

újra a rendszert a D3F8 címen.

Példa: az 5678h címen akarjuk a forrásprogramot elhelyezni.

. 1A88 78 56
G D3F8

A töréspontokat a rendszer úgy kezeli, hogy RST 28 utasításokat ír a megfelelő címekre és átírja az RST 28 ugrótáblát a 0. lapon. Ha RST 28 utasítást írunk a programunkba, az is töréspontként fog viselkedni.

1. BEVEZETŐ

A TVC-ASSEMBLER a 64 kilobyte-os TV-COMPUTER-hez készült; 32 k-s gépen a futtatásához megfelelő RAM bővítés szükséges. Perifériaként kazettás magnetofon és printer használható.

A programot a gép bekapcsolása után a LOAD paranccsal tölthetjük be. Sikeres betöltés esetén az alábbi üzenetet kapjuk:

TVC-ASSEMBLER 1.1
(C) 1987 Kós Géza
Ok

A továbbiakban ismertetendő utasításokat a BASIC-ben megszokott full-screen editor segítségével gépelhetjük be; 128 karakternél hosszabb sor esetén a rendszer a fölösleges karaktereket nem veszi figyelembe.

Ha valamilyen okból a RESET-et használjuk, a rendszer a RUN paranccsal újraindítható; ilyenkor nem törli az általunk írt assembler forrásszöveget.

2. Az ASSEMBLER utasításai

Az ASSEMBLER a szerkesztő és fordító utasításokon kívül különféle monitor- és programbelövő funkciókat is tartalmaz. Az utasítások nevében és a (nem decimális) konstansokban nem különbözteti meg a kis- és nagybetűket.

2.1. Szerkesztő és fordító utasítások

Az assembler forrásprogram a BASIC programokhoz hasonlóan számozott sorokból áll, a sorok beírása is teljesen hasonló a BASIC-hez.

Egy sor három részből áll, amelyek közül legfeljebb kettő hiányozhat:

- a) CÍMKE: Angol nagybetűkből és számjegyekből álló szöveg amely a fordítás során az aktuális fordítási címet jelöli. A végét kötelezően kettőspont (:) jelöli.
- b) UTASÍTÁS: Z80 assembler vagy valamilyen fordításvezérlő utasítás (lásd később).
- c) MEGJEGYZÉS: Pontosvessző (;) után ál-

ló szöveg, amelyet a rendszer fordítás közben nem vesz figyelembe.

Az egyes részek, illetve a Z80 utasítások és paramétereik között tetszőleges számú szóköz helyezhető el azzal a megkötéssel, hogy paraméter nélküli utasítás esetén a megjegyzés előtt legalább egy szóköznek szerepelnie kell.

2.1.1. Szerkesztő utasítások

A (*)-gal megjelölt utasítások végrehajtása a szóköz leütésével felfüggeszthető, a RETURN-nel megszakítható. Az n, n1, n2, d paraméterek decimális számokat jelentenek, a többi hexadecimális.

- L n1-n2 A BASIC LIST utasítás megfelelője; n1 és n2, illetve a - jel éppúgy hiányozhat, mint BASIC-ben. (*)
- P n1-n2 Ugyanaz, mint az L utasítás, de printerre nyomtat. (*)
- K n1-n2 Törli az n1-n2 sorokat. n1 és n2 éppúgy hiányozhat, mint az L és a P utasításnál (pl. a K

utasítás az egész programot törli).

Eszöveg Kinyomtatja azokat a sorokat, amelyek a szöveget tartalmazzák. (+)

n, d Automatikus sorszámozás. Több sor begépelésekor használhatjuk. Hatására a rendszer kinyomtatja az aktuális sorszámot, amely kezdetben n és minden sor után d-vel nő. Ha nem frunk be sort (pl. CTRL-Y majd RETURN) a sorszámozás abbamarad. n és d hiányozhat, a hiányzó paraméter értéke 10.

N n, d Átszámozza a forrásprogramot. Az első sor száma n lesz, a sorszámok d-esével nőnek.

2.1.2. Fordító utasítások

A Lefordítja a forrásprogramot és beírja a memóriába. Ha hibát talál, abbahagyja a fordítást.

AD Lefordítja a forrásszöveget, de nem tölti be a memóriába, hanem kinyomtatja a képernyőre. (+)

AP Ugyanaz, mint az AD utasítás,

de printerre nyomtat. (+)
S Fordítás után kilistázza a címkeket és értékeiket. (+)
SP Ugyanaz, mint az S, de printerre nyomtat. (+)

2.2. Monitor funkciók

cím b1 b2 ... bn

A címtől kezdődően beírja a memóriába a b1, b2, ... bn byte-okat. Ha az utasítás "törlés" és valamilyen szöveg áll, ezeket nem veszi figyelembe. A rendszer nem ad Ok üzenetet.

M cím

A cínnél kezdődő 64 byte-ot nyomtatja ki 8 sorban. A sorok elé "."-ot ír, így a memória tartalom megváltoztatása leegyszerűsödik. Ha a cím hiányzik, a legutóbbi M vagy D utasítás végétől kezd. (+)

MP cím1-cím2

A cím1 és cím2 közti byte-okat nyomtatja ki printerre. (+)

C cím1, cím2, h

A cím1-en kezdődő h db. byte-ot

átmásolja cím2-re.

D cím

A cím-en található gépi kódú utasítást visszafordítja assemblerré. A szóköz hatására fordítja le a következő utasítást, RETURN hatására leáll. Ha a cím hiányzik, a legutóbbi M vagy U utasítás végétől kezd.

DP címl-cím2

A címl és cím2 közti gépi kódú utasításokat visszafordítja és kinyomtatja printerre. (*)

I cím, h, b

A cím-nél kezdődő h byte-ot feltölti b-vel.

2.3. Programbelövő utasítások

Az ASSEMBLER segítségével kipróbálhatjuk gépi kódú programjainkat; sőt a futást akár lépésenként is nyomon követhetjük. Ha a programot nem lépésenként hajtjuk végre, kijelölhetünk maximum 8 töréspontot.

X Kinyomtatja a Z80 regiszterek tartalmát.

X reg névt

A megadott regiszterbe betölti t-t.

0

Kinyomtatja a töréspontokat.

D cím

Kijelöl egy új töréspontot.

QT cím

Törli a címen levő töréspontot.

QT

Törli az összes töréspontot.

G cím

Elindítja a cím-nél kezdődő gépi kódú programot. Ha a cím hiányzik, a PC regiszterben levő címtől indítja.

T cím

Lépésenkénti végrehajtás. A címet (ha nem hiányzik) betölti a PC regiszterbe, majd kinyomtatja a regiszterek tartalmát, a bekapcsolt flag-eket, lefordítja a soron következő utasítást és vár. Ha lenyomjuk a RETURN-t, a rendszer abbahagyja a parancs végrehajtását és Ok üzenetet ad. Ha a szöközt nyomjuk le, végrehajtja a gépi kódú utasítást és előlről kezd. Ha a G-t nyomjuk le, a Következő utasításig hajtja végre a gépi kódú programot (pl. ha egy LDIR utasítás vagy egy szubrutin működését nem akarjuk lépésenként nyomon követni, ezt használjuk)

2.4. Egyéb utasítások

H Kinyomtatja a forrásprogram határait a címketáblázat nélkül.

nl Az nl alapú számrendszerben felírt p számot kinyomtatja n2 alapú számrendszerben.

W "név" cím,h Kimentti a címen levő h byte-ot. A név hiányozhat, a filenév ilyenkor NÉVTELEN.

WI "név" Kimentti a forrásprogramot. A név itt is hiányozhat.

R "név" cím Betölti a név nevű file-t a címre. Ha a név hiányzik, az első file-t tölti be. Ha a cím hiányzik, oda tölti be, ahonnan kimentettük.

RI "név" Betölti a név nevű forrásszöveget a már meglevő szöveg végére (egyszerű merge). Mivel ezután a sorszámok nem megfelelő sorrendben lesznek, célszerű egy N utasítást is kiadni.

Z cím,h,ind,"név1","név2" Két file-t ment ki. Az első egy

BASIC program, a második a címen kezdődő h db. byte. Ha BASIC-ben betöltjük az elsőt, az automatikusan elindul, betölti a másodikat és elindítja az ind címen levő gépi kódú programot. Ezt az utasítást kész programok kimentésére használhatjuk.

3. A forrásprogramban felhasználható utasítások

3.0. Az utasítások konstans paramétereit

Paraméterként tetszőleges számokból, címkékből és aritmetikai jelekből (+, -, *, /) álló kifejezés állhat, a számokat a rendszer előjel nélküli egészként, modulo 65536 értelmezi.

- A számok lehetnek:
 - decimálisak;
 - binárisak, ezek elé % jelet írunk;
 - oktálisak, ezek elé # jelet írunk;
 - hexadecimálisak, ezek elé \$ jelet írunk;
 - valamilyen szöveg első karakterének ASCII kódja, a szöveget " vagy ' je-

lek határolhatják.

3.1. Fordításvezérlő utasítások

- ORG n** Az aktuális fordítási címet állítja be n-re. A fordítás megkezdésekor a fordítási cím 0.
- END** A lefordítandó szöveg végét jelzi, nem hiányozhat.
- LOAD n** Használata lehetővé teszi, hogy a lefordított programot ne az ORG által megadott címre töltsük be. Az AD és AP utasításokra nincs hatással.

3.2. Z80 utasítások

A Z80 utasítások szintaktikája megegyezik a megszokottal, kivéve az RST utasítást, amely után kötelezően egy hexadecimális szám áll (§ nélkül).

3.3. Egyéb utasítások

- DS n** A programban kihagy n byte-ot.
- DB bl, ..., bn** A programba belefordítja a bl...bn byte-okat. A paraméterek kö-

zött lehetnek szövegek is, ezeknek minden karakterét lefordítja. A szöveget " vagy ' jelek határolhatják.

DW w1, ..., wn

A w1, ..., wn kétbyte-os számokat fordítja le. Itt is szerepelhetnek szövegek, ezeket ugyanúgy fordítja le, mint a DB utasítás.

címke: EQU n

Hatására a címke értéke nem az aktuális fordítási cím, hanem n lesz.

4. Az ASSEMBLER felépítése, memóriatérkép

A program a 3. lapon, a ROM helyén van, így csaknem a teljes felhasználói memória használható:

- 1A80-1AFF Az ASSEMBLER változói
- 1B00-BF7E A felhasználható memória
- BF7F-BFFF Input puffer

A forrásprogram kezdete a rendszer elindításakor az 1B00 címen van. Ha máshol akarjuk elhelyezni, írjuk be az új címet az 1A88-on levő változóba és indítsuk