

Ami igaz az igaz, az LSI kiadásában megjelent SINCLAIR SPECTRUM JÁTÉK ÉS PROGRAM c. könyvsorozat I. kötetében már olvashattunk információt a GENS/MONS párosról. Mindettől függetlenül sok-sok Olvasói levél tartalma azt bizonyította, hogy sokan vannak, akiknek ez a könyv nincs birtokában, s ha jól tudjuk ennek a kötetnek az LSI már az utánnyomását sem tervezi. Íme tehát elevenítsük fel, mit is tud ez a két jól ismert program.

### GENS3 ASSEMBLER

Az általunk is tesztelt program felépítése.:

by: GENS3 (8254 bytes)

Itt megjegyeznénk, hogy forgalomba került a GENS3 M ill. M2 kibővített változata is, amely hatékony microdrive ill. lemezes kezelést tesz lehetővé. Ez utóbbi főkódjának hossza kb. 10.5 kbyte.

Az assemblert LOAD ""CODE xxxxx, vagy LOAD "GENS3"CODE xxxxx, utasítással tölthetjük be. Az xxxxx értéke tetszőleges, de célszerűen a RAM terület elején érdemes elhelyezni, mert mögé fog kerülni a forrásszöveg és a címtábla is.

LOAD "GENS3"CODE n esetén a program a RANDOMIZEUSR n utasítással indítható, az újraindításra két lehetőségünk van:

RANDOMIZEUSR (n + 56) - hidegindítást

RANDOMIZEUSR (n + 61) - melegindítást

A hidegindítás azt jelenti, hogy törölődik a forrásszöveg és a címtáblák is, míg melegindításnál csak a forráskódoktól búcsúzunk, a szövegrész és a címtáblák megmaradnak.

Indítás után az első kérdés a 'Buffer Size?', amelyre egy 0 és 9 közötti számmal válaszolhatunk. Ha csak 'ENTER'-t nyomunk, akkor beáll az alapértékre, ez 4. Itt állítjuk be a pufferek számát. Ezek a pufferek 64 byte-osak és a forrásszöveg beolvasásához is szükségesek lehetnek.

A GENS3 megszakítást nem engedélyez, így ha viszatérünk a BASIC-hez, akkor nekünk külön kell gondoskodni a megszakítás engedélyezéséről.

A 'TABLE SIZE?' kérdés a címtábla nagyságára kérdez rá (az 'A' nevű Editor-parancsnál). Elég itt is, ha csak 'ENTER'-rel válaszolunk.

Fordításnál különböző fordítási opciók közül választhatunk. Az 'option' kérdésre az alábbi opciókhoz tartozó számok összegét kell megadnunk:

- 1 - címtábla keletkezik a fordítás végén
- 2 - tárgykód nem keletkezik a fordítás végén
- 4 - nem kapunk assembly listát a fordítás végén
- 8 - az assembly listát nyomtatóra küldjük
- 16 - elhelyezi a tárgykódot
- 32 - kikapcsolja az ellenőrzést, a fordítás gyorsabb lesz  
pl. a 63-os opció megadása jelenti a leggyorsabb fordítást.

A forrássorok sorszámot kapnak a BASIC-hez hasonlóan. A bináris konstansok %-kal kezdődnek (pl. %111 = 7). A címkék pedig L előtaggal szerepelnek (pl. L13). A címszámlálóra való hivatkozásnál \$ (dollárjelet) használunk. A műveleteknél a modulus függvényre "?" jelre hivatkozunk, a logikai vagy függvényre

"@" (kukac) jellel, míg a logikai kizáró vagy függvényre a "!" (felkiáltó)-jelet használjuk.

#### Assembler direktívák (fordítónak szóló üzenetek):

ORG n : a tárgykód kezdőcímét jelöli ki. Előtte csak komment sorok állhatnak. Az n lehet kifejezés is.

EQU : az EQU kifejezést címkének kell megelőznie. Értékkadó utasításoknak számítanak.

DEFB kifejezés, kif., kif... : a fentivel szemben mindegyik kifejezést 8 bitre értékkel ki, és konstansként helyezi el

DEFW kif., kif... : a fentivel szemben most 2 byte-on teszi le a kifejezést

DEFS kif. : a kifejezésnek megfelelő számú byte-ot foglal le

DEFN "string": egy stringet tesz le a hosszának megfelelően (max. 256 byte)

If kif. : ha a kif = 0 akkor a forrássorokat nem veszi figyelembe, amíg nem találkozik vagy az ELSE vagy az END utasításokkal

ELSE : Ld. az If-nél

END : ismét bekapcsolja a fordítót

Megjegyzés : ezek a feltételes "ÁLKÓDOK" nem ágyazhatók be egymásba

#### Az assembler parancsai:

Az assembler direktívákhoz hasonlóan nincs hatásuk (közvetlen hatásuk) a tárgykódra, csupán az assembly lista formáját módosítják. E parancsok a forrássor elején + -gal (csillaggjellel) kezdődnek, ezután betű (nagy betű) következik, és ezt esetleg "+" vagy "-" jelek követhetik. A sor hátralevő része tetszőleges lehet, kivéve az L+, L-, D+, D- parancsokat. A következő parancsok érthetők el:

- \*E (eject) : 3 üres sort emel a kérelyőn vagy a nyomtatón
- \*Hs (head) : \*E valamint az "s" stringet kinyomtatja
- \*S : eddig a sorig listáz, a listázás bármely billentyűvel folytatható, viszont idő előtt leáll, ha +L vagy +S parancsokkal találkozik.
- \*L- : e sortól kezdve már nem listáz
- \*L+ : e sortól kezdve listáz
- \*D- : a címszámlálót ezután HEX formában írja ki
- \*D+ : a címszámlálót ezután DEC formában írja ki
- \*C- : a sor után rövidített assembly listát ad
- \*C+ : a sor után a teljes assembly sort kiírja
- \*F (filename) : szalagról beolvas egy szövegfile-t, amelyet lefordíthatunk, ennél fogva a szöveg csak minimális helyet foglal le a tárbán, amelyet kisebb adagokban olvashatunk be. A szövegfile-t korábban a "T" parancssal vittük ki a 'Buffer size?'-nak megfelelően. Ekkor a szövegfile blokkokban helyezkedik el, és amikor az "F" parancssal beolvasunk egy blokkot, akkor azt rögtön le is fordítjuk. A blokkok beolvasása mindaddig tart, amíg be nem olvastuk valamennyi blokkot.

Megjegyzés: a "F" parancs erre nem jól! Tehát, ha a forrásszöveg nagyon hosszú, akkor először azt a "T" parancssal kivisszük, majd "F"-el visszaolvassuk kisebb adagokban.

#### Az EDITOR parancsai (szövegfile szerkesztéséhez)

A GENS3 indítás után közvetlenül az editor parancsait várja. A továbbiakban a Symbol Shift-et SS-sel, a Caps Shift-et CS-sel fogjuk rövidíteni.

Az editor a ">" jelet használja parancskéréshez.

I n,m : Sorokat beszúrhatunk az n. sortól kezdve m lépéssel. Kilépés CS + 1-re. A már létező sorszámú sorokat törli (felülírja). Egy szövegsor 64 karakter lehet (buffer size).

L n,m : Listáz az n-től az m-ig. Kilépés CS + 1-gyel.

K n : Listázáskor legfeljebb n sor látható.

D (n,m) : Töröl n-től m-ig. Egyetlen sort n=m révén törölünk.

|                  |   |
|------------------|---|
| <b>M n,m</b>     | : Sorokat mozgathatunk át. Az n. sort az m-be viszi át. Az m elhagyható.  |
| <b>N (n,m)</b>   | : Átsorszámozza a sorokat. Az első sor sorszáma n, a lépésköz pedig m.  |
| <b>F n,m,f,s</b> | : Egy f stringet keres az n és m sorok között. Ha ilyet talál, akkor leáll, és az "E" (EDIT) paranccsal az f string az adott stringre cserélhető. A további keresés "F"-el elintézhető. |
| <b>E (n)</b>     | : Az n sorszámú sort a pufferba másolja át, ahol tetszés szerint módosítható, de szükség esetén ismét előhívható az eredeti sor is.   |

Az edit-nek több alparancsa is van, amelyet a CS-tel aktivizálhatunk. Pl. CS " " (Szóköz) esetén meghagyja az eredeti karaktert, azaz megnöveli a soron belüli pointert, egészen a sor végéig.

|              |  |
|--------------|--|
| <b>CS0</b>   | : A Basic-hez hasonlóan karaktert töröl.   |
| <b>CS8</b>   | : A szövegpointert egy tab-pozícióval nagyobbra állítja.   |
| <b>ENTER</b> | : A sor kész.  |
| <b>Q</b>     | : Kilép az edit-ből és törli a változtatásokat.  |
| <b>R</b>     | : Újra betölti az adott sort a pufferba, miközben a változtatásokat elfelejti.   |
| <b>L</b>     | : A sor hátralevő részét meghagyja és a pointert a sor elejére állítja.  |
| <b>K</b>     | : Törli az adott karaktert.  |
| <b>Z</b>     | : Törli az összes karaktert a pillanatnyi pozíciótól kezdve a sor végéig.  |
| <b>F</b>     | : A korábban megadott stringet keresi. Ha ilyet nem talál, akkor kilép az edit-módból.   |
| <b>S</b>     | : A korábban megadott stringet lecseréli a már megadott s stringre   |
| <b>I</b>     | : Az adott pozíciótól kezdve karaktereket szűr be. Kilépés 'ENTER'-rel ezen belül a BASIC-hez hasonlóan működik a CS0 és a CS8-is. |
| <b>X</b>     | : A sor végére állítja a pointert és a BASIC-hez hasonlóan szerkeszthető a sor.  |
| <b>O</b>     | : Felülírható az adott karakter. Kilépés 'ENTER'-rel, valamint a BASIC szerkesztő utasításai használhatók.                         |

#### A magnetofon-kezeléssel kapcsolatos parancsok:

|                |  |
|----------------|--|
| <b>P n,m,s</b> | : Szövegfile-t ment ki a szalagra az n-től kezdve m-ig folyamatosan, s néven   |
| <b>G s</b>     | : Egy s file-t keres a szalagon és betölti azt. Ha már van a memóriában szövegfile, akkor e mögé tölti be a szalagról, miközben a teljes szöveget átsorszámozza.           |
| <b>T n,m,s</b> | : A szövegfile-t blokkokban viszi ki az n sortól kezdve m-ig s néven. A blokk méretét a 'Buffer size?'-nál adtuk meg. A *F parancsnál olvassuk majd vissza a szövegfile-t. |

#### Az editor parancsai fordításhoz és futtatáshoz:

|          |   |
|----------|---|
| <b>A</b> | : Lefordítja a szövegfile-t.  |
| <b>R</b> | : Az ENT-nek megfelelő címtől futtatja a tárgykódot, amely RET-tel végződik |

#### Egyéb parancsok:

|                |   |
|----------------|---|
| <b>B</b>       | : Visszatérés a BASIC-hez.  |
| <b>C</b>       | : Lehetőség van a GENS1-ben írt programok konvertálására a GENS 3 alatt. Ekkor "G"-vel beolvassuk a GENS1-ben írt programot, majd a "C" paranccsal a GENS 3-nak megfelelő formára alakítjuk. Ettől kezdve már az ismert módon formálhatjuk a programot. |
| <b>d</b>       | : A delimiter (határolójel) megváltoztatható a d string első karakterével.  |
| <b>p</b>       | : Kinyomtatja az alapértelmezéseket, ahol is a sorszámok 1 - 32767 között lehetnek, míg a stringhosszak max. 20 karakteresek.   |
| <b>W (n,m)</b> | : Kinyomtatja az n és m közé eső sorokat. A K értékének megfelelően bizonyos számú sor kinyomtatása után szünetet tart, amit tetszőleges billentyű lenyomása után folytat   |

|          |   |
|----------|---|
| <b>X</b> | : Kinyomtatja a kezdő és a végcímet. Ekkor lehetőség van arra, hogy pl. MONS3 által létrehozott szövegfile-t a GENS3 editora átvegyen |
|----------|---|

A GENS3 által használt szövegfile-t az xxxxx + 54 és xxxxx + 55 végcímek határozzák meg, tehát ha egy szövegfile a RAM-ban van, majd betöltjük a GENS3-at és BASIC-ből kiadjuk a POKE xxxxx + 54, végcím (alsó byte) és POKE xxxxx + 55, végcím (felső byte) parancssort, akkor RANDOMIZE USR xxxxx nem jó, hanem RANDOMIZE USR xxxxx + 61 utasítással lesz a szövegfile a GENS3-nak megfelelő!

#### A GENS3 hibakódjai:

1. – a hiba ebben a sorban van
  2. – a mnemonikot nem ismeri fel
  3. – az utasítás rossz "formájú"
  4. – a szimbólumot többször is definiálták
  5. – a sorban illegális karakter van
  6. – az operandusok egyike illegális
  7. – a szimbólumként használt szó "foglalt"
  8. – tiltott regiszterhasználat
  9. – sok a regiszter
  10. – a 8 bites kifejezés többre sikerült mint 8 bit
  11. – a JP (IX + n) és JP (IY + n) tiltott
  12. – az assembler direktíva formája hibás
  13. – illegális hivatkozás, pl. az EQU-ban olyan vál-tozó szerepel, amely még nem kapott értéket
  14. – osztás zérussal
  15. – szorzásnál fellépő túlcsoordulás
- BAD ORG ! – hibás ORG cím fordult elő  
 Out of Table space! – a szimbólumtábla számára kevés a hely  
 Bad Memory! – nincs több hely a szövegfile részére

#### MONS3 MONITOR/DISASSEMBLER

A mi általunk tesztelt program hossza:

by: MONS3 (5760 bytes)

Mindenekelőtt töltsük be azt a programot, amit vizsgálni szeretnénk. Ezután töltsük be a MONS3 programot is:

LOAD""CODE xxxxx, ahol xxxxx tetszőleges, de a RAMTOP felé érdemes betölteni. A MONS3 indítása (feltételezett) első indításnál RANDOMIZE USR xxxxx, ismételt indításnál: RANDOMIZE USR (xxxxx + 29)  
 A programból való kilépés az EDIT funkcióval (CAPS SHIFT + 1) történik.

A program indításakor a képernyőn legfelül egyetlen kiragadott disassemblált sor látható. A sor elején a memória-mutató (pointer) értéke látható, amelyre mindegyik parancs hivatkozik. Alatta a regiszterek és a regisztereknek megfelelő értéktől kezdve HEXADECIMÁLIS dump (memorialista) látható a regiszterekkel egy sorban. Az AF regiszterpár alatt 8 sorban a memória-mutató környezetének HEXADECIMÁLIS dumpja látható.

Itt is a továbbiakban a SYMBOL SHIFT jelet SS-sel, A CAPS SHIFT-et pedig CS-sel fogjuk rövidíteni.

- SS + 4 (\$)** : Az MP (memória-pointer)-nek megfelelő címtől egy lapnyi disassemblált listát ad. Visszatérés ismét a SS + 4-gyel, továbbá bármelyik billentyűvel.
- ENTER** : MP = MP + 1
- CS + 7** : MP = MP - 1
- CS + 5** : MP = MP - 8
- CS + 8** : MP = MP + 8
- "," (vessző)** : MP = a nyíllal jelölt regiszter értéke.
- G** : Stringkereséshez. Ezután kétjegyű HEX számot adunk meg 'ENTER'-rel elválasztva. Kilépés szintén 'ENTER'-rel. Ismétlés kereséshez: "N" (NEXT).
- H** : Decimálisból HEXADECIMALISBA konvertál.
- I** : (Int.copy) blokkmásoláshoz
- J** : Az adott (MP-nek megfelelő) címtől indít egy assembly programot. Ez a parancs végrehajtása előtt törölhető a CS + 5-tel.
- SS + K** : A program végrehajtása a PC (Program Counter) értékének megfelelően folytatódik (a legközelebbi BREAK-ig). Ennek akkor lesz szerepe, amikor egy programot lépésenként futtatunk (ld. SS + Z parancsot), de egy bizonyos pontig szeretnénk "azonnal" lefuttatni az assembly utasításokat (ezt a pontot BREAK-vel, "W"-vel kellett beállítani), majd a ponttól kezdve ismét lépésenként futtatni a programot. Így a CALL utasításokat gyorsan kiadhatjuk.
- L** : Listáz egy blokkot az MP-nek megfelelően. Kilépés CS + 5-tel.
- M** : Az MP értékének tetszőleges beállítása.
- N** : Ld. a "G" parancsnál.
- O** : Relatív ugrás.
- P** : A memória egy részét egy megadott byte-tal tölti fel.
- Q** : Az EXX assembler utasításnak felel meg.
- SS + T** : BREAK-point-ot helyez el a jelenlegi utasítás mögé és a végrehajtást folytatja. Amennyiben a SS + Z érvényes, akkor a SS + 7 hatására pl. egy CALL utasítást azonnal végrehajtja, majd ismét "lépeget".
- T** : Disassemblál egy blokknyi memóriát és esetleg nyomtatóra küldi. Először a disassemblálendő szöveg kezdő és végcímét adjuk meg. Ezután a 'Text:' kérdésnél vagy HEX. címet adunk meg, vagy 'ENTER'-t ütünk. Ha egy címet

adunk, akkor ettől a címtől kezdve egy assembly forrásszöveget állít elő a disassemblálás alapján. Ezt a szövegfile-t adhatjuk át azután a GENS3 nak. E szövegfile előállítás után kírhatja a végcímet is. E két cím birtokában beállíthatjuk a GENS3 ban a kezdőcímet az "X" alparanccsal, ill. ismerjük a végcímet is. Ekkor a "Workspace:"-re egy címet vár, amelyre egy "primitív" szimbólumtáblát tesz le. Elég 'ENTER'-rel válaszolni. Ha a "Text:"-re 'ENTER'-rel válaszolunk, akkor "csak" a képernyőn jelenik meg a disassemblált szöveg. Ezután ismét egy kezdő- és egy végcímet vár. Most kijelölhetünk a korábban kijelölt blokkon belül egy részblokkot, amelyet a MONS3 nem Z80-as utasításnak tekint, hanem konstansterületnek, és DEFB utasításként kezel. A "T"-ből CS + 5-tel lehet kilépni. Amennyiben az előállított szövegben olyan ugrás szerepel, amely a kijelölt blokkba mutat, akkor az abszolút cím elé "L" betűt helyez el. Amennyiben ismeretlen utasítással találkozunk, akkor egy NOP és csillagjel következik. Figyelem! Ha a forrásszövegben BREAK fordul elő (ld. "W"-t), akkor lehet hogy kettévágtuk az assemblerutasítást, és lehet, hogy a disassemblált szöveg "hamis".

- U** : Az "O" utasítással együtt létezhet. Amíg az "O" egy relatív ugrást hajt végre, addig az "U" visszatérést jelent. Ha több "O" is előfordul, akkor csak a legutoljára kiadott "O"-ra ugrik vissza, és a többi "O"-t nem veszi figyelembe.
- V** : Az "X" parancs fordítottjának tekinthető. Amíg az "X" egy abszolút ugrást hajt végre, addig a "V" visszatérést jelent. Csak a legutoljára kiadott "X"-nek megfelelően ugrik vissza, míg a többi "X" elveszik.
- W** : BREAK-point-ot helyez el a forrásszövegben. E címtől 3 byte-ot lecserél egy CALL utasításra.
- X** : Egy abszolút CALL vagy JP utasításnak megfelelő címre ugrik el.
- Y** : Az MP-nek megfelelő címtől ASCII karaktereket írhatunk be. Kilépés CS + 5-tel.
- SS + Z** : Egy lépést hajt végre az MP-től függően (lépésenkénti végrehajtásnál láthatók a regiszterek is).
- SS + P** : Nyomtatóra listáz.

## QUARX • Your Sinclair

A pontszám-táblába írjuk be: HI GREG, majd nyomjuk meg a <Z>, <S>, <D>, <G>, <H>, és a <B> billentyűket együtt. Ekkor kapunk 8 életet, és átkerülünk a következő szintre.