

Glibc 2 HOGYAN

Eric Green, ejg3@cornell.edu

v1.6, 1998. június 22.

A glibc 2 HOGYAN leírja, hogyan telepítsd és használd a GNU C programkönyvtár 2-es verzióját (libc 6) a Linux operációs rendszeren.

Contents

1	Bevezetés	2
1.1	Amit a glibc 2-ről tudni kell	2
1.2	Erről a dokumentumról	3
1.3	Mi változott a közelmúltban?	3
2	Telepítési módszer kiválasztása	3
3	A könyvtár beszerzése	4
4	Telepítés teszt könyvtárként	5
4.1	Fordítás és telepítés	5
4.1.1	Telepítési követelmények	5
4.1.2	A forrás kicsomagolása	5
4.1.3	Beállítás	6
4.1.4	Fordítás és telepítés	6
4.2	A dinamikus betöltő (dynamic loader) frissítése	6
4.3	Beállítás a gcc fordítóhoz	7
4.4	A fejlécfájlokra mutató kapcsok frissítése	7
4.5	A telepített programkönyvtár tesztelése	7
5	Telepítés elsődleges C könyvtárként	8
5.1	A könyvtár fordítása forráskódból	8
5.1.1	Telepítési követelmények	8
5.1.2	A forrás kicsomagolása	9
5.1.3	Beállítás	9
5.1.4	Fordítás	9
5.2	Felkészülés a telepítésre	9
5.3	Telepítés bináris csomagból	11
5.4	Telepítés forráskódból	11
5.5	A gcc specs fájljának frissítése	11
5.6	A feltelepített programkönyvtár tesztelése	12

6 Fordítás a nem elsődleges libc-vel	12
6.1 Figyelmeztetés, ha nem elsődleges C programkönyvtárat használsz	12
6.2 Programok fordítása teszt glibc-vel	13
6.3 Programok fordítása libc 5-el, ha a glibc az elsődleges programkönyvtár	13
7 C++ programok fordítása	14
7.1 A libg++ telepítése teszt glibc könyvtárral	14
7.2 A libg++ telepítése elsődleges glibc könyvtár esetén	14
7.3 C++ programok fordítása a nem elsődleges libc könyvtárral	14
8 Problémák	15
8.1 Nem működik a gépnevek feloldása	15
9 Hibák bejelentése	15
10 Példa specs fájl	16
11 Egyéb	17
11.1 További információ	17
11.1.1 Web lapok	17
11.1.2 Hírcsoportok	17
11.1.3 Levelezőlisták	17
11.2 Köszönet	18
11.3 Visszajelzés	18
11.4 Szerzői jogok	19
11.5 Magyar fordítás	19

1 Bevezetés

1.1 Amit a glibc 2-ről tudni kell

A Glibc 2 a GNU C programkönyvtár legújabb változata. Jelenleg módosítás nélkül fut GNU Hurd rendszereken és a Linux i386, mk68k és alpha rendszereken. A 2.1-es verzió támogatni fogja a Linux PowerPc-t valamint a MIPS, Sparc, Sparc-64 és Arm platformokat. A könyvtár a jövőben más architektúrákat és operációs rendszereket is támogatni fog.

A Linux rendszereken a glibc 2, a 6-os verziószámú libc-vel egyezik meg, amely a libc 5 utódja. A Linuxos libc fejlesztői a libc 6-al fel akarják váltani a libc 5-öt. A 2.0.6-os verziótól kezdve, a libc megbízhatónak tekinthető. A 2.1-es változat, amely hamarosan napvilágot lát, készen áll majd a széleskörű felhasználásra, és további platformokat is fog támogatni.

Három opcionális kiegészítő létezik a glibc 2-höz:

Crypt

Az UFC-crypt csomag. Ezt a csomagot nem integrálták a glibc-be, az exporttilalmak miatt.

LinuxThreads

A Posix 1003.1c "pthread" interfész egy implemetációja.

Locale data

Ez tartalmazza a locale adatfájlok előállításához szükséges adatokat, ezekre akkor van szükség, ha használni akarsz a glibc nemzetközi lehetőségeit.

A crypt és LinuxThreads kiegészítő csomagok erősen ajánlottak... Ha nem használod őket, inkompatibilitási problémák léphetnek fel más rendszerek különböző könyvtáraival. (Ha ezek után mégsem akarsz használni őket, meg kell adnod a `-disable-sanity-checks` opciót a configure szkript futtatásánál.)

1.2 Erről a dokumentumról

Ez a HOGYAN arról szól, hogyan telepítsd a glibc 2 könyvtárat egy meglévő Linux rendszerre. A leírás abból indul ki, hogy libc 5-öt használasz i386-os architektúrára, de más rendszerek és könyvtárak (pl. glibc1) felhasználói is hasznát vehetik, csak le kell cserélni a megfelelő fájl- és architektúra-neveket a szituációnak megfelelően.

Eme HOGYAN legfrissebb változatát megtalálod a

Linux Documentation Project <<http://sunsite.unc.edu/LDP>> részeként vagy a következő kiszolgálón:
<<http://www.imaxx.net/~thrytis/glibc/Glibc2-HOWTO.html>> .

1.3 Mi változott a közelmúltban?

Különbségek a dokumentum 1.6-os és 1.5-os változatai között:

- Kijavítottam a bináris glibc csomag telepítésére vonatkozó utasításokat.

Különbségek a dokumentum 1.5-os és 1.4-es változatai között:

- Indexelés hozzáadása, köszönet Ed Bailey-nek.
- Megváltoztattam az e-mail címemet.

2 Telepítési módszer kiválasztása

A glibc-t többféleképpen telepítheted. Feltelepítheted tesztelés céljára, ebben az esetben a régi könyvtárakat használod továbbra is alapértelmezésben, de ha fordítasz egy programot, megadhatod, hogy az használja a teszt könyvtárat. Ha ezt a telepítési módszert választod, később könnyen eltávolíthatod a glibc-t a gépedről (persze a hozzá kapcsolódó programok nem fognak működni). Ha ilyen módon tesztelni akarsz a glibc-t, forrásból kell fordítsd, mert nem készült tesztelési célból bináris csomag (pl. rpm). Erről a megoldásról a [4](#) (Telepítés teszt könyvtárként) fejezetben olvashatsz.

Ezzel ellentétben, telepítheted a glibc-t elsődleges programkönyvtárként is. Ilyenkor minden új program, amit a gépen fordítasz, alapértelmezésben a glibc-t fogja használni, kivétel persze, ha fordításkor a régi könyvtárat adod meg. Fordíthatod a könyvtárat magad, vagy feltelepítheted bináris csomagból is. Ha változtatni akarsz a glibc beállításain, az optimalizálással akarsz variálni, vagy olyan kiegészítőt akarsz használni, ami a bináris

csomagban nincsen benne, le kell töltened a forráskódot, és le kell fordítanod a könyvtárat. Ez esetben a telepítési útmutatót megtalálod a 5 (Telepítés elsődleges C könyvtárként) fejezetben.

Frodo Looijaard leír egy harmadik telepítési módszert. Az ő módszerének lényege, hogy a glibc-t másodlagos könyvtárként telepíted, és beállítasz egy keresztfordítót (cross compiler), ami a glibc-t használja. Ez a telepítési módszer kicsit bonyolultabb, mint a tesztkönyvtárként telepítés, de utána egyszerűbb olyan programokat fordítani, amik a glibc-t használják. Ennek a módszernek a leírását a

A glibc-2 telepítése Linuxra <<http://huizen.dds.nl/~frodol/glibc/>> című dokumentumba találsz.

Ha 1.3-as Debiánt használasz, és nem akarsz a glibc miatt az "unstable" változatot használni, olvasd el a

Debian libc5 to libc6 Mini-HOWTO-t <<http://www.gate.net/~storm/FAQ/libc5-libc6-Mini-HOWTO.html>>, ez leírja, hogyan frissítsd Debian csomagokkal rendszeredet.

Ha a glibc 2-t egy fontos gépre telepíted, valószínűleg a teszt telepítés a jobb választás. Még ha a glibc-ben nincsenek is hibák, egyes programokat esetleg módosítani kell, hogy le tud fordítani őket, mert változhatnak a könyvtárban a függvények prototípusai és típusai.

3 A könyvtár beszerzése

A glibc 2 négy csomagból áll: a glibc csomagból és a három kiegészítő csomagból, ezek: LinuxThreads, Locale és Crypt. A forrásokat letöltheted a következő helyekről:

- <<ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.6.tar.gz>>
- <<ftp://prep.ai.mit.edu/pub/gnu/glibc-linuxthreads-2.0.6.tar.gz>>
- <<ftp://prep.ai.mit.edu/pub/gnu/glibc-localedata-2.0.6.tar.gz>>
- <<ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.6.tar.gz>>

A teljes glibc fordításához és telepítéséhez kb. 150 MB lemezterületre lesz szükséged. Ha bináris csomagból telepíted az alap glibc-t, akkor kb. 50MB-al kell számolnod.

A 2.0.6-os változathoz nincsenek bináris csomagok. A 2.0.4-es verzió bináris csomagjai i386-os és m68k architektúrákra, valamint a 2.0.1-es változat alphára megtalálható itt:

- Intel x86:
 - <<ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.4.bin.i386.tar.gz>>
 - <<ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.4.bin.i386.tar.gz>>
- Alpha:
 - <<ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.1.bin.alpha-linux.tar.gz>>
 - <<ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.1.bin.alpha-linux.tar.gz>>
- m68k:
 - <<ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.4-m68k-linux.bin.tar.gz>>
 - <<ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.4-m68k-linux.bin.tar.gz>>

A Crypt kiegészítő csomagra exportkorlátozások vannak érvényben. Az Egyesült Államokon kívül élő felhasználók ezt a csomagot a

<<ftp://ftp.ifi.uio.no/pub/gnu>> kiszolgálóról szerezhetik be.

Ha Red Hat összeállítást használsz, letöltheted a 2-es glibc-t rpm csomagokban a Red Hat ftp kiszolgálójáról: [<ftp://ftp.redhat.com/pub/redhat/>](ftp://ftp.redhat.com/pub/redhat/) . A Red Hat a glibc 2 könyvtárat használja elsődleges C könyvtárként az összeállítás 5.0-as verziója óta.

Ha Debiant használsz, letöltheted a 2-es glibc-t deb csomagokban a Debiantól

[<ftp://ftp.debian.org/debian/dists/unstable/main/>](ftp://ftp.debian.org/debian/dists/unstable/main/) . A fájlokat libc6-nak hívják. A glibc 2 része a Debian hamm verziójának, elsődleges C könyvtár lesz a Debian 2.0 megjelenésétől.

4 Telepítés teszt könyvtárként

Ez a rész azt írja le, a glibc 2 telepítését teszt könyvtárként. Ha bármilyen programot fordítasz, azt a fordító a régi glibc-vel szerkeszti össze, hacsak nem adod meg külön, hogy a teszt könyvtárat szeretnéd használni. Fordításkor az elérési útvonalak belekerülnek néhány fájlba, ezért valószínűleg forrásból kell telepítsd a könyvtárat.

4.1 Fordítás és telepítés

4.1.1 Telepítési követelmények

- Kb. 150 MB üres lemezterület
- GNU make 3.75
- gcc \geq 2.7.2 (még jobb, ha 2.7.2.1)
- binutils 2.8.1 (alphához egy snapshot-ra van szükség)
- bash 2.0
- autoconf 2.12 (ha megváltoztatod a configure.in fájlt)
- texinfo 3.11

Egy i586@133 gépen 64 MB RAM-mal, kb. 3 órába telik, amíg elkészül a teljes könyvtár a kiegészítőkkal. Egy erősen leterhelt i686@200 gépen ugyanez kb. fél órába kerül.

4.1.2 A forrás kicsomagolása

Ahhoz, hogy le tud fordítani a forrást, ki kell csomagolnod. Ezt legkönnyebben a következő módon teheted meg:

```
tar xzf glibc-2.0.6.tar.gz
cd glibc-2.0.6
tar xzf ../glibc-linuxthreads-2.0.6.tar.gz
tar xzf ../glibc-crypt-2.0.6.tar.gz
tar xzf ../glibc-localedata-2.0.6.tar.gz
```

Ez a glibc-2.0.6 könyvtárba csomagolja ki a linuxthreads, crypt és locale kiegészítőket, ahol a configure megtalálja őket.

4.1.3 Beállítás

A glibc-2.0.6 könyvtárba hozzá létre egy compile nevű könyvtárat, majd lépj bele. Mindent ebben a könyvtárban fogsz csinálni, ez egyszerűsíti a takarítást. (A fejlesztők eddig nem törték magukat, hogy a 'make clean' tökéletesen működjön.)

```
mkdir compile
cd compile
```

Futtasd le a `./configure` szkriptet. Ha kiegészítő csomagokat is akarsz használni, meg kell adnod a `-enable-add-ons` kapcsolót, pl.: `-enable-add-ons=linuxthreads,crypt,localedata`. Ki kell válaszdnod a célkönyvtárat, ahova a könyvtárat telepíteni fogod. Erre a `/usr/i486-linuxglibc2` egy jó választás. Ehhez a `configure` paraméterei a következők:

```
./configure --enable-add-ons=linuxthreads,crypt,localedata --prefix=/usr/i486-linuxglibc2
```

4.1.4 Fordítás és telepítés

A fordításhoz és ellenőrzéshez írd be a következőket:

```
make
make check
```

Ha a 'make check' sikeresen lefutott, telepítsd a könyvtárat root felhasználóként (még mindig a `compile` könyvtárból):

```
make install
```

4.2 A dinamikus betöltő (dynamic loader) frissítése

1. Hozzá létre egy kötetet, ami az új `ld.so` -ról a `/lib/ld-linux.so.2` -ra mutat:

```
ln -s /usr/i486-linuxglibc2/lib/ld-linux.so.2 /lib/ld-linux.so.2
```

Ez az egyetlen könyvtár, amelynek rögzített a helye, miután egy programhoz hozzá lett kapcsolva. Ha a `/lib` könyvtárban létrehozol egy kapcsolót (link) hozzá, akkor később könnyebb lesz áttérni a glibc-re, mint elsődleges C könyvtárra, amikor megjelenik a stabil verziója.

2. Szerkeszd a `/etc/ld.so.conf` fájlt. A fájl végére be kell írnod a lib könyvtár elérési útvonalát, amiben az új programkönyvtárak vannak. Valahogy így fog kinézni: `<prefix>/lib`, tehát pl. `/usr/i486-linuxglibc2/lib` a fenti esetben. A `/etc/ld.so.conf` fájl elmentése után futtasd le a következő parancsot:

```
ldconfig -v
```

4.3 Beállítás a gcc fordítóhoz

A telepítés utolsó lépése a `/usr/lib/gcc-lib` frissítéséből áll. Ez azért szükséges, hogy a gcc tudja használni az új programkönyvtárakat. Első körben duplikálni kell a gcc-t. Hogy megtudd melyik az éppen működő gcc konfiguráció, használd a gcc `-v` kapcsolóját:

```
% gcc -v
Reading specs from /usr/lib/gcc-lib/i486-unknown-linux/2.7.2.2/specs
gcc version 2.7.2.2
```

Ebben az esetben az `i486-unknown-linux` a rendszer típusa, és a `2.7.2.2` a gcc verziószáma. Át kell másolnod a `/usr/lib/gcc-lib/<system>` könyvtárat az új tesztrendszer könyvtárába:

```
cd /usr/lib/gcc-lib/
cp -r i486-unknown-linux i486-linuxglibc2
```

Lépj be az új tesztrendszer könyvtárába, majd a verziószámmal jelölt könyvtárba,

```
cd /usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2
```

ezek után módosítsd a `specs` fájlt, ami ebben a könyvtárban található. Írd át a `/lib/ld-linux.so.1` szöveget `/lib/ld-linux.so.2`-re. Ha ez megvan, töröld ki minden `%{...:-lgmon}` fájlra mutató hivatkozást, mert a glibc nem használja a gmon könyvtárat. Találsz egy példa spec fájlt a **10** (Példa specs fájl) fejezetben.

4.4 A fejlécfájlokra mutató kapcsok frissítése

Kapcsokat (link) kell létrehoznod az új include könyvtárban, amik egyéb include könyvtárakra mutatnak:

```
cd /usr/i486-linuxglibc2/include
ln -s /usr/src/linux/include/linux
ln -s /usr/src/linux/include/asm
ln -s /usr/X11R6/include/X11
```

Előfordulhat, hogy vannak más programkönyvtáraid, mint pl. az ncurses, ezeknek a fejlécfájlokat is ebbe a könyvtárba kell helyezni. Ezeket a fájlokat át kell másolnod vagy kapcsoznod (link) a `/usr/include` könyvtárból. (Lehetnek olyan programkönyvtárak is, amiket újra kell fordítanod, hogy a glibc2-vel működni tudjanak. Ebben az esetben egyszerűen fordítsd le őket, és a telepítésnél add meg a `/usr/i486-linuxglibc2` könyvtárat.)

4.5 A telepített programkönyvtár tesztelése

A programkönyvtár teszteléséhez, hozz létre egy `glibc.c` fájlba következőkkel:

```
#include <stdio.h>

main()
{
    printf("hello world!\n");
}
```

Fordítsd le a következő opciókkal: "-b <base install directory> -nostdinc -I<install directory>/include -I/usr/lib/gcc-lib/<new system dir>/<gcc version>/include":

```
% gcc -b i486-linuxglibc2 -nostdinc -I/usr/i486-linuxglibc2/include \  
-I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include glibc.c -o glibc
```

Az ldd parancs segítségével ellenőrizd, hogy a fordító tényleg a glibc2-vel szerkesztette-e össze a programcskát és nem a régi libc-vel:

```
% ldd glibc  
libc.so.6 => /usr/i486-linuxglibc2/lib/libc-2.0.6.so (0x4000d000)  
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Ha a programot le tudtad fordítani, a gcc az új glibc2-höz szerkesztette és amikor elindítod, azt írja ki, hogy "hello world!", a telepítés sikeres volt.

5 Telepítés elsődleges C könyvtárként

Ez a rész arról szól, hogy hogyan telepítsd a glibc 2-t elsődleges C könyvtárként. Ezek után minden program amit a rendszereden fordítasz az új C programkönyvtárat fogja használni, hacsak nem beszéled le erről speciális fordító kapcsolók segítségével.

Ha RedHat vagy Debian Linuxot használsz és letöltötted a megfelelő rpm vagy deb csomagokat, átugorhatod ezt a részt, nézd meg a RedHat vagy Debian telepítési útmutatót.

5.1 A könyvtár fordítása forráskódból

Ez a rész leírja, hogy hogyan fordítsd le a glibc 2-t és a kiegészítőit a forrásból. Akkor kell fordítanod a glibc-t, ha változtatni akarsz az alapbeállításain vagy az optimalizáláson, vagy olyan kiegészítőt akarsz használni, amihez nincsen bináris csomag.

5.1.1 Telepítési követelmények

- Kb. 150 MB üres hely a lemezen
- GNU make 3.75
- gcc >= 2.7.2 (még jobb, ha 2.7.2.1)
- binutils 2.8.1 (az alphához kell egy snapshot)
- bash 2.0
- autoconf 2.12 (ha megváltoztatod a configure.in fájlt)
- texinfo 3.11

Egy i586@133 gépen 64 MB RAM-mal, kb. 3 órába telik, amíg elkészül a teljes könyvtár a kiegészítőkkal. Egy erősen leterhelt i686@200 gépen ugyanez kb. fél órába kerül.

5.1.2 A forrás kicsomagolása

Ahhoz, hogy le tudd fordítani a forrást, ki kell csomagolnod. Erre a legjobb megoldás a következő:

```
tar xzf glibc-2.0.6.tar.gz
cd glibc-2.0.6
tar xzf ../glibc-linuxthreads-2.0.6.tar.gz
tar xzf ../glibc-crypt-2.0.6.tar.gz
tar xzf ../glibc-localedata-2.0.6.tar.gz
```

Ez a glibc-2.0.6 könyvtárba csomagolja ki a linuxthreads, crypt és locale kiegészítőket, ahol a configure megtalálja őket.

5.1.3 Beállítás

A glibc-2.0.6 könyvtárba hozzáál létre egy compile nevű könyvtárat majd lépj bele. Mindent ebben a könyvtárban fogsz csinálni, ez egyszerűsíti a takarítást. (A fejlesztők eddig nem törték magukat, hogy a 'make clean' tökéletesen működjön.)

```
mkdir compile
cd compile
```

Futtasd le a `./configure` szkriptet. Ha kiegészítő csomagokat is akarsz használni, meg kell adnod az `-enable-add-ons` kapcsolót, pl.: `-enable-add-on\s=linuxthreads,crypt,localedata`.

Valószínűleg meg akarod adni a célkönyvtárat is, ahova a glibc-t telepíteni akarod. Mivel a szabványos Linux disztribúcióhoz akarsz igazodni, add meg a `-prefix=/usr` kapcsolót. (Ha egy Linux rendszeren az `/usr` könyvtárat adod meg, a configure magától be tud állítani egyéb fontos elérési útvonalakat, ahova a `libc.so` és egyéb más fontos a `/lib`-ben lévő könyvtárakat kell tenni. A configure parancssora valahogy így fog kinézni:

```
./configure --enable-add-ons=linuxthreads,crypt,localedata --prefix=/usr
```

5.1.4 Fordítás

A fordításhoz és ellenőrzéshez írd be a következőket:

```
make
make check
```

5.2 Felkészülés a telepítésre

Akár fordítod a glibc-t, akár bináris csomagokból rakod fel, néhány fájlt át kell helyezned, mert bár minden újonnan fordított program az új glibc-t fogja használni, a régi (nem statikusan linkelt) programoknak szükségük van a régi programkönyvtárra (`libc5`), ezért azt nem írhatod egyszerűen felül.

1. Hozz létre egy új könyvtárat a régi fájloknak:

```
mkdir -p /usr/i486-linuxlibc5/lib
```

2. A régi fejléc fájlokat ki kell helyezni a `/usr/include` könyvtárból:

```
mv /usr/include /usr/i486-linuxlibc5/include
```

3. Csinálj egy új include könyvtárat és hozd létre a szükséges, más include könyvtárakra mutató kapcsokat:

```
mkdir /usr/include

ln -s /usr/src/linux/include/linux /usr/include/linux
ln -s /usr/src/linux/include/asm /usr/include/asm
ln -s /usr/X11R6/include/X11 /usr/include/X11
ln -s /usr/lib/g++-include /usr/include/g++
```

Ezek a kapcsok lehet, hogy kicsit eltérnek a megadottól, mert az include könyvtárak helye összeállítástól függő. Legalább is a Slackware Linuxokon a g++ fejléc fájlok a `/usr/local/g++-include` könyvtárban vannak, míg a Debian alatt a `/usr/include/g++` könyvtárban található és a `/usr/lib/g++-include` kapocs a `/usr/include/g++-ra` mutat. Az utóbbi esetben valószínűleg szerencsésebb az eredeti g++ include könyvtárat visszahelyezni a `/usr/include` könyvtárba.

4. Állíts vissza minden plusz fejléc fájlt és kapcsot, ami esetleg az eredeti `/usr/include` könyvtárban volt. Egyes szabványtalan programkönyvtárak, mint pl. az ncurses fájlokat raknak be a `/usr/include` könyvtárba, vagy kapcsokat hoznak ott létre, melyek az adott programkönyvtár include könyvtárára mutatnak. Ezeket a fájlokat és kapcsokat vissza kell állítanod, hogy továbbra is használni tudd a kérdéses programkönyvtárakat.
5. Add hozzá az új programkönyvtár könyvtárát (pl. `/usr/i486-linuxlibc5/lib`) az `/etc/ld.so.conf` fájl *elejéhez*. Legalább az 1.8.8-as verziójú ld.so-ra van szükséged, különben bizarr hibáüzenetekkel találod magad szembe a glibc telepítése után.

6. Helyezd át az összes régi C programkönyvtárat az új könyvtárba.

```
mv /usr/lib/libbsd.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libc.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libgmon.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libm.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libmcheck.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libc.so /usr/i486-linuxlibc5/lib
mv /usr/lib/libm.so /usr/i486-linuxlibc5/lib
cp /lib/libm.so.5.* /usr/i486-linuxlibc5/lib
cp /lib/libc.so.5.* /usr/i486-linuxlibc5/lib
```

A `libm.so.5` és `libc.so.5` fájlokat másolni kell és nem áthelyezni, ha a `/usr` nem a `/` partíción van, mert ezekre a programkönyvtárakra a Linuxnak induláskor szüksége van, ezért a `/` köteten kell legyennek.

7. Helyezd át az `/usr/lib/*.o` fájlokat az új könyvtárba.

```
mv /usr/lib/crt1.o /usr/i486-linuxlibc5/lib
mv /usr/lib/crti.o /usr/i486-linuxlibc5/lib
mv /usr/lib/crtn.o /usr/i486-linuxlibc5/lib
mv /usr/lib/gcrt1.o /usr/i486-linuxlibc5/lib
```

8. Frissítsd a programkönyvtár gyorsítótárát, miután átrendezted a programkönyvtárakat.

```
ldconfig -v
```

5.3 Telepítés bináris csomagból

Ha előre fordított bináris csomagokból telepíted a glibc-t, először ellenőrizd a csomagok tartalmát:

```
tar -tzvf glibc-2.0.bin.i386.tar.gz
tar -tzvf glibc-crypt-2.0.bin.i386.tar.gz
```

Ha elégedett vagy az eredménnyel, telepítheted a glibc-t:

```
cd /
tar -xzf glibc-2.0.bin.i386.tar.gz
tar -xzf glibc-crypt-2.0.bin.i386.tar.gz
ldconfig -v
```

Ha más architektúrát vagy verziót használsz, értelemszerűen helyettesítsd a fájlneveket.

A glibc legújabb verziója általában nincs meg bináris csomagban, viszont a legújabb verzió használata mindig javasolt, a hibajavítások miatt. Ha nem tudod a glibc-t lefordítani a saját gépeden, tölts le egy bináris programkönyvtárat tartalmazó csomagot egy olyan disztribúcióból, ami a glibc-t használja (pl. RedHat) és telepítsd azt.

5.4 Telepítés forráskódból

A programkönyvtár forráskódból való telepítéséhez, add ki root felhasználóként a `compile/` könyvtárból a következő utasításokat:

```
make install
ldconfig -v
```

5.5 A gcc specs fájljának frissítése

A telepítés utolsó lépéseként (mind a bináris csomagok, mind a forráskódból való telepítés esetén) frissíteni kell a `gcc specs` fájlját, hogy a fordító az új glibc-vel szerkessze össze a későbbiekben fordított programokat. Hogy meghatározd, a gcc melyik specs fájl használja, add ki a következő parancsot:

```
% gcc -v
reading specs from /usr/lib/gcc-lib/i486-unknown-linux/2.7.2.2/specs
gcc version 2.7.2.2
```

Ebben az esetben a `i486-unknown-linux` írja le a rendszert és a `2.7.2.2` a verziót. Át kell másolnod a `/usr/lib/gcc-lib/<system>` könyvtárat a régi rendszer könyvtárba:

```
cd /usr/lib/gcc-lib/
cp -r i486-unknown-linux i486-linuxlibc5
```

Lépjél be az eredeti könyvtárba, majd a verziószám által jelölt könyvtárba:

```
cd /usr/lib/gcc-lib/i486-unknown-linux/2.7.2.2
```

Ezek után módosítsd a `specs` fájlt, ami ebben a könyvtárban található. Írd át a `/lib/ld-linux.so.1` szöveget `/lib/ld-linux.so.2`-re. Ha ez megvan, töröld ki minden `%{...:-lgmon}` fájlra mutató hivatkozást, mert a `glibc` nem használja a `gmon` könyvtárat. Találsz egy példa spec fájlt a 10 (Példa specs fájl) fejezetben.

5.6 A feltelepített programkönyvtár tesztelése

A programkönyvtár ellenőrzéséhez, hozz létre egy `glibc.c` nevű fájlt a következő tartalommal:

```
#include <stdio.h>

main()
{
    printf("hello world!\n");
}
```

Fordítsd le a programot.

```
% gcc glibc.c -o glibc
```

Az `ldd` programmal ellenőrizd, hogy a fordító tényleg az új programkönyvtárat használja:

```
% ldd glibc
libc.so.6 => /lib/libc.so.6 (0x4000e000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Ha a programot sikerül lefordítani, és futása eredményeként kiírja, hogy "hello world!" a telepítés sikerült.

6 Fordítás a nem elsődleges libc-vel

Vannak olyan alkalmak, amikor nem az elsődleges programkönyvtárat szeretnéd használni fordításkor. Ez a rész azt írja le, hogy ezt hogyan teheted meg. A felhasznált könyvtár- és fájlnevek megegyeznek az előző részben használtakkal. Ne felejtse el megváltoztatni ezeket a neveket a telepítésednek megfelelően.

6.1 Figyelmeztetés, ha nem elsődleges C programkönyvtárat használsz

Mielőtt bármilyen olyan programot fordítanál, amit a Linux a rendszer indulásakor használ (a nem root partíciók felcsatolása előtt), ne felejtse el, hogy ha a program dinamikusan linkelt, szüksége van a programkönyvtárakra a root partíción. Ha megfigyelted az előző részben leírt telepítést, hiába a `glibc` az új C programkönyvtár, a régi `libc` is megmaradt a `/lib` könyvtárban, ami a root partíción van. Ez azt jelenti, hogy a régi `libc`-vel összeszerkesztett programok továbbra is használhatók az indulási folyamat során. Ha viszont a `/usr` nem a root partíción van és a `glibc`-t teszt könyvtárként telepítetted a `/usr/i486-linuxglibc2` könyvtárba, minden új, a teszt könyvtárral fordított program működésképtelen lesz, amíg nincs felcsatolva a `/usr` könyvtárat tartalmazó partíció.

6.2 Programok fordítása teszt glibc-vel

Ahhoz, hogy egy programot a teszt glibc könyvtárral fordíts, be kell állítanod az include könyvtárak és fájlok elérési útvonalait úgy, hogy azok a teszt glibc könyvtárban lévő include fájlokra mutassanak. Ha a parancssoron megadod a "-nostdinc" kapcsolót, az törli a hivatkozásokat az alapértelmezett include elérési útvonalakat, a "-I/usr/i486-linuxglibc2/include" pedig megadja a glibc include fájlok helyét. Meg kell még add a gcc include fájljait, amelyek a /usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include könyvtárban találhatóak (akkor, ha 2.7.2.2-es gcc-t használsz és a teszt programkönyvtárat az i486-linuxglibc2 könyvtárba telepítetted).

Meg kell add ezeken kívül a gcc beállításait. Ezt megteheted a "-b i486-linuxglibc2" kapcsolóval.

A legtöbb programnál megadhatod ezeket a kapcsolókat a makefile \$CFLAGS és \$LDFLAGS opcióinál:

```
CFLAGS = -nostdinc -I/usr/i486-linuxglibc2/include \
         -I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include \
         -b i486-linuxglibc2
LDFLAGS = -b i486-linuxglibc2
```

Ha configure szkriptet használsz a fordítás előtt, állítsd be a \$CFLAGS és \$LDFLAGS környezeti változókat (csh/tcsh-ban használd az env/setenv parancsokat, sh/bash-ban használd a set/export-ot) a configure futtatása előtt. Az így generált makefile fájlokban szerepelnie kell a megfelelő \$CFLAGS és \$LDFLAGS beállításoknak. Nem minden configure szkript olvassa ki a parancsértelmező környezetéből a változókat, ezért ellenőrizni kell a makefile-t, és kézzel javítani szükség esetén.

Ha a fordítandó program csak a gcc-t hívja meg (és nem egyenesen a cpp-t vagy a binutils-t), használhatod a következő szkriptet a kapcsolók megadására:

```
#!/bin/bash
/usr/bin/gcc -b i486-linuxglibc2 -nostdinc \
            -I/usr/i486-linuxglibc2/include \
            -I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include "$@"
```

Használd ezt a szkriptet fordításnál a "gcc" helyett.

6.3 Programok fordítása libc 5-el, ha a glibc az elsődleges programkönyvtár

Ha a régi libc könyvtárral akarsz programot fordítani és a glibc-t elsődleges könyvtárként telepítetted, vissza kell állítsad az include fájlok elérési útvonalát a régi include fájlokéra. Meg kell add a "-nostdinc" kapcsolót, ez törli az alapértelmezett beállításokat, majd a "-I/usr/i486-linuxlibc5/include" kapcsolóval be kell állítanod a libc5 include fájljainak az elérési útvonalát. Meg kell még adnod a "-I/usr/lib/gcc-lib/i486-linuxlibc5/2.7.2.2/include" kapcsolót, amely a gcc specifikus include fájlok helyét adja meg. Ne felejtse el, az itt megadott könyvtárneveket a saját rendszerednek megfelelően megváltoztatni (könyvtárnevek és gcc verzió).

Végül meg kell add a gcc beállításait, a "-b i486-linuxlibc5" kapcsolóval.

A legtöbb programnál megadhatod ezeket a kapcsolókat a makefile \$CFLAGS és \$LDFLAGS opcióinál:

```
CFLAGS = -nostdinc -I/usr/i486-linuxlibc5/include \
         -I/usr/lib/gcc-lib/i486-linuxlibc5/2.7.2.2/include \
         -b i486-linuxlibc5
LDFLAGS = -b i486-linuxlibc5
```

Ha configure szkriptet használsz a fordítás előtt, állítsd be a `$CFLAGS` és `$LDFLAGS` környezeti változókat (csh/tcsh-ban használd az `env/setenv` parancsokat, sh/bash-ban használd a `set/export-ot`) a configure futtatása előtt. Az így generált makefile fájlokban szerepelnie kell a megfelelő `$CFLAGS` és `$LDFLAGS` beállításoknak. Nem minden configure szkript olvassa ki a parancsértelmező környezetéből a változókat, ezért ellenőrizni kell a makefile-t, és kézzel javítani szükség esetén.

Ha a fordítandó program csak a gcc-t hívja meg (és nem egyenesen a cpp-t vagy a binutils-t), használhatod a következő szkriptet a kapcsolók megadására:

```
#!/bin/bash
/usr/bin/gcc -b i486-linuxlibc5 -nostdinc \
-I/usr/i486-linuxlibc5/include \
-I/usr/lib/gcc-lib/i486-linuxlibc5/2.7.2.2/include "$@"
```

Használd ezt a szkriptet fordításnál a "gcc" helyett.

7 C++ programok fordítása

A libg++ használja a matematikai programkönyvtárat, tehát a libm könyvtárhoz van kapcsolva. Mivel a meglévő libg++ könyvtárak a régi C programkönyvtárat használja, újra kell fordítanod a glibc-vel, vagy telepítened kell egy bináris könyvtárat (amit glibc-vel fordítottak). A libg++ legfrissebb változatát (forráskódot vagy bináris csomagot x86-os architektúrához) letöltheted a

[<ftp://ftp.yggdrasil.com/private/hjl/>](ftp://ftp.yggdrasil.com/private/hjl/) ftp helyről.

7.1 A libg++ telepítése teszt glibc könyvtárral

Ha teszt könyvtárként telepítetted a glibc-t, abba a könyvtárba kell telepítened a fájlokat, amelyekben a glibc található (pl. `/usr/i486-linuxglibc2`, az előző fejezetekben leírt példák alapján). Ha bináris csomagból telepítetted a libg++ könyvtárat (amit forrón ajánlok, mert még sohasem volt szerencsém így libg++ könyvtárat fordítani), csomagold ki a fájlokat egy átmeneti könyvtárba, majd helyezd át a `usr/lib/` könyvtárban lévő fájlokat az `<install directory>/lib/` könyvtárba, a `usr/include/`-ban lévő fájlokat a `<install directory>/include/` könyvtárba (ne felejtse el először törölni az `include/g++` kapsot!) és a `usr/bin/`-ben lévő fájlokat az `<install directory>/bin/` könyvtárba.

7.2 A libg++ telepítése elsődleges glibc könyvtár esetén

Ha a glibc-t elsődleges könyvtárként telepítetted, először át kell helyezned a régi libg++ fájljaidat a régi libc könyvtárba, abban az esetben ha ezen túl is akarsz fordítani g++ programokat a régi libc-vel. valószínűleg ennek az a legegyszerűbb módja, hogy telepítesz egy libc5-el fordított új libg++, mint az előző fejezetben és utána feltelepítetted a glibc-t normálisan.

7.3 C++ programok fordítása a nem elsődleges libc könyvtárral

Ha C++ programokat szeretnél nem elsődleges libc-vel fordítani, meg kell adnod a g++ include könyvtárat, ami a fenti példák esetén `/usr/i486-linuxglibc2/include/g++` lenne teszt glibc konfiguráció esetén vagy `/usr/i486-linuxlibc5/include/g++` elsődleges glibc konfiguráció esetén. Ezt általában el lehet érni a `$CXXFLAGS` változó módosításával:

```
CXXFLAGS = -nostdinc -I/usr/i486-linuxglibc2/include \  
           -I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include \  
           -I/usr/i486-linuxlibc5/include/g++ -b i486-linuxglibc2
```

8 Problémák

A glibc csomagban található egy GYIK (FAQ), amiben további információt és megoldásokat találhatsz a problémákra. A GYIK-nek van egy hálózaton elérhető változata is: <<http://www.imaxx.net/~thrytis/glibc/glibc-FAQ.html>> . Most kiemelnék néhány problémát, amikről a GYIK-ben nincs (vagy nem elég részletesen van) szó.

8.1 Nem működik a gépnevek feloldása

A glibc 2 más módszert használ a gépnevek feloldására, mint a libc5. A glibc "name server switch" (NSS) kódrésze keresi a `/etc/nsswitch.conf` fájlt. Ha nem működik a gépnév feloldás, amikor glibc 2 könyvtárát használó alkalmazásokat futtatsz, és a `/etc/resolv.conf` fájlod megvan, és tartalmazza a szükséges információkat, nézd meg, hogy van-e `/etc/nsswitch.conf` fájlod. Ha nincs ilyen nevű fájl, akkor hozz létre egyet, a következő tartalommal:

```
hosts:      files dns
```

Ezzel beállítottad, hogy a névkiszolgálókra vonatkozó információk a `/etc/resolv.conf` fájlban vannak.

Ha további részletek is érdekelnek, olvasd el a libc info lapját, amiben részletesen le van írva a nsswitch.conf fájl felépítése.

9 Hibák bejelentése

Ha úgy gondolsz, hogy hiba van a programkönyvtárban, olvasd el először a GYIK-et. Nem kizárt, hogy másnak esetleg ugyanaz a problémája volt, mint Neked, és van egyszerű megoldás. Ellenőrizd még a glibc csomagjában található `INSTALL` fájl "Recommended Tools to Install the GNU C Library" részét, mert a problémád adódhat más program hibájából is, nem feltétlenül a glibc működik rosszul.

Ha hibát találtál, először győződj meg róla, hogy tényleg igazi hibáról van szó. Ennek ellenőrzésére egy jó módszer, ha összehasonlítod a GNU C programkönyvtár és valamilyen másik C programkönyvtár viselkedését. Ha ugyanaz az eredmény mind a két programkönyvtárral, akkor valószínűleg nem hibáról van szó (bár nem 100%). Ha a két könyvtár viselkedése nem egyezik meg, akkor valószínű, hogy hibát találtál.

Ezek után nézd meg a <<http://www-gnats.gnu.org:8080/cgi-bin/wwwnats.pl>> címen elérhető hibákat tartalmazó adatbázist (bug database). Lehet, hogy valaki már előtted bejelentette a hibát. Érdemes megnézni a glibc csomagban található `BUGS` fájlt, amiben szintén az ismert hibák listája található.

Ha már biztos vagy benne, hogy új hibát találtál, próbáld megállapítani a lehető legpontosabban, egy nagyon egyszerű tesztprogrammal, hogy mi okozza a hibát. A C program könyvtár esetében ez azt jelenti, hogy meg kell állapítanod, hogy melyik függvényhívás idézi elő. Ez nem túl nehéz feladat.

A legutolsó lépés, amikor megvan az egyszerű kis tesztprogram, a hiba bejelentése. A hiba bejelentésénél, küldd el a tesztprogramot, az eredményeket amiket kaptál, az eredményeket amiket vártál, esetleg arról

valamit, hogy szerinted mi lehet a gond (ha van valami ötleted), a rendszer típusát, a GNU C programkönyvtár verzióját, a gcc fordító verzióját és a GNU Binutils verzióját. Mellékelj még a `config.status` és `config.make` fájlokat, ezeket abban a könyvtárban találod, ahonnan a `configure` szkriptet futtattad.

Minden a GNU C könyvtárral kapcsolatos hibabejelentést a `glibcbug` shell szkript segítségével kell elküldeni a `<bugs@gnu.org>` címre (működik még a régi `<bugs@gnu.ai.mit.edu>` cím is), vagy a GNATS webes interfészén keresztül a `<http://www-gnats.gnu.org:8080/cgi-bin/wwwgnats.pl>` honlapon.

Javaslatokat és kérdéseket a `<bugs-glibc@prep.ai.mit.edu>` levelezőlistára lehet küldeni. Ha nem olvasod a `gnu.bug.glibc` hírcsoportot, feliratkozatsz a listára a `<bug-glibc-request@prep.ai.mit.edu>` honlapon.

Kérlek NE küldjél GNU C könyvtárral kapcsolatos hibabejelentést a `<bug-gcc@prep.ai.mit.edu>` címre. Ez a lista a GNU CC hibáival foglalkozik. A GNU CC és a GNU C könyvtár két külön dolog, különböző emberek tartják karban.

10 Példa specs fájl

Ide beszúrtam egy `specs` fájlt, amit a gcc használ a glibc 2-vel fordításnál és linkelésnél. A fájl a `/usr/lib/gcc-lib/<new system dir>/<gcc version>` könyvtárban található. Ha x86-os rendszered van, valószínűleg bemásolhatod a fájlba ezt a részt egy az egyben.

```
*asm:
%{V} %{v:%{!V:-V}} %{Qy:} %{!Qn:-Qy} %{n} %{T} %{Ym,*} %{Yd,*} %{Wa,*:*}

*asm_final:
%{pipe:-}

*cpp:
%{fPIC:-D__PIC__ -D__pic__} %{fpic:-D__PIC__ -D__pic__} %{!m386:-D__i486__}
%{posix:-D_POSIX_SOURCE} %{pthread:-D_REENTRANT}

*cc1:
%{profile:-p}

*cc1plus:

*endfile:
%{!shared:crtend.o%s} %{shared:crtendS.o%s} crtn.o%s

*link:
-m elf_i386 %{shared:-shared}    %{!shared:      %{!ibcs:      %{!static:
%{rdynamic:-export-dynamic}    %{!dynamic-linker:-dynamic-linker /lib/ld-linux.so.2}}
%{static:-static}}

*lib:
%{!shared: %{pthread:-lpthread}  %{profile:-lc_p} %{!profile: -lc}}

*libgcc:
-lgcc

*startfile:
%{!shared:      %pg:crt1.o%s} %{!pg:%p:crt1.o%s}          %{!p:%{profile:gcr1.o%s}
%{!profile:crt1.o%s}}}}    crti.o%s %{!shared:crtbegin.o%s} %{shared:crtbeginS.o%s}
```



```
*switches_need_spaces:

*signed_char:
%{funsigned-char:-D__CHAR_UNSIGNED__}

*predefines:
-D__ELF__ -Dunix -Di386 -Dlinux -Asystem(unix) -Asystem(posix) -Acpu(i386) -Amachine(i386)

*cross_compile:
0

*multilib:
. ;
```

11 Egyéb

11.1 További információ

11.1.1 Web lapok

- *Az FSF GNU C Programkönyvtár Weblapja* <<http://www.gnu.org/software/libc/libc.html>>
- *GNU Libc 2 használata Linuxszal* <<http://www.imaxx.net/~thrytis/glibc/>>
- *Glibc-2 telepítése Linuxra* <<http://huizen.dds.nl/~frodol/glibc/>> .
- *Debian libc5-ről libc6-ra Mini-HOWTO* <<http://www.gate.net/~storm/FAQ/libc5-libc6-Mini-HOWTO.html>> .

11.1.2 Hírcsoportok

- comp.os.linux.development.system
- comp.os.linux.development.apps
- linux.dev.kernel
- gnu.bugs.glibc

11.1.3 Levelezőlisták

Glibc 2 Linux fórum.

Ez a lista olyan Linux felhasználók vitafóruma, akik feltelepítették a glibc2-t, az új GNU C programkönyvtárat. A témák: kompatibilitással összefüggő kérdések, programok fordítása során felmerülő problémák (Linux/glibc környezetben). A feliratkozáshoz küldj egy levelet a *Majordomo* <<mailto:Majordomo@ricardo.ecn.wfu.edu>> címre, a levél törzsében legyen: "subscribe glibc-linux <a saját e-mail címed>".

A levelezőlista archívuma megtalálható a <<http://www.progressive-comp.com/Lists/?l=linux-glibc&r=1&w=2##linux-glibc>> címen.

11.2 Köszönet

Ezen információ nagy részét a *GNU Libc* <<http://www.gnu.org/software/libc/libc.html>> honlapról loptam, valamint Ulrich Drepper <drepper@gnu.ai.mit.edu> glibc 2 bejelentéseiből és megjegyzéseiből. Andreas Jaeger-től <aj@arthur.rhein-neckar.de> származik a hibabejelentést leíró fejezet egy része.

A következők voltak segítségemre e dokumentum összeállításában:

- Alex <allex@ms2.accmail.com.tw>
- Mark Brown <M.A.Brown-4@sms.ed.ac.uk>
- Ulrich Drepper <drepper@gnu.ai.mit.edu>
- Scott K. Ellis <ellis@valueweb.net>
- Aron Griffis <agriffis@coat.com>
- Andreas Jaeger <aj@arthur.rhein-neckar.de>
- Hank Leininger <hlein@progressive-comp.com>
- Frodo Looijaard <frodol@dds.nl>
- Ryan McGuire <rmcguire@freenet.columbus.oh.us>
- Shaya Potter <spotter@capaccess.org>
- Les Schaffer <godzilla@futuris.net>
- Andy Sewell <puck@pookhill.demon.co.uk>
- Gary Shea <shea@gtsdesign.com>
- Stephane <sr@adb.fr>
- Jan Vandenbos <jan@imaxx.net>
- Michael Wolf <wolfm@rpi.edu>

Ezt a dokumentumot különböző nyelvekre fordítják:

- Kínai: Alex <allex@ms2.accmail.com.tw>
- Francia: Olivier Tharan <tharan@int-evry.fr>
- Japán: Kazuyuki Okamoto <ikko-@pacific.rim.or.jp>
- Magyar: Kovács Emese <emese@eik.bme.hu>

11.3 Visszajelzés

Ezen HOGYAN írásától, valamint a *glibc 2 for Linux* <<http://www.imaxx.net/~thrytis/glibc>> honlap karbantartásától eltekintve semmi közöm sincs a glibc projekthez. Távol állok attól, hogy mindent tudjak a témakörrel, ennek ellenére szívesen megpróbálok segíteni, ha gondjaid támadnak. Szívesen látok minden visszajelzést, javítást vagy kiegészítést a ejg3@cornell.edu <<mailto:ejg3@cornell.edu>> címen.

11.4 Szerzői jogok

Copyright © Eric Green, 1997.

Ez a dokumentum továbbadható az LDP licencében rögzített feltételek mellett.

11.5 Magyar fordítás

A magyar fordítást

Kovács Emese <mailto:emese@eik.bme.hu_NO_SPAM> készítette (1999). A lektorálást

Daczi László <mailto:dacas@freemail.hu_NO_SPAM> végezte el (2002.12.28). Bármilyen fordítással kapcsolatos észrevételt a

linuxhowto@sch.bme.hu <mailto:linuxhowto@sch.bme.hu_NO_SPAM> címre küldjétek. Eme dokumentum legfrissebb változata megtalálható a

Magyar Linux Dokumentációs Projekt <<http://tldp.fsf.hu/index.html>> honlapján.