

Glibc telepítés HOGYAN

Kai Schlachter

`linux (at) murphyslantech.de`

Glibc telepítés HOGYAN

Kai Schlachter

Megjelent 2004.03.19

Verziótörténet

Verzió: 1.01 2004.03.19 Átdolgozta: KC

Első kiadás, az LDP által átnézve.

Tartalomjegyzék

1. Előszó	1
1.1. Szerzői jog	1
1.2. Verziótörténet	1
1.3. Köszönetnyilvánítás	1
1.4. Magyar fordítás	1
2. Bevezetés	2
2.1. Miért?	2
2.2. Mit?	2
3. Előkészületek	3
3.1. Szükséges dolgok	3
3.2. Speciális dolgok	3
4. A glibc telepítése.....	10
4.1. A forrás letöltése és fordítása	10
4.2. A telepítés.....	10
4.3. Miután a rendszer betöltődött.....	11
5. Hibakeresés — valami nem sikerült	13
5.1. Hibák a configure vagy a make futásakor a glibc fordítása közben.....	13
5.2. Ha a make install során történik valami baj	13

Fejezet 1. Előszó

1.1. Szerzői jog

Copyright (c) 2004 by Kai Schlachter

Ez a dokumentum másolható, terjeszthető és/vagy módosító a Free Software Foundation (Szabad Szoftver Alapítvány) által kiadott GNU Free Documentation License (Szabad Dokumentációs Licenc) 1.2 vagy későbbi változatának megfelelően, változatlan részek, első és hátsó borítószövegek borítószövegek nélkül. A licenc egy példányát megtalálod a következő címen: <http://www.gnu.org/licenses/fdl.html>.

1.2. Verziótörténet

- v1.01: Néhány rossz helyen lévő idézőjel korrigálása.

1.3. Köszönetnyilvánítás

Különösen szeretném megköszönni Nico Schmoignak, hogy segített helyreállítani az összeomlott rendszeremet, miután make install-al telepítettem a glibc-t, miközben a rendszer teljesen funkcionált. Főként emiatt írtam ezt a HOGYANt.

Ugyancsak köszönöm mindenkinek, akik segítettek munka közben a HOGYAN megfelelő formába önteni, a sajtóhibákat megkeresni stb.

1.4. Magyar fordítás

A magyar fordítást Diós Gergely (mailto: Dios.Gergely[kukac]stud.u-szeged[pont]hu) készítette (2004.06.09). A lektorálást Daczi László (mailto: dacas[kukac]freemail[pont]hu) végezte el (2004.06.16). A dokumentum legfrissebb változata megtalálható a Magyar Linux Dokumentációs Projekt (<http://tldp.fsf.hu/>) honlapján. A dokumentum fordítása a Szegedi Tudományegyetem (<http://www.u-szeged.hu/>) nyílt forráskódú szoftverfejlesztés speciálkollégium (<http://www.inf.u-szeged.hu/~havasi/opensource/>)ának segítségével valósult meg.

Fejezet 2. Bevezetés

Ebben a HOGYANban kifejtem, hogyan telepítsd a glibc új változatát a rendszeredre.

Azért írtam ezt a kézikönyvet, hogy a megkíméljek másokat azoktól a problémáktól, amelyekkel én is találkoztam.

Ez a HOGYAN egyfajta irányelv beállításokból és módszerekből, amelyek nálam működtek, ezért a dokumentum tartalmáért semmilyen felelősséget nem vállalok. Saját felelősségedre használd ezeket a példákat, elveket. Erősen ajánlott, hogy a nagyobb telepítések előtt, illetve bizonyos időközönként is készíts biztonsági mentést rendszeredről.

Ha bármilyen javaslatod van a HOGYANNal kapcsolatban, vagy találtál egy bugot valamelyik disztribúcióban és javítását, küldj egy e-mail-t a <linux_at_murphyslantech.de> címre.

2.1. Miért?

Jó kérdés. Miért is kéne egy újabb glibc-t feltelepíteni? Nos, több oka is lehet :

- fejlesztő vagy és szükséged lenne az újabb funkciókra
- le akarsz fordítani (compile) egy újabb programot, ami az újabb programkönyvtárat (library) használja
- szereted az új változatban lévő hibák okozta izgalmakat ;-)

2.2. Mit?

Ha még nem tudod mi is az a glibc, ne aggódj! Amikor először voltak ilyen problémáim egy program fordítása során, csak annyit tudtam, hogy a glibc változatom nem megfelelő a fordításhoz. Most, hogy már többet tudok, megpróbálok egyszerűen elmagyarázni, hogy tulajdonképpen mit is csinál a glibc.

A glibc csomag egy C nyelvben írt programkönyvtárat tartalmaz. A programkönyvtárak rendkívül hasznosak programírás során, ahelyett, hogy rögtönözve kellene írunk eljárást olyan egyszerűbb műveletek számára, mint például egy szám gyökének meghatározása, sok ilyen és hasonló függvény le van tárolva egy külön fájlban — az úgynevezett programkönyvtárban. Ha egy programkönyvtár újabb változata megjelenik, gyakran tartalmaz újabb függvényeket, vagy sokkal hatékonyabb algoritmusokat használ már meglévő függvényekben és így tovább.

Ez az oka annak, amiért néhány program panaszkodik a régebbi változatú glibc miatt, az ugyanis nem tartalmazza az összes függvényt, ami szükséges a program futásához.

Tudom, ez így technikailag nem pontos magyarázat, de a lényegét megértheted belőle.

Fejezet 3. Előkészületek

A glibc telepítése nem könnyű feladat, ezért sok dolgot előre meg kell csinálnod, legfőképp arra az esetre, ha valami tönkremegy. (Mint ahogy velem is történt az első glibc telepítésemkor, mivel nem tettem semmi előkészületet.)

3.1. Szükséges dolgok

Alapvetően két dologra lesz szükséged: már meglévő szoftverekre (például előre telepített programok), valamint pár program forráskódjára.

3.1.1. Szükséges szoftverek

- egy működő gcc
- a glibc egy régebbi változata ;-)
- GNU-binutils
- GNU-make
- a GNU-core-utils
- GNU-tar
- bash, vagy valami más shell
- jó ha van: Midnight Commander
- egy szerkesztő(vi, jed stb.)

3.1.2. Szükséges források

- a bash, vagy a kedvenc shell
- GNU-tar
- GNU-core-utils
- GNU-make
- természetesen a glibc
- talán a gcc is

3.2. Speciális dolgok

Képzelted milyen problémákat okozhat, ha eltávolítod az alap-programkönyvtárat, amire egy csomó program épül.

Nálam minden rendben ment, amíg ki nem adtam a **make install** parancsot. Körülbelül a telepítés felénél hibaüzenetet kaptam, miszerint az **rm** nem tudott lefutni, aztán észrevettem, hogy még az olyan alapparancsok sem működnek, mint a **cp**, **ls**, **mv**, **ln**, **tar** stb. Mind azt írta ki, hogy nem találják a szükséges programkönyvtárak egy részét.

Vizsont erre is van megoldás. Programfordításkor a fordítót (compiler) rá lehet kényszeríteni, hogy a programkönyvtárakból használt függvényeket belefordítsák a programba, így azoknak már nem kell ezeket megkeresni a programkönyvtárból.

Ezért ebben a fejezetben a telepítéshez szükséges segédprogramok statikus változatát készítjük el.

3.2.1. Amikre mindenképpen szükséged lesz

3.2.1.1. A GNU-Binutils

1. A legújabbat az ftp.gnu.org/gnu/binutils webhelyről töltheted le; a dokumentum íráskor ez a 2.14-es változat volt

2. Csomagold ki a forrást:

```
tar xIvf binutils-2.14.tar.bz2
```

3. Lépj be a könyvtárba:

```
cd binutils-2.14
```

4. Állítsd be a makefile-okat:

```
./configure
```

5. Fordítsd le a forrásokat:

```
make
```

6. Telepítsd:

```
make install
```

Ha a fordítás során valamilyen gettextel kapcsolatos problémába ütköznél (mint például: “undeclared reference to lib_intl” vagy valami hasonló), akkor szedd le és telepítsd a legújabb változatot az ftp.gnu.org/gnu/gettext webhelyről.

Ha ez nem segít, akkor próbáld meg kikapcsolni az anyanyelvi támogatást a következőképpen:

```
./configure --no-nls
```

A binutils statikus változatát nem muszáj elkészíteni, de nem árt, mert sok régi változatút használó rendszerrel találkoztam, amelyekkel majdnem mindig valamilyen hiba volt, ezért szerintem jó ötlet itt megemlíteni ezeket.

3.2.1.2. GNU make

A **make** parancs a felelős a források lefordításáért, a gcc és más, a fordításához szükséges programok hívásáért. Ha valami baj történik az új glibc-vel, szükséged lehet valamit lefordítani, ezért nem árt, ha van egy statikus változatod, másként nem menne.

1. Töltsd le a forrást az ftp.gnu.org/gnu/make/ webhelyről; a dokumentum írásakor a legújabb a 3.80 változat volt

2. Csomagold ki:

```
tar xIvf make-3.80.tar.bz2
```

3. Lépj be a létrehozott könyvtárba:

```
cd make-3.80
```

4. Ügyelj rá, hogy a binárisokat statikusra fordítsd le:

```
export CFLAGS="-static -O2 -g"
```

5. Futtasd a beállító szkriptet:

```
./configure
```

6. Fordítsd le:

```
make
```

7. Telepítsd a binárisokat :

```
make install
```

8. Ellenőrizz:

```
make -v
```

Az újonnan telepített változatot kell látnod. Ha nem, keresd meg a régi binárisokat, és cseréld ki őket egy szimbolikus hivatkozással (link) az új változatra.

Gratulálok! Ismét elkészítetted egy program statikus változatát.

3.2.1.3. A GNU core-utils

A core-utils a **cp**, **rm**, **ln**, **mv** stb alapparancsokból áll. Ha valami hiba történik a telepítés során, ezekre mindenképp szükség van a hiba helyrehozatalához, tehát itt a statikus változat igazán fontos.

1. Megint csak töltsd le a forrást az ftp.gnu.org/gnu/coreutils/ webhelyről; a dokumentum írásakor a legújabb az 5.0 változat volt.

2. Csomagold ki:

```
tar xIvf coreutils-5.0.tar.bz2
```

3. Lépj be a könyvtárba:

```
cd coreutils-5.0
```

4. Ügyelj rá, hogy a binárisokat statikusra fordítsd:

```
export CFLAGS="-static -O2 -g"
```

5. Futtasd a beállító szkriptet:

```
./configure
```

6. Fordítsd le:

```
make
```

7. És telepítsd:

```
make install
```

8. Ellenőrizd, hogy az új változat-e az aktuális:

```
cp --version
```

Az új változat számát kell látnod, ha nem, akkor cseréld ki az összes régit egy szimbolikus hivatkozással az új változatra.

Most, hogy ezeknek az alapsomagoknak a statikus változatát használod, biztos lehetsz benne, hogy minden körülmény közt működni fognak.

3.2.1.4. A GNU tar

Már használtad a GNU tar programot, amikor kicsomagoltad és telepítetted a programokat. De talán szükséged lehet majd újabb programokat fordítani, ha az új glibc miatt a rendszer összeomlik, tehát nem árt, ha ebből is van egy statikus változat (ezt személyesen tapasztaltam). Szükséged lesz a **tar** mellett a bz2 tömörítési algoritmusra, amely nincs benne a tar hivatalos forrásában.

1. Töltsd le a tar forrását az ftp.gnu.org/gnu/tar webhelyről; a dokumentum írásakor a legújabb az 1.13 változat volt.

2. Mivel sok forrás bz2 algoritmussal van tömörítve, szükséged lesz a beépített támogatásra, mert jobb mint csövekkel (pipe) dolgozni. Tehát töltsd le a foltot az

```
ftp://infogroep.be/pub/linux/lfs/lfs-packages/4.1/tar-1.13.patch
```

3. Csomagold ki:

```
tar xzvf tar-1.13.tar.gz
```

4. Másold át a foltot a tar forráskönyvtárába:

```
cp tar-1.13.patch tar-1.13/
```

5. Foltozd a forráskódot:

```
patch -Np1 -i tar-1.13.patch
```

6. Állítsd be a fordító kapcsolóit a statikus fordításhoz:

```
export CFLAGS="-static -O2 -g"
```

7. Most már beállíthatod:

```
./configure
```

8. Fordítsd le:

```
make
```

9. Végül telepítsd a csomagot:

```
make install
```

10. Egy gyors ellenőrzés, hogy feltelepült-e az új változat:

```
tar --version
```

Az új változatnak kell megjelenie, ellenkező esetben keresd meg a régieket és cseréld ki az újra mutató szimbolikus hivatkozásokkal.

Ha valami hibát észlelsz a make futtatásakor, próbáld meg kikapcsolni az anyanyelvi támogatást (nls). Ezt a következő kapcsolókkal érheted el:

```
--disable-nls
```

Megjegyzés: A tar ezen új változatában a -j kapcsolót kell használnod a .bzip2 fájlok kicsomagolásához, tehát a következő helyett:

```
tar xIvf anyfile.tar.bz2
```

így kell:

```
tar xjvf anyfile.tar.bz2
```

Nem tudom, hogy ez miért változott meg, mindenesetre így szépen működik.

3.2.1.5. A Bash shell

Én a Bash shell-t részesítem előnyben, ha másikat használasz, ügyelj arra, hogy a statikus változatát telepítsd, a glibc telepítése előtt.

1. Töltsd le a Bash forrást az ftp.gnu.org/gnu/bash/ webhelyről; a dokumentum írásakor a legújabb a 2.05b változat volt.

2. Csomagold ki:

```
tar xzvf bash-2.05b.tar.gz
```

amely egy bash-2.05b könyvtárat hoz létre, benne a forrásokkal.

3. Lépj be a könyvtárba:

```
cd bash-2.05a
```

4. Állíts be mindent a statikus programkészítéshez:

```
export CFLAGS="--static -O2 -g"
```

5. Állítsd be a makefile-okat:

```
./configure
```

Ha valami extrát szeretnél a Bash-hoz, nézd meg a súgót:

```
./configure --help
```

az összes kapcsolóról.

6. Fordítsd le:

```
make
```

7. Telepítsd fel a binárisokat:

```
make install
```

Ez az `/usr/local/bin/` könyvtárba másolja a binárisokat.

8. Ügyelj arra, hogy ne legyen más változat elérhető (mint a SuSE Linuxnál a `/bin/` könyvtárban). Egyszerűen másold a binárist a `/bin/` könyvtárba is:

```
cp /usr/local/bin/bash /bin/
```

Itt nem használunk szimbolikus hivatkozásokat, mivel a rendszerbetöltésnél és a bash indításakor ebből problémák adódhatnak.

Már van egy telepített statikus változatú Bash a rendszeren, ami így ugyan nagy, de legalább minden körülmény közt működik.

Ha másik shell-t szeretsz, használhatod azt is, csak ügyelj arra, hogy statikusan fordított változatot használj. Nyugodtan írd meg hogyan készítetted az általad használt shell statikus változatát, valószínűleg a HOGYAN következő változatában már szerepelni is fog.

3.2.2. Szoftverek, amelyek jól jöhetnek

3.2.2.1. Midnight Commander

A Midnight Commander egy nagyon hasznos kis fájlkezelő, sok extrával, például tömörített fájlok kicsomagolása, beépített általános parancsok, mint másolás, áthelyezés, és egy szerkesztő.

Lefordításához, fent kell lennie glib programkönyvtárnak. Néhány disztribúcióban alaptól fent van, ha esetleg a **make** során olyan hibaüzenetet kapsz, hogy az ld nem találja a glibet, akkor először fel kell telepítened ezt a programkönyvtárat. A forrásokat az ftp.gnome.org/pub/gnome/sources/glib/2.2/ webhelyről töltheted le, a telepítés egyértelmű.

A Midnight Commander elkészítésének lépései:

1. Töltsd le a forrást a <http://www.ibiblio.org/pub/Linux/utils/file/managers/mc/> (<http://www.ibiblio.org/pub/Linux/utils/file/managers/mc/>) webhelyről; a dokumentum írásakor a legújabb a 4.6.0 változat volt. volt.

2. Csomagold ki:

```
tar xzvf mc-4.6.0.tar.gz
```

3. Lépj be az imént létrehozott könyvtárba:

```
cd mc-4.6.0
```

4. Állítsd be a konfigurációs fájlokat:

```
./configure
```

5. Fordítsd le:

```
make
```

6. Telepítsd:

```
make install
```

Fejezet 4. A glibc telepítése

Elérkeztünk a legfontosabb részhez, a glibc telepítéséhez.

4.1. A forrás letöltése és fordítása

A glibc számos változata elérhető, de nem biztos, hogy a legújabb változat jobb, mint egy régebbi. A legjobb, ha utánanézel különböző internetes fórumokon, melyiket érdemes használni és melyiket nem. Vagy ha ismersz valakit, aki járatos a témában, kérdezd meg tőle, hátha tudja melyik változat mennyire jó.

Én a glibc-2.2.4-et telepítettem, mivel azt mondták, ez elég szépen működik, de más változatot is feltehetsz, ha az jobban tetszik.

Akkor munkára fel:

1. Töltsd le a forrást az ftp.gnu.org/gnu/glibc/ webhelyről; amint említettem, a 2.2.4-es változatot használtam.

2. Csomagold ki:

```
tar -xzvf glibc-2.2.4.tar.gz
```

3. Még szükséged lesz a "linuxthreads" csomagra is, amelyet a linuxthreads könyvtárban megtalálsz az ftp.gnu.org szerveren. A fájl neve:

```
glibc-llinuxthreads-2.2.4.tar.gz
```

Ügyelj arra, hogy a linuxthreads változatszáma megegyezzen a glibc-ével.

4. Másold a linuxthreads csomagot a glibc forrás-könyvtárába:

```
cp glibc-linuxthreads-2.2.4.tar.gz glibc-2.2.4
```

5. Lépj be a glibc könyvtárába:

```
cd glibc-2.2.4
```

6. Csomagold ki a linuxthreads forrás t:

```
tar xzvf linux-threads-2.2.4.tar.gz
```

7. Állítsd be:

```
./configure --enable-add-ons=linuxthreads
```

Ez a kapcsoló úgy állítja be a csomagot, hogy fordításkor vegye figyelembe a linuxthreads csomagot is. Ez a más Linux rendszerekkel való kompatibilitás miatt szükséges. E nélkül például nem biztos, hogy az általad fordított programok futni fognak más gépeken.

8. Majd indítsd el a glibc fordítását:

```
make
```

Ez eltart egy darabig (körülbelül fél óráig tartott a 1.5 GHz-es Duron XP gépen).

Most, hogy a programkönyvtár is le lett fordítva, minden készen áll a telepítéshez, de az már nem lesz ilyen könnyű feladat.

4.2. A telepítés

A glibc telepítéséhez egy olyan rendszerre van szükség, amin semmi sem fut, mivel számos folyamat (például a sendmail) mindig megpróbálja használni a programkönyvtárat, így megakadályozza annak lecserélését. Tehát egy "lecsupaszított" rendszerre van szükségünk, amin tényleg csak a legszükségesebb dolgok futnak. Ezt úgy tudod elérni, hogy a boot opcióknál beállítod az

```
init=/bin/bash
```

értéket a rendszermaghoz. A rendszerbetöltőtől függően még meg kell tenned pár dolgot. A következőkben ezt fejtem ki, a két legelterjedtebb rendszerbetöltő, a LILO (LIⁿux LOader) és a GNU grub esetére.

4.2.1. LILO

Egy "alap" rendszer használatához indítsd újra a számítógépet, és a LILO prompt-nál válaszd ki a neked megfelelő kernel-image nevet, és fűzd hozzá a

```
init=/bin/bash
```

sort, mielőtt **Enter**-t nyomsz. Ha gyakrabban akarsz glibc-t cserélni, célszerű egy külön beállítást adni a `/etc/lilo.conf` fájlba. A részletekért nézd meg a LILO man oldalát.

4.2.2. Grub

A Grub egy újabb rendszerbetöltő, amely támogat többféle operációs- és fájlrendszert (például támogatja a reiser fájlrendszerű partíciókról való rendszerbetöltést). Ha többet akarsz megtudni róla, látogass el a <http://www.gnu.org/software/grub/> webhelyre, ahol minden szükséges dolgot megtalálsz.

Ha már van egy telepített Grub a gépen, akkor valószínűleg szöveges felületet használsz a megfelelő rendszer kiválasztására. A Grub-nak van egy szép jellegzetessége — ahelyett, hogy kézzel kéne beírnod mindent, egyszerűen kiválasztod a menüből a bejegyzést, és nyomsz egy **e**-t, amely megjelenít egy opciók menüt. Ebben a menüben látod a parancsokat, amelyeket a Grub a rendszerbetöltést megelőzően hajt végre. Válaszd ki a következő sort :

```
kernel="/ahol/a-kerneled-van és az-opcióid-vannak"
```

és nyomj újra egy **e**-t. Most szerkesztheted a sort. Itt add hozzá az

```
init=/bin/bash
```

sort, eztán nyomj **Enter**-t, hogy a változtatások érvényesüljenek, és nyomj **b**-t a rendszerbetöltéshez.

4.3. Miután a rendszer betöltődött

... Egy abszolút minimális bash környezetben találsz magad.

Még a felhasználónevet és a jelszót sem kellett begépelned! Most te vagy az egyetlen szuper-felhasználó minden hatalommal, mivel a rendszer egyfelhasználós üzemmódban működik, tehát gondold végig mit csinálsz. Még fájl-elérési jogok és hasonlóak sincsenek!

A prompt körülbelül így néz ki:

```
init-x.y#
```

A gyökérkönyvtár most csak olvashatóként van felcsatolva, így nem tudod ráírni az új programkönyvtárat sem. Az írhatóvá (r/w) tételhez, elég kiadni ezt a parancsot:

```
mount -o remount,rw /
```

Ha a forrásod más partíción van, akkor azt is fel kell csatolni (nálam a raid rendszert kell):

```
mount -t reiserfs /dev/md0 /usr/src
```

Amint látod, megadtam a fájlrendszer típusát is, mivel a **mount** nem nézi meg az `/etc/fstab` fájlban.

Most már beléphetsz a forrás-könyvtárába, és elkezdheted a fordítást:

```
make install
```

Ha akarsz, imádkozhatsz, hogy minden működjön... ;-)

Ha minden jól ment, akkor telepítés után mindenféle hibáüzenet nélkül visszakapod a promptot. Ha mégsem, akkor olvasd el az 5 fejezetet.

Ha nem volt semmi hiba, írd be ezt a parancsot:

```
ldconfig -v
```

a programkönyvtáraid frissítéséhez.

Gratulálok! A programkönyvtárat sikeresen telepítetted. Most írd be: **mount -o remount,ro /** biztosítékképpen, hogy minden adat a merevlemezre íródott.

Indítsd újra a rendszert:

```
exit
```

Ez egy hibáüzenetet fog eredményezni, miszerint rendszermag-pánikot okoztál. Ha lehet, **CTRL-ALT-DEL**-el próbáld meg újraindítani a számítógépet, ha nem megy, akkor csak a reset gomb marad.

Próbáld meg a rendes rendszermagot betölteni. Ha minden jól megy, már használhatod is az új programkönyvtáradat.

Fejezet 5. Hibakeresés — valami nem sikerült ...

Ha annak ellenére, hogy pontosan követted az összes utasítást és ehhez a részhez érkeztél, valószínűleg a különböző Linux kiadások miatt van valami probléma, némelyikük nem ott tárol bizonyos dolgokat, ahol azt elvileg kéne. A Suse Linux a leghíresebb az ilyen buta dolgokról, de a többivel is előfordulhat hasonló probléma. Ha ilyesmivel találkozol, és megtalálod a hiba okát — és remélhetőleg a megoldását is — tudasd velem, én hozzá fogom adni ehhez a HOGYANhoz.

Azt hiszem ez a rész sosem lesz teljesen kész, de leírok majd pár lehetséges hibát és megoldásukat.

5.1. Hibák a configure vagy a make futásakor a glibc fordítása közben

Néha találkozhatasz olyan beállítási hibával, amely szerint valamely kellék hiányzik — általában valamilyen szoftver, vagy a programkönyvtárak túl régiek. Ezzel egy csomó programnál találkoztam, különösen a statikus fordítások során. Általában semmi baj, szerezd be a szükséges szoftvereket, vagy programkönyvtárakat, azután fordítsd le őket a forrásban található leírásoknak megfelelően (általában README, INSTALL, vagy valami hasonló a nevük).

Viszont van néhány eset, amikor ez így nem működik. Például, az újabb változatú binutils fordításakor, amely szükséges a glibc fordításához (ezért is említettem a szükséges dolgok között), problémák adódtak. A binutils configure szkriptje azt írta, hogy “A glibc túl régi”. Tehát azt hittem, *a kígyó itt a saját farkába harap*. Szerencsére erre a problémára is van megoldás, ha nem tudsz egyszerre megtenni egy nagy lépést, próbálj több, kisebb lépésben haladni.

A disztribúciómban egy 2.1.1-es változatú glibc volt. A probléma megoldásához először megpróbáltam lefordítani a 2.1.3-as glibc-t, ami sikerült is. Ezután feltelepítettem ezt az újabb változatot. Mikor újra megpróbáltam a binutils-t lefordítani, ezúttal mindent rendben talált.

Ha egy hasonló “ördögi körrel” találkoznál, nézz utána, hogy minimum milyen változatú szoftverre van szükség, aztán töltsd le azt. (Azt hiszem ezért is tartanak az FTP szervereken olyan sok régi változatot). A sikeres fordítás és telepítés után próbáld meg fordítani azt a szoftvert, ami panaszkodott az előbbi szoftver változata miatt, a legtöbb esetben már le tudod fordítani. Lehetséges, hogy ezt a lépést többször is meg kell ismételned egy hiányzó, vagy régi szoftver fordításánál. Ez az amit úgy hívok, hogy “a patkány hosszú farka” vagy “dominó effektus”. Valami megtételéhez több dolgot meg kell tenned. Ez elég idegesítő lehet, de a jó oldala a dolognak, hogy utána már biztos lehetsz benne, az igazán régi programok frissítve lesznek a telepítés végére.

5.2. Ha a make install során történik valami baj

A legáltalánosabb hiba, hogy nincsenek meg a statikus alapkellékek; ebben az esetben csak a **cd** parancsot tudod használni, semmi más. Ezért írtam le a HOGYANban részletesen, hogyan tegyük azokat a kellékeket statikussá.

Az egyetlen nem statikus kellék a **mount**, amely a linux-utils csomagban van (véleményem szerint nagyon helyesen), amely tartalmazza még többek között a **login**, **passwd** stb. programokat. Mivel nem lehet statikus változatokat használni PAM vagy más biztonsághoz kapcsolódó szoftverekkel keverve, elég ésszerűtlen dolog lenne bármilyen körülmények közt is statikusan fordítani őket. Természetesen megteheted, ha tényleg tudod mit csinálsz.

5.2.1. Visszatérés egy működő rendszerhez

Visszaállítani egy működő rendszert elég egyszerű, ha megvannak a statikus kellékek: lépj be a `/usr/local/lib/` könyvtárba, és tedd át az összes újonnan telepített fájlt egy másik könyvtárba (például `/usr/local/lib/eltarolva`). Könnyen azonosíthatod őket a változatszámuk alapján, amely ugyanaz, mint a telepített glibc programkönyvtaré, valamint a létrehozási dátum alapján. Elég ritka, hogy két különböző programkönyvtárnak azonos a változatszáma és létrehozási ideje — én személy szerint sosem láttam ilyet — de a biztonság kedvéért ellenőrizd a dátum/időt, hogy semmi fontosat ne törölj le. Ehhez jó segítséget nyújt a Midnight Commander, ha telepítve van.

Megpróbálhatod letörölni a `ld-2.2.4.so` és a `libc-2.2.4.so` fájlokat, majd kiadni egy **ldconfig -v** parancsot, mielőtt eltávolítanád az összes tönkrement fájlt. Ezzel lehetővé válik a legtöbb program használata, a Midnight Commander-t minden esetben tudod használni.

Ne felejsd el legalább egy **ldconfig -v** parancsot kiadni, miután eltávolítottad az összes fájlt.

5.2.1.1. A sikertelen telepítés okának megszüntetése

Gyakori oka a problémáknak, hogy a disztribúciód az összes programkönyvtárat egy másik helyen tárolta, mint ahol az újonnan telepített rutin keresné, valamint gyakran több változat is fut párhuzamosan, zavarva egymást. Nálam sok problémát okozott a `libc6.so` egy második példánya a `/lib` könyvtárban. Egy szimbolikus hivatkozás az `/usr/local/lib` könyvtárban lévő megfelelő fájlra megoldja ezt a problémát.