

### BETÖLTÉSE:

Az assembler/editor gépi kódú program, betöltése a SYSTEM utasítással történik, fileneve EDI, belépési pontja 17152. Így betöltéskor a  $\alpha?$  promtra EDI választ ~~ké~~ kell adni, betöltés után  $\alpha?$  /17152-vel indítható és újraindítható. Ezután a  $\rightarrow$  INPUT prompt jelenik meg.

### INPUT:

A parancsokat a promtra válaszként kell megadni. Először a parancs betűjét, majd a címet/címeket kell megadni. A címek decimálisak, hexadecimálisak /végén H/ vagy oktálisak lehetnek/ végén O/.

Ha több paraméter szükséges, akkor mindegyikre külön prompt jelenik meg. A gépelési hibák visszalépéssel /balra nyúl/ javíthatók. A promptrra NEW LINE választ adva az assembler/editor vissza-tér az Input parancshoz.

### POINTER PARANCSONK:

Az assembler/editor sor-orientált, miképp maga a Z 80 assembler is. A forrásfile aktuális sorára belső pointer mutat.

- T - top - a pointer a file elejére megy
- B - botton - a pointer a file végére megy
- Dn - down - a pointer n sorral lejjebb megy /default n=1/
- Un - up - a pointer n sorral feljebb megy /default n=1/
- Fn - print - a képernyőre n sort listáz /default n=1/, a pointer ezután az utolsó listázott sorra mutat
- L - locate - a forrásfileben megkeres egy karaktersringet, a pointert a megtalált stringre állítja, ha nem talál ilyen stringet, a pointer a file végére áll

### ÁLLTALÁNOS PARANCSONK:

- H - how big - kiírja a forrásfile start és end címét
- K - kill - törli a forrásprogramot a memóriából
- S - sort - az előző assemblálás során keletkezett szimbólumtáblát rendezi és kiírja, a parancs után egy betűt írva csak a megadott betűvel kezdődő szimbólumokat írja ki  
opció? V=video/képernyő, E=external/printer
- A -assembler - a startól az end pszeudó utasításig terjedő forrásprogramot lefordítja  
opció? V=videó/képernyő, E=external/printer, C=cassette /object program/  
6 esetén újabb kérdés: address? filenév?  
default a leggyorsabb mód, amíg a program hibás

### KAZETTA PARANCSONK:

A forrás- és object programok save-elhetők és load-olhatók, object program esetén a parancs után O betűt kell írni.

- W - write - file-t ír a kazettára, forrás módban a memóriában lévő forrásfile-t írja ki, object módban prompt: address?, mindkét módban prompt: filenév?
- R - read - kazettáról file-t olvas, forrásmódban a memóriában lévő kurrens file után tölt /egyszerű merge/, object módban a file-t címek szerint tölti be, mindkét módban prompt: filenév?  
load-olás közben "L" látható, hibás load után "E", hibátlan load után a fileméret jelenik meg
- V - valid - ellenőrzi, hogy a kiírt file hibátlan-e /ugyanaz, mint a read, de nem tölt a memóriába/



AZ ASSEMBLER:

A forrásprogram utasításai Z 80 assembler utasítások. Minden programsor legfeljebb négy mezőt tartalmaz:

CIMKE: OPERÁTOR OPERANDUS; MEGJEGYZÉS

CIMKE:

Az utasításokat címkével lehet megjelölni. Ha olyan operátort használunk, mint JP vagy CALL, akkor operandusként meg kell jelölni a célt /pl. hogy hova kell ugrani/. Az assembler megengedi szimbolikus nevek címkékként történő alkalmazását.

Szimbolikus nevek kétféleképpen adhatók meg: az utasítás elejére írva /az assembler a programszámláló értékét rendeli hozzá/, és EQUATE PSZEUDO utasítással /ezzel tetszés szerinti értéket lehet a szimbólumokhoz rendelni, pl. BACKSPACE: EQU 8/.

A szimbólum első karaktere csak betű lehet, egyébként tartalmazhat számokat hossza legfeljebb 7 karakter. Operátor nevek, regiszter nevek és feltétel kód nevek nem lehetnek szimbólumok.

OPERÁTOROK:

Az assemblernek a Z 80 utasításkészlet 74 operátorán kívül néhány pszeudo utasítása is van, mégpedig:

- END - end assembly, operandus nincs  
ezzel kell lezárni a forrásprogramot, különben hiba üzenetet kapunk
- DS - define storage, egy operandus  
megadott számú object byte-ot üresen hagy, pl. szöveg buffer, stack, stb. számára helyfoglalás
- DW - define word, egy operandus  
az object file-ban egy kétbyte-os szót generál, a byte-okat fordított sorrendben, ahogy az a Z 80 utasítások számára szükséges, pl. BUFFER: DW VALUE
- DB - define byte s, több operandus  
az operandus értékét az object file-ba generálja, több operandus lehet vesszőkkel elválasztva, pl.: DB 6, 93H, 'I', 80H  
az operandusok kifejezések is lehetnek, a program számláló minden operandus után növekszik, mintha azok külön sorban lennének, ASCII stringek is lehetnek az operandusok, pl.: MSG: DB 'JO REGGELT'
- EQU - equate, egy operandus  
szimbólumhoz értéket rendel, pl. NEW LINE: EQU 12
- ORG - orgin, egy operandus  
az object fileban kezdő helyét határozza meg, a programban tetszés szerint többször alkalmazható a kódrészek különböző kezdőpontokra helyezéshez
- LOAD - load memory, egy operandus  
object kód memóriába betöltése az előállításkor megadott címre, ez a load teljesen független az assembler indításkor megadott output opciótól, a load-olás a következő ORG-ig tart

OPERANDUSOK:

Az operandusok száma nulla /pl. NOP/, egy /pl. CP/, kettő /pl. BIT/, és egy vagy kettő /pl. JR/ lehet. Az operandusok lehetnek regiszter nevek /pl. A, B, DE, IX, .../ és feltétel nevek /pl. Z, NZ, C, M, .../, ASCII literálok /'vagy " köze zárt karakterek/, számjegyek /számjeggyel kezdődő decimális, hexadecimális, oktális értékek/, programszámláló /futáskor PC-t módosítja/.

Az előző adattípusokból matematikai műveletekkel kifejezések alkothatók, amelyek mindenütt használhatók számok helyett.

MEGJEGYZÉS:

Megjegyzések az assemblerben bárhol elhelyezhetők, a megjegyzés vége a sor vége.

HIBAKEZELES:

Ha az assembler hibát talál a forrásprogramban, megáll, hibaüzenetet ad, a hibás sort kiírja, a pointert erre a sorra állítja.

- DOUBLE SYMBOL - többször deklarált szimbólum
- RSVD - szimbólum neve lefoglalt szó /reserved word/
- FULL - szimbólum tábla betelt
- ORG - nincs kezdőpont /ORG utasítás/
- OPND - operandus hiba
- UNDEF - hiányzó szimbólum
- EOF - hiányzó END
- HUH? - sületlenség /syntax error/

ASSEMBLER LISTA:

Képernyőre vagy printere készülhet. A lista oldalanként készül, rövid szünetekkel. A szünet ideje alatt bármely billentyűt megnyomva az oldal bármely következő billentyű megnyomásáig a képernyőn marad. Az oldalméret képernyőn 15, printeren 16 sor.

EDITOR PARANCSONK:

- Zn - ZAP - a pointer helyétől kezdve n sort töröl /default n=1/
  - E - ENTER - beszúrás a pointer helye elé, sorszámokat ad, beszúrás vége csak "." karaktert tartalmazó sor
  - N - NEW - a pointer sorát új sorra cseréli
- A sorszámok csupán az editálást segítik, a filenak nem részei.

DEBUG /programbelövő/ PARANCSONK:

- G - GOTO - vezérlés átadás a felhasználói programnak, prompt? belépési cím? breakpoint cím? /default az aktuális programszámláló/, breakpoint default nincs, ha van, akkor egy cím felhasználásával a byte megőrződik és egy RST 38H utasítás kerül beszúrára
- C - COPY - kezdőcímtől végcímig terjedő memória blokk áthelyezése cél-címre
- F - FILL - kezdőcímtől a végcímig feltölti a memóriát egy 'data' konstanssal
- X - EXAMINE - a felhasználói gépi kód státuszát írja ki, felül a fő regiszterekkel, a mások sorban az alternatív regiszterekkel
- M - MODIFY - a memóriaa tartalmát vizsgálja és módosítja, a megadott címet és tartalmát kiírja, a tartalom átírható, majd a következő címre ugrik, kilépés "."-tal
- Q - QUERY - a memóriából 8 byte-ot kiír a következő formában: cím, hexadecimális, ASCII a 7. bit nélkül, ASCII a 7. bittel /a harmadik mező megjeleníti azokat a karaktereket, amelyekben a 7. bit 1, a negyedik mező ezeket grafikus karakterenként mutatja/, a kontroll karakterek "."-tá alakítva jelennek meg. Q a 0000 címtől Qnnn az nnnn címtől kezd a kiírást, majd Q onnan folytatja tovább.