

MACRO ASSEMBLER

Az ASSEMBLER Programozás szinte elképzelhetetlen, vagy legalábbis nehézkes egy jól kezelhető, címkézhető fordító program nélkül. A következő leírás egy ilyen ASSEMBLER fordítóprogram kezelését és használatát mutatja be.

A macro indítása

Az EDITOR-ASSEMBLER program a HomeLab-3-as géphez készült, amelyet kazettáról a BASIC LOAD utasításával lehet betölteni.

Az epromban lévő MACROT BASICBŐL call 63810-el, ASSEMBLERBŐL 63810-el, vagy a DEBUGGER programból az Ó (hosszú O) billentyű megnyomásával lehet indítani.

A program a gép belső softveréből nagyon keveset használ, mivel önálló editorral rendelkezik.

A program a szabad memória területből Hex. 4300-ig terjedő területen terül el, de ezen kívül az indítása után több rendszerváltó táblát készít a saját működéséhez. Így Hex. 4300-alá is kerül belőle, ezért az alatta lévő területet ne használj fel programozásra.

A KIADHATÓ PARANCSONK:

- a parancsok egybetűsek
- minden parancs után CR- kell
- a parancsok két részre vannak bontva, általános és editor parancsokra.

ÁLTALÁNOS PARANCSONK:

- H - a program letárolásának START és END cimeit módosítja,
formátuma: H nnnn xxxx

- K** - törli a SURCLE File tartalmát a memóriából, ill. a H-val kiírható címtől címig.
- I** - csak üres surcle-file esetén adható ki általános üzemmódban. Új program készítésekor kell kiadni ezt a parancsot. Ekkor megjelenik az 1 sorszám és a kurzor. Ide már a programodat írhatod. Minden sort CR-el kell terminálni. A begépelte sor ne legyen hosszabb 1 képernyősornál. Minden CR után új sorszámot ad automatikusan. A sorszámoknak csak tájékoztatás szempontjából van jelentőségük!
Az első utasításod lehetőleg ORG..... legyen vagy REM-nek megfelelő; legyen, különben hibajelzést fogsz kapni.
Ha a sorszám után; gépelsz, akkor utána bármilyen karaktert használhatsz megjegyzésként. Az utasítások után is használhatsz megjegyzést.
Ha a sorszám után CR.-t gépelsz, akkor a programírás üzemmód (I) megszűnik.
- E** - Ha van SURCLE File a memóriában akkor Debuggerből történő melegindításnak felel meg. Ha a program írása során gépelési hibát követtél el és azt ki akarod javítani, akkor a sorszám után R-el lépj ki az írás üzemmódból, majd "E" nyomása után az utoljára látszó sort javíthatod. Javítás után CR-el kell a sort elküldeni.
- T** - az előzőleg az assemblálás során keletkezett szimbólum, ill. címke táblázatot írja ki rendezve, de előtte kiírja OPTION:-t, ami annyit jelent, hogy megkérdezi, hogy hová akarod kiírni. A válaszok lehetnek
- CR : ilyenkor sehova nem írja ki
 - V : videóra, azaz képernyőre
 - P : printerre
- Vigyázz! Ne használj nagyon hosszú címkeket, mivel azokat egy külön táblázatban rakja le.

A táblázat így néz ki:

	1 56COH	1 600H
EDI	1	SIMBOL
		1 SURCLE FILE

Ha hosszú szimbólumokat használsz, akkor ez a szimbólum táblázat hamar megtelik és belegyalogolsz a programod elejébe.

- A - Az ORG-tól az első END pszeudo utasításig terjedő SURCLE File-t lefordítja Z-80 utasításokra, egyben szintaktikai ellenőrzést végez, valamint cím és címkeszámítást is.

Előzőleg megkérdezi, hogy hova fordítsa le (OPCION)

A válaszok lehetnek

- V : videóra
- P : printerre
- CR : fordítás elkészül de azt nem írja ki sehova, illetve nem listázza ki.

Ha nagyobb programot akarunk lefordítani, akkor CR-el érdemes válaszolni, mert így kb. 4-szer olyan gyorsan fordítja le mint kiíratással. Ha fordítás során hibát talál, akkor hibaüzenetet küld, és kiírja a hibás sor sorszámát.

- M - kilépés a Macroból

EDITOR:

- ebben az üzemmódban lehet a már meglévő SURCLE-File-ban javításokat végezni.
- minden editálási parancs a képernyőre utoljára kiírt sorra vonatkozik.
- ide mutat az editor belső pointerre

ÁLTALÁNOS EDITOR PARANCSONK:

- CR - a pointer a file elejére megy, azaz az 1-es számú sorra (*)

- SP+CR - a pointer a file végére megy (x)
- F2+cursor le - listázás egyesével lefelé
- F2+cursor fel - listázás egyesével felfelé
- F2+SH+cursor le - listázás 31 sorral előre (x)
- F2+SH+cursor fel - listázás 31 sorral vissza (x)

x (csak akkor működik ha egyáltalán van mit 31 sorral előre vagy visszafelé listázni.)

- I - hatása nem azonos az üres SURCLE-Filevel kiadható I-vel. Az utolsónak kiírt sor után lehet vele beszúrást csinálni. Új sorszámot ad és a mögötte lévő sorszámokat automatikusan növeli eggyel, így könnyen lehet bárhova beszúrni egy vagy több sort. CR után új sorszámot ad, ekkor még egy CR-el tudunk kilépni a beszúrásos üzemmódból.
- D - az utolsónak kiírt sort törli
- E - az utolsónak kiírt sor módosítása, az E kiadása után a kurzor az utoljára kiírt soron jelenik meg. Az utasítás, ill. címke módosítását a Basicben megszokott módon végezzük. A módosítás végén CR-el terminálunk.

CR - kilépés EDITOR-ból

A SURCLE FILE KIMENTÉSE:

- W - hatására megkérdezi a program nevét. Lehetőleg rövid nevet adjunk, indítsuk el a magnót, majd CR-el elkezdjük a kimentést.

A SURCLE FILE BETÖLTÉSE:

- R - hatására a program Load üzemmódba lép

ASSEMBLER UTASÍTÁSOK:

Az assembler a Z-80-as CPU utasításait ismeri, ez ki van egészítve néhány pszeudo utasítással.

Ezek:

a./ ORG operandus

az object-file kezdő helyét határozza meg, azaz a fordítás után milyen címre kezdje el kiszámítani a címeket, illetve címkéket.

pld.:

1 ORG 7000H

2 RET

A példa alapján az első utasítást Hex. 7000- címre rakja le a fordítás során.

Az operandus egy szám, amely lehet Hex. vagy Decimális

pld.:

ORG 7000H /Hex/

ORG 12288 /Dec/

b./ EQU operandus

Szimbólumhoz értéket rendel

pld.:

11 PIPACS: EQU 3000H /Hex/

12 LAPAT: EQU 12288 /Dec/

a továbbiakban ha PIPACS nevű szimbólumra (címkére) hivatkozunk, akkor a program a fordítás során 3000-ret fog értelmezni és behelyettesíteni.

c./ LOAD operandus

Az operandus által meghatározott helyre tölti be az OBJEST programot. Hatása a következő ORG-ig tart.

pld.:

1. ORG 6000H
2. LOAD 8000H
3. PUSH AF

A fenti program alapján a fordítás során a lefordított PUSH AF utasítás Hex. 6000-es címre számítja ki, de a memóriába Hex. 8000-res címre rakja le a programot.

Ennek az eljárásnak akkor látjuk a hasznát, ha pld. 8000H-ra LOAD-oljuk le kipróbálás céljából, de az abban szereplő utasítások végleges helye 6000H lesz. Tehát ha el akarjuk kerülni azt, hogy a fordítás során nehogy felülírjuk az Assemblerünket, azaz "rágyalogjunk" akkor érdemes ezt az elkülönítést alkalmaznunk, ugyanis az ORG által definiált címre kiszámítja a programot, de a RAM területre az ORG nem tölti be. Ezt a feladatot csak a LOAD hajtja végre.

d./ END (nincs operandusa)

-- A forrásprogramot zárja le, használata kötelező, mivel hiánya hibát okoz a fordítás során.

e./ DS operandus

A DS operandusában megadott számú byte területet foglal le az OBJECT-File-ban.

pld.:

1. ORG 7000H
2. LOAD 7000H
3. VÁLTOZÓ: DS 5
4. LD A88H
5. LD (VÁLTOZÓ+3),A
6. END

A fenti program 7000-7005-ig terjedő területet lefoglalja és a 7003-as rekeszbe betölti A értékét.

A DS segítségével pld.: a rendszer-változóinknak tudunk helyet biztosítani a programban.

f./ DB operandusok, literálok, kifejezések

A DB az operandus értékét az OBJECT-File-ban generálja.
Több operandus lehet vesszővel elválasztva

pld.:

1. ORG 7000H
2. LOAD 7000H
3. BUFF: DB 1, 2, 3,4,ERTEK, 5BH
4. END

A fenti példában Hex. 7000-re lerakja 1-et

7001-re 2-ot

7002-re 3-at

7003-re 4-et

A 7004-es rekesztől kezdődően lerakja az "ERTEK" szó, ill.
kifejezés Hex. kódjait.

7004-re 5BH /E/

7005-re 52H /R/

7006-ra 54H /T/

7007-re 5BH /E/

7008-ra 4BH /K/

továbbá 7009-re 5BH

Tehát a DB utasítással könnyen tudunk táblázatokat készíteni,
amelyben szöveg is szerepelhet és ennek a szövegnek nem kell
nekünk kikeresni minden betűhöz tartozó HEX kódot, mert ezt
automatikusan elvégzi a program.

g./ DW operandus

az OBJECT-File-ban eg 2 bytes szót (címet) generál és azt for-
ditott sorrendben rakja le, úgy ahogy azt a Z-80 megkívánja

pld.:

1. ORG 7000H
2. LOAD 7000H

- utasítás-mnemonik: - csak Z-80 utasítások lehetnek és nem ismeri a Homelab assembler eltéréseit.
pld.: nem használható a MV utasítás

- operandusok: - lehetnek regiszternevek
feltételnevek
ASCII LITERAlok
szimbólumnevek
számjeggyel kezdődő Dec., Hex. értékek,
kifejezések, matematikai műveletek.

- megjegyzés: - ; -vel kezdődő ASCII karaktersorozat a file-
ban bárhol elhelyezhető. A hatása a sor vé-
géig tart. Ha sorszám után áll, akkor az e-
gész sor megjegyzésre használható.
Vigyázz, mert ha hosszú megjegyzéseket vagy sok
betűközt használasz, akkor a SURCLE-File gyorsan
növekszik és elérheti a LOAD-olt programod elejét.

ASSEMBLER HIBAKEZELÉS:

Az assembler a fordítás során magyar hibaüzenetet jelez ki.
A fordítás az első hiba után leáll, és kiírja a hibás sor
sorszámát, és a hibakódot.

KEZELÉSI TANÁCSOK A PROGRAMOZÁSHOZ:

- a programozásodat ORG utasítással kezdő
- a LOAD legalább 7000H-en kezdődjön
- rövid címkeket használja
- a címke után legyen betűköz
- soron belül ne használj a betűhöz helyett a TAB funkciót
- az utasításokat es operandusokat nagybetűvel írd
- ne felejtse el az END utasítást
- először definiálj és csak utána hivatkozz rá
- ha nincs bekötve a Centronix printer, akkor a P betűt ne
használj utasításként

3. LALI: DB BCH, 55H, 33H
4. BUFF: DW LALI
5. END

A fenti program így rakja le a byte-kat

7000H-ra	BCH	
7001H-ra	55H	DB
7002H-ra	33H	
7003H-ra	00H	DW
7004H-ra	70H	

Tehát a DW a 7000H címet rakta le, amelyen megtalálhatjuk a DB-ben definiált byte-k kezdő-címét. Ezek után ha BUFF-ra hivatkozunk, akkor az alatt 7000H-et kell értenünk.

A FORRÁSPROGRAM:

A forrásprogram utasítása a Z-80 assembler utasításai. Ismeri valamennyi standard utasítást, amely a Z-80-as CPU utasítás készletében szerepel. A gyártó által nem applikált utasításokat a program nem tudja kezelni, mivel ezek az utasítások könnyen helyettesíthetők, illetve feleslegesek még egy profi programozó számára is.

Minden programsor 4 mezőt tartalmazhat:

címke: utasítás-mnemonik; operandus; megjegyzés

- címke: - használata nem kötelező. Segítségével az u.n. szimbolikus címzést lehet megvalósítani
- első karaktere csak betű lehet
- utasítások, regiszternevek, és feltételkódok nem lehetnek az első karakterei
- a címkeket :-al kell lezárni
- a címke és a :-között nem lehet SPACE

- először papíron csináld meg a programodat
- először a "nagy" programodnak csak kis részleteit, szubrutinjait csináld meg és csak utána állítsd össze
- próbáld meg olyan szubrutinokat csinálni, amit más feladatok elvégzésére is tudsz majd alkalmazni
- Hex. 3000 felett már nem látszik a második memória lapon a programod, ezért a képernyő kezeléseket 8000H alatt csináld meg
- kevés PUSH műveletet használj, mert egy idő után már nem tudod követni, hogy mit raktál be és vettél elő
- a címkeket utólag is be tudod írni az utasítás elé, ha szükséges, ezért nem szükséges a sok címke, ha nem használod el csak a helyet foglalja
- ha a operandusban Hex.-számot adsz meg, és ez a szám betűvel kezdődik feltétlenül írd el egy 0-át
- ahogy nő a programod, úgy tárold ki rendszeresen magóra, nehogy több órai munkád vesszen kárba áramkimaradás, vagy program hiba miatt.

RST utasításokat a számok 8-szorosával kell használni.

pld.: az RST5 helyett RST 40 /Dec/

RST 28H /Hex/