROGRAM xxxx vvvv  
uuuu začátek datového úseku  
vvvv adresa první volné slabiky za datovými úseky  
xxxx začátek kódových úseků  
vvvv adresa první volné slabiky za datovými úseky  
Po každém příkazovém řádku se vypíše všechna neuskokojená jména  
Na závěr zpracování vstupuje zpráva:  
[ xxxx vvvv zz ]  
-----  
xxxx startovací adresa programu  
vvvv první volná slabika za programem  
zz počet stránek o délce 256 slabik (desítkové)  
Zprávy o chybách  
-----  
%2ND COMMON LARGER /jméno/  
chvba při spojování úseku COMMON  
-----  
%MULTI.DEF GLOBAL jméno  
byla zveřejněna další definice jména  
-----  
%OVERLAYING ( ) AREA PUBLIC = jméno ( xxxx )  
.STARI = xxxx  
.EXTERNAL = jméno ( xxxx )  
Počátky dat a kódu nastavené parametry /o /d vzájemně  
překrývají  
?CAN I SAVE OBJECT FILE chvba při ukládání na disk  
?COMMAND ERROR chvbný příkazový řádek  
PROGRAM  
? INTERSECTING ( ) AREA PUBLIC = jméno (xxxx)  
EXTERNAL = jméno (xxxx)  
došlo k překrytí programové nebo datové oblasti  
?LOADING ERROR  
zadaný vstupní soubor má nesprávný formát - není typu REL  
?NO STARI ADDRESS  
program se má spustit, ale v žádném modulu není adresa vstupu  
?NOTHING LOADED  
bvl použit parametr /S /E nebo /G, ale není vstupní soubor  
?OUT MEMORY  
nedostatečná paměť pro zavádění programu  
? STARI SYMBOL - jméno - UNDEFINED  
jméno zadané v parametru /E nebo /G jako startovací adresa  
? jméno NOT FOUND  
soubor zadaný v příkazu neexistuje  
ABOVE  
ORIGIN ( BELOW

po zpracování parametru /G /E datová nebo programová oblast  
zasahuje nad nebo pod meze vyhrazené v paměti pro uživatelský  
program. Volba Y = spojovací program oblast přesune a pokrač-  
čuje, volba takéhokoliv jiného znaku vyvolá ukončení činnosti  
spojovacího programu  
-----  
LIB knihovnik  
slouží k vytváření knihoven relativních modulů genero-  
vaných překladačem M80 a spojitelných do souboru .COM  
-----  
Formát příkazového řádku  
-----  
LIB jméno knihovny vstupu.../p  
jméno knihovny obsahující všechny spojované moduly je ukončeno  
znakem =. Činnost je nevyhodnější ukončit pomocí C = návrat  
do systému. Ukončení pomocí /I vede k chybě v knihovně.  
-----  
Parametry  
-----  
/C opuštění zpracování knihovny a návrat na začátek  
/E uložení souboru a ukončení činnosti. nedoporučujeme  
/H používat  
/I hodnoty ve výpisech jsou v hexadecimálním formátu  
vypisí jméno modulu, délky modulů veřejných jmen a  
odkazů na veřejná jména  
/O hodnoty jsou udávány v oktalovém tvaru  
/R ukončení činnosti na zpracování knihovny a její uložení  
/U vypisuje všechna neuskokojená volání vnějších jmen  
Zadání samotného LIB bez parametru /P přejdeme do opakovacího  
režimu. Návrat do CP/M se provede pomocí C.  
Program LIB dovoluje vyjmout z knihovny některé moduly a uložit  
je do nové knihovny. Postup:  
LIB NEXTLIB=FORLIB<COS.SIN.EXP>/E  
Vytvoří novou knihovnu NEXTLIB z knihovny FORLIB. Jsou vyjmuty  
moduly COS, SIN A EXP. Další možnosti je interval modulu  
LIB NEXTLIB=FORLIB<COS,EXP>/E  
Vybere všechny moduly od modulu COS až do modulu EXP.  
Chybová hlášení  
-----  
CAN I ENTER FILE nelze vytvořit soubor na disku  
COMMAND ERROR chvba v příkazovém řetězci  
FILE NOT FOUND vstupní soubor nenašel  
FILE READ ERROR chvba při čtení  
FIRST MODULE IN UNIT1 CLEUSE AFTER LASI  
modul M1 se ve specifikaci <M1..M2>  
v knihovně nalézá až za modulem M2  
LIBRARY WRITE ERROR chvba při zápisu do knihovny  
MODULE NAME/NUMBER NOT FOUND  
modul zadaného jména nebyl nalezen  
nedostatek místa v paměti  
-----  
OUT OF MEMORY  
-----  
CREF80 (vstupní) vstup  
-----  
Vstupem je soubor přeložený překladačem M80 a parametrem  
/C. Vstupem je potom soubor je s příponou .CRF.  
Tento speciální soubor se dále zpracuje CREF80. Vytvoří  
se soubor se standardní příponou .PRN.  
-----  
Formát výpisu  
-----  
každý zdrojový řádek je opatřen číslem řádku, na které je v  
případě potřeby odkazováno v tabulce křížových odkazů  
na konci výpisu je zveřejněna tabulka křížových odkazů, ve  
kterých jsou jména v abecedním pořadí. Číslo řádku na kterém  
je jméno definováno, je označeno znakem #