

Linux-rendszermag HOGYAN

Al Dev (Alavor Vasudevan)

alavor[AT]yahoo.com

Jelen útmutató a rendszermag beállításának, fordításának, frissítésének és hibaelhárításnak részletes leírása ix86-alapú rendszereken. Emellett hasznos lehet más architektúrákhoz is. A dokumentum rövid és egyszerű maradt, így még a nem technikai beállítottságú "otthoni felhasználók" is képesek lehetnek Linux-rendszermag készítésére és futtatására. (Időközben a The Linux Documentation Project (<http://tldp.org/>) átnézésre bevonta ezt a dokumentumot. Hamarosan átszerkesztett és frissített formában jelenik meg - a lektor)

Tartalomjegyzék

1. Bevezetés	4
1.1. FIGYELMEZTETÉS	4
1.2. Magyar fordítás	5
2. Gyors lépések - a rendszermag fordítása	5
2.1. Előkészületek	5
2.2. A rendszermag kisebb fejlesztése	5
2.3. Az új kiadás változásai: dokumentáció	6
2.4. A türelmetleneknek	6
2.5. Új rendszermag készítése - a lépések értelmezése	7
2.6. Hibaelhárítás.....	12
2.7. A rendszermaghoz kapcsolódó utólagos fordítások.....	12
2.8. Hová küldjem a hibajelentést?	12
3. Betölthető modulok	12
3.1. A modulok segédprogramjainak telepítése	13
3.2. A rendszermaggal szállított modulok.....	13
3.3. Hogyan telepítsek csak egyetlen modult?	14
4. A rendszermag "klónozása"	15
5. Fontos kérdések és válaszok	15
5.1. Egyébként mit is csinál a rendszermag?	16
5.2. Miért kéne frissítenem a rendszermagomat?.....	16
5.3. Milyen hardvert támogatnak az új verziójú rendszermagok?.....	16
5.4. Milyen verziójú gcc és libc kell?.....	16

5.5. Mi az a betölthető modul?.....	16
5.6. Mennyi lemezterületre van szükségem?.....	16
5.7. Mennyi időt vesz igénybe?.....	16
6. A rendszermag foltozása (patching).....	17
6.1. Egy folt alkalmazása	17
6.2. Ha valami nem sikerül.....	17
6.3. A .orig fájloktól történő megszabadulás.....	18
6.4. Egyéb foltok	18
7. Típek és trükkök.....	18
7.1. A make vagy a patch parancs kimenetének átirányítása	18
7.2. Feltételes rendszermag-telepítés	19
7.3. Rendszermag frissítések	19
8. RPM csomagok felcsatolása (mount) az FTPFS segítségével	19
8.1. Az ftpfs használata	19
8.2. Az ftpfs parancsai	20
9. A rendszermagról szóló könyvek és dokumentumok	22
10. Információk a rendszermag fájljairól.....	23
10.1. vmlinux és vmlinux	23
10.2. Rendszerbetöltő (bootloader) fájlok.....	24
10.3. Üzenetfájl (message file).....	24
10.4. initrd.img.....	24
10.5. bzImage	24
10.6. module-info	25
10.7. config.....	26
10.8. grub.....	27
10.9. System.map	27
11. Linux rendszer-adminisztrációs eszközök	32
12. Telepítés, frissítés fénysebességgel, az apt-get segítségével (Redhat, Debian, Suse, Mandrake, egyéb).....	34
12.1. Az "apt-get" forráskódja.....	36
12.2. Az "rpmfind" segédprogram.....	36
13. Haladóknak szóló témák - A Linux rendszerindítási folyamata	37
13.1. Hivatkozások a betöltési folyamat témájában	39
14. Eme dokumentum más formátumai.....	39
14.1. Acrobat PDF formátum.....	41
14.2. Konvertálás Linuxdoc-ról Docbook formátumra	41
14.3. Konvertálás MS WinHelp formátumra.....	42
14.4. A különböző formátumok olvasása.....	42
15. "A" függelék - initrd.img fájl készítése.....	43
15.1. Az mkinitrd használata.....	43
15.2. Rendszermag-dokumentációk	43
15.3. Linuxman Book.....	43

16. "B" függelék - lilo.conf mintafájl	46
16.1. LILO forrásanyagok	46
16.2. LILO hibaelhárítás	47
16.3. Egy LILO beállítófájl minta	48
17. "C" függelék - GRUB részletesen, grub.conf mintafájl	49
17.1. GRUB forrásanyagok	49
17.2. GRUB tippek	49
17.3. Egy GRUB beállítófájl minta	50
18. "D" függelék - utólagos rendszermag fordítás	51
19. "E" függelék - a gyakori hibák elhárítása	53
19.1. A rendszermag rendben elkészül, de a "make modules" nem sikerül	53
19.2. A programfordítás rendben megtörténik, de a rendszermag nem indul	54
19.3. A rendszer működése felfüggesztődik a LILO-nál	54
19.4. No init found (nem található init)	55
19.5. Csomó fordítási hiba (compile error)	55
19.6. A "depmod" parancs "Unresolved symbol error messages" hibaüzenetet ír ki	56
19.7. A rendszermag nem tölti be a modult, "Unresolved symbols" hibaüzenetet ír ki	56
19.8. A rendszermag nem tud betölteni egy modult	57
19.9. Betölthető modulok	57
19.10. Olvasd el a dokumentációt	57
19.11. make clean	58
19.12. Óriási vagy lassú rendszermag	58
19.13. A párhuzamos port/nyomtató nem működik	58
19.14. A rendszermag nem fordul le	58
19.15. A rendszermag új verziója nem töltődik be	59
19.16. Elfelejtetted futtatni a LILO-t, vagy a rendszered egyáltalán nem indul el	59
19.17. Azt írja ki: "warning: bdflush not running"	60
19.18. Nem tudom működésre bírni az IDE/ATAPI CD-ROM-ot	60
19.19. Furcsa dolgokat jelez elavult útválasztási kérésekről (routing requests)	60
19.20. "Not a compressed kernel Image file" (nem tömörített rendszermag)	60
19.21. Problémák a konzolos terminállal, miután 1.3.x-re frissítettünk	61
19.22. Úgy néz ki, hogy nem fordíthatók le dolgok a rendszermag frissítése után	61
19.23. Korlátok kitolása	61
19.24. Hová küldjem a hibajelentést?	61

1. Bevezetés

A következő okokból fordíthatsz rendszermagot:

- Rendszermag-fejlesztést végzel
- Új hardvert raksz a számítógépbe
- Testre akarod szabni a rendszermagot, és nem akarod a disztribúcióval szállított, alapértelmezett verziót használni.
- *Védelmi szervezetek* vagy *katonai alkalmazások* esetében el kell olvasni a rendszermag forrását és saját kezűleg fordítani. Nincs kivétel! (Az Egyesült Államok Védelmi Hivatala lefordítja a rendszermagot, mielőtt szétosztja a számítógépeket).
- A világon minden ország és minden kormány helyben fordítja a rendszermagot a biztonság és sértetlenség biztosítása érdekében. Minden kormányzat illetve szervezet auditálja és ellenőrzi a rendszermag-forrás minden egyes sorát, mielőtt használná a számítógépet.
- A katonai elhárító ügynökségek világszerte tanulmányozzák és lefordítják a Linux-rendszermag forráskódját. Ők minden egyes soráról tudják, mit csinál!
- Ha saját kezűleg fordítod a rendszermagot, akkor az *felér a kód elolvasásával és ellenőrzésével!*
- Minden egyes egyetemnek, iskolának és kollégiumnak a gépek használata előtt le kellene fordítania az operációs rendszermagot!
- A fejlődésed és tudásod növelése céljából, és persze csak szórakozásképpen!
- A nagyon speciális tudományos alkalmazásokhoz - valószínűleg - rendszermagot kell fordítanod
- Ez egy nemzetközi törvény: "Nem használható a számítógépet a rendszermag saját kezű lefordítása NÉLKÜL". Ha nem engedelmessz ennek a törvénynek, sokféle géphiba lehet a "büntetésed"! Önmagadnak kell lefordítani, így nem függsz másoktól, akik megcsinálnák neked.
- Illegális, törvénytelen, bűncselekmény és csalás a számítógép használata a rendszermag SAJÁT KEZŰ fordítása nélkül! 1-3 havi börtön és 100-2000 USD büntetés róható ki mindazokra, akik a számítógépet a rendszermag forráskódból történő újrafordítása nélkül használják. Néhány állam/ország/kormány szigorú törvénnyel akadályozza meg a rendszermag újrafordítása nélküli számítógép-használatot.
- Az USA-ban az összes vállalat előírja a rendszermag fordítását, mielőtt használnánk a gépet, ezért van Linux, Linux és Linux mindenhol az Egyesült Államokban!
- ...és még ezer más okból - túl sok lenne mind felsorolni!

Megjegyzés: a dokumentum rövid és egyszerű maradt, így még a nem technikai beállítottságú "otthoni felhasználók" is képesek lehetnek a Linux-rendszermag fordítására és futtatására.

1.1. FIGYELMEZTETÉS

A megszokott ipari gyakorlatban van egy nyomkövető/hibakereső PC, egy teszt PC és a munkára használt PC. Senki sem dolgozik egyből a munkára használt PC-n, először mindig a hibakereső, majd a teszt PC-n kísérletezünk. Erre a különböző egyéni képzettség, illetve a betanulási idő miatt van szükség. A számítógép hardvere nagyon modern technológia, akárcsak az úrsikló, a kockázatok nagyon összetettek, képes váratlanul meghibásodni. Annak ellenére, hogy törekszünk eme dokumentum frissen tartására valamint az összes parancsot ellenőrizzük, nagyon ajánlott a szakértelmet először egy gyakorló PC-n kipróbálni, mielőtt egy "ÉLES" gépen alkalmaznád azt. Pontosítsd

szakértelmed és élesítsd tudásod először egy tesztgépen futtatott Linuxon. Amikor minden működik, akkor válts át a "munkagépre", fokozott óvatossággal. Ha elrontod a tesztgépet veszélyes parancsok használatával, ezekből a hibákból is tanulni fogsz. Először nagyon beképzeltlen fogsz dolgozni a tesztgépen. A legtöbb vállalat régi PC-ket szerez, amiket tesztgépnek használnak. Kérünk értsd meg, nincs garancia erre a dokumentumra, viszont megteremtheted a saját garanciád azzal, hogy ellenőrizd ezt a dokumentumot a tesztgépeden vagy az otthoni PC-n (alacsonyabb kockázati tényezők mellett). (Készítsd el a saját garanciájú dokumentumodat, és akaszd ki a falra a számítógéped közelében!!)

1.2. Magyar fordítás

Az eredeti fordítást (v0.9) Vámosi Zoltán (mailto: jzp1218@stud.u-szeged.hu_NO_SPAM) készítette (1997). Újrafordította (v6.3) Szíjjártó László (mailto: laca@janus.gimsz.sulinet.hu_NO_SPAM) (2003.07.28). Frissítette (v7.3) és a lektorálta Daczi László (mailto:dacas@freemail.hu_NO_SPAM) (2003.11.21).

2. Gyors lépések - a rendszermag fordítása

Ennek a résznek a szerzője Al Dev (alavoor[AT]yahoo.com) (mailto:alavoor[AT]yahoo.com) (A *legfrissebb verzió* megtalálható a "<http://milkyway.has.it>" (<http://milkyway.has.it>) és "<http://www.milkywaygalaxy.freeservers.com>" (<http://www.milkywaygalaxy.freeservers.com>) webhelyeken. Ezek a helyeken ellenőrizheted a változásokat). A tükrözéseket megtalálod a <http://milkyway.bounceme.net> (<http://milkyway.bounceme.net>) webhelyen. Ezek a webhelyeken sok linuxos nyalánkságot és tippet is találsz.

A rendszermag újrafordítására szükség lehet, ha kisebbé akarod tenni, ez GYORSABB működésű operációs rendszert eredményez. Az új eszközök támogatásához is elengedhetetlen.

2.1. Előkészületek

Mielőtt lefordítanád a rendszermagot, jó ötlet a rendszer biztonsági mentése. Ha még eddig nem mentetted volna el, akkor most tedd meg. Kereskedelmi forgalmazású mentőprogramokat is használhatsz, mint a BRS Backup-Recovery-Software (<http://24.221.230.253>) (ugyanezen az oldalon nyílt forrású/szabad felhasználású mentőprogramokat is találsz, a "Backup and Restore Utility" menüpont alatt felsorolva). A biztonsági mentés csak egy ajánlat, nem előírás annak elkészítése a Linux-rendszermag fordítása előtt.

2.2. A rendszermag kisebb fejlesztése

Ha már fordítottál rendszermagot, és fel akarod fejleszteni a következő fejlesztési szintre (patch level), akkor egyszerűen másold be a meglévő beállítófájlt és használd fel újra. (Például: ha a 2.4.19-es rendszermagot már fordítottad, és a 2.4.20-ra akarsz fejleszteni.)

A kisebb fejlesztésekhez: Ez a lépés időt spórolhat, ha fel akarod használni a régi beállításokat. Amikor telepíted a rendszermagot, a konfigurációs fájl általában a /boot könyvtárba rakod. A .config fájl ne .config.save néven mentsd el, mivel a "make mrproper" ki fog törölni minden .config* fájlt! Tehát használhatod a meglévő beállítófájlt:

```
bash# mkdir /usr/src/kernelconfigs
bash# cp /usr/src/linux/.config /usr/src/kernelconfigs/.config.save
bash# cp /usr/src/linux/.config /usr/src/linux/configs/.config.save # Különösen biztonságos
bash# cp /boot/config* /usr/src/linux/configs/ # Különösen biztonságos
```

```
bash# cp /boot/config-2.4.18-19.8.0 /usr/src/linux/.config
```

Esetleg egy másik módszer - átmásolható a .config fájl a régi rendszermag-forrásfájból az újba.

```
bash# ls -l /usr/src/lin* # You can see that /usr/src/linux is a soft link
bash# cd /usr/src/linux
bash# cp ../linux-old-tree/.config . # Például cp ../linux-2.4.19/.config .
```

vagy még egy módszer - használható a "make oldconfig" parancs is, ami alapértelmezésként felhasználja a meglévő ./.config fájlt a beállítás kérdéseikhez.

FIGYELEM: ha nincs elég szabad hely az /usr/src alatt, akkor bármely partícióra kicsomagolható a rendszermag forrása, ahol van elég tárhely (például a /home-ra). Ez azért van, mert a rendszermag fordítása sok szabad helyet igényel az objektum-fájlok, mint a *.o, számára. Ezért az /usr/src/linux könyvtárnak egy szimbolikus hivatkozásnak (soft link) KELL lennie, ami arra a könyvtárra mutat, ahol a forrás van.

Ezek után lásd a következő részt, a programfordításról és a telepítésről.

2.3. Az új kiadás változásai: dokumentáció

Ha letöltötted az új rendszermag forrását, olvasd el a következő fájlokat az /usr/src/linux/Documentation könyvtárban

- Changes: ebben olvashatók a minimális követelmények valamint néhány megjegyzés
- kernel.txt: ebben van a "Linux-rendszermag magyarázata"
- Olvasd el az összes *.txt fájlt az /usr/src/linux/Documentation könyvtárban, de legfontosabbak a kernel*.txt fájlok.

2.4. A türelmetleneknek

1. Csomagold ki a forrást
2. cd /usr/src/linux; mkdir /usr/src/kernelconfigs ; cp /usr/src/linux/.config /usr/src/kernelconfigs/.config.save; make clean; make mrproper; make clean; cp /usr/src/kernelconfigs/.config.save .config # Akkor, ha újra akarod használni a beállítófájlt ??
3. Opcionális - másold át a konfigurációs fájlt: átmásolhatod a beállítófájlt a régi rendszermag-forrásfájból az újba (időt spórolhat, ha a régi beállításokat akarod használni).
4. make xconfig # Szebb, de korlátai vannak; csak "X" alatt fut
5. make dep
6. Adj egy egyedi nevet az új rendszermagnak - szerkeszd az /usr/src/linux/Makefile fájlt és az EXTRAVERSION bejegyzést

7. nohup make bzImage
8. "make modules" és "make modules_install"
9. ...és elmehetsz vacsorázni vagy le is fekhetsz (szép Linuxos álmokat!), majd amikor visszajössz a rendszer kész! Nézd meg a naplófájlt a "less nohup.out" paranccsal.
10. make install # Ez "NEM ajánlott" - használd a "cp /usr/src/linux/arch/i386/boot/bzImage /boot/bzImage.myker" parancsot
Másold át a System.map fájlt a /boot könyvtárba: # cp System*.map* /boot/
Másold át a rendszermag konfigurációs fájlt (az általad kiválasztott opciókat). Nem elengedhetetlen, de később hasznos lehet. cp .config /boot/config.KERNEL_VERSION.Name
11. Állítsd be a GRUB vagy a LILO rendszerbetöltőt.
12. Indítsd újra a gépet, és ellenőrizd az új rendszermag betöltődését
13. Készíts biztonsági mentőlemezt a "bzdisk" vagy "mkbootdisk" paranccsal
14. Opcionális - make rpm # RPM csomag készítéséhez
15. Opcionális - make clean (ha tárhelyet akarsz felszabadítani)

Lásd a fenti lépések részleteit a következő fejezetekben...

2.5. Új rendszermag készítése - a lépések értelmezése

Az előző részben említett lépések részletei:

Figyelem: Az alábbiakban a "bash#" jelzi a bash készenléti jelét (prompt), azokat a parancsokat kell begépelni, amiket a "bash#" jelzés után következnek. A következő parancsokat RedHat Linux alatt, a 2.4.7-10 verziójú rendszermaggal teszteltük, de más terjesztéseknél is (esetleg kisebb eltérésekkel működni kell. Ugyanúgy működni kéne a régi, 2.2, 2.0 és 1.3 verziójú rendszermagokkal is. Szintúgy a jövőbeli vagy új verzióúakkal (kis eltérésekkel - értesítetek róla).

- *Figyelem:* Sok rendszermagot (kernel image) tárolhatsz a rendszereden. A következő lépéseket követve nem írod felül vagy teszed tönkre a meglévő rendszermagodat. Ezek a lépések *nagyon biztonságosak* és a jelenlegi rendszermagot nem érintik, azt nem módosítják.

1. *Csomagold ki a forrást:* A következő lépések végrehajtása során "root" felhasználóként legyél bejelentkezve. Csatold fel a RedHat Linux CD-ROM-ot (a RedHat 9-ben Disc 2) és telepítsd a rendszermag forrás-RPM csomagját:

```
bash$ su - root
bash# cd /mnt/cdrom/RedHat/RPMS
bash# rpm -i kernel-headers*.rpm
bash# rpm -i kernel-source*.rpm
bash# rpm -i dev86*.rpm
bash# rpm -i bin86*.rpm
```

A bin86*.rpm és az "as86" csak a *RÉGEBBI Linux* verziókhoz kell, mint a Redhat 5.x. Szerezd be az "as86" Intel assembler csomagot a dev86*.rpm-ből a CD-ROM-ról vagy a bin86-mandrake (<http://rpmfind.net/linux/RPM/mandrake/7.1/Mandrake/RPMS/bin86-0.4-12mdk.i586.html>) honlapról, bin86-kondara (<http://rpmfind.net/linux/RPM/kondara/jirai/i586/bin86-0.4-8k.i586.html>). A legújabb, 9-es RedHat kernel-source.rpm csomagja tartalmazza a rendszerem fejlesztőfájlokat. Általában a forrás a RedHat lemezeken vagy a SRPM lemezeken található, egyébként beszerezheted a rpmfind.net kernel-source (<http://rpmfind.net/linux/rpm2html/search.php?query=kernel-source&submit=Search+...&system=&arch=>) (keresd a RawHide-ot a legújabb RedHat-hez) és rpmfind.net kernel-headers (<http://rpmfind.net/linux/rpm2html/search.php?query=kernel-headers&submit=Search+...&system=&arch=>) honlapról. Bizonyosodj meg arról is, hogy az /usr/src/linux egy szimbolikus hivatkozás, ami a megfelelő kicsomagolt forrásra mutat.

```
bash# cd /usr/src
bash# ls -l      # Láthatod, hogy az /usr/src/linux egy szimbolikus hivatkozás, ami a forrásra mutat
lrwxrwxrwx    1 root    root          19 Jan 26 11:01 linux -> linux-2.4.18-19.8.0
drwxr-xr-x   17 root    root        4096 Jan 25 21:08 linux-2.4.18-14
drwxr-xr-x   17 root    root        4096 Mar 26 12:50 linux-2.4.18-19.8.0
drwxr-xr-x    7 root    root        4096 Jan 14 16:32 redhat
```

Ha nem szimbolikus hivatkozás, akkor nevezd át az /usr/src/linux könyvtárat /usr/src/linux-2.4.yy könyvtárra és készíts egy szimbolikus hivatkozást.

FIGYELEM: ha nincs elég szabad hely az /usr/src alatt, akkor bármelyik partícióra kicsomagolhatod a forrást, ahol van elegendő (mint a /home). Ez azért van, mert a rendszerem fordítása sok szabad helyet igényel az objektum-fájlok, mint a *.o, számára. Ezért az /usr/src/linux könyvtárnak egy szimbolikus hivatkozásnak KELL lennie, ami arra a könyvtárra mutat, ahol a forrás van.

2. *Takarítás:* Mielőtt az "mrproper" parancsot kiadnánk, elmentheted a .config fájlt.

```
bash# cd /usr/src/linux
      bash# mkdir /usr/src/kernelconfigs ;
      bash# cp /usr/src/linux/.config /usr/src/kernelconfigs/.config.save;
bash# cp /usr/src/linux/.config /usr/src/linux/configs/.config.save # Különösen biztonságos
bash# cp /boot/config* /usr/src/linux/configs/ # Különösen biztonságos
bash# make clean
bash# make mrproper # "EL KELL VÉGEZNEDE ezt az mrproper-t", különben rengeteg problémával szembesülsz
bash# make clean
bash# cp /usr/src/kernelconfigs/.config.save .config # ha újra akarod használni a beállítófájlt
```

3. *Opcionális - másold át a beállítófájlt:* Ez a lépés időt spórolhat meg, ha fel akarod használni a régi beállításokat. Amikor telepítéd a rendszerem, akkor a beállítófájlt általában a /boot alá rakod. Van néhány alapszintű beállítófájl, az /usr/src/linux/configs/kernel-*.config, amiket felhasználhatsz. Tehát használhatod a meglévő konfigurációs fájlt:

```
      bash# mkdir /usr/src/kernelconfigs ;
      bash# cp /usr/src/linux/.config /usr/src/kernelconfigs/.config.save;
bash# cp /usr/src/linux/.config /usr/src/linux/configs/.config.save # Különösen biztonságos
bash# cp /boot/config* /usr/src/linux/configs/ # Különösen biztonságos
bash# cp /boot/config-2.4.18-19.8.0 /usr/src/linux/.config
```

Vagy másold át az alapértelmezett beállítófájlt az /usr/src/linux/configs könyvtárból:

```
bash# cp /usr/src/linux/configs/kernel-2.4.18-i686.config /usr/src/linux/.config
Vagy Athlon processzorok esetében
bash# cp /usr/src/linux/configs/kernel-2.4.18-athlon.config /usr/src/linux/.config
```

Esetleg egy másik módszer - átmásolhatod a .config fájlt a régi rendszermag-forrásfából az újba.

```
bash# ls -l /usr/src/lin* # Látható, hogy az /usr/src/linux egy szimbolikus hivatkozás
bash# cd /usr/src/linux
bash# cp ../linux-old-tree/.config . # Például cp ../linux-2.4.19/.config .
```

Sőt egy másik módszer - használhatod a "make oldconfig" parancsot, ami alapértelmezésként felhasználja a meglévő .config fájlt a beállítás kérdéseire.

4. Beállítás:

- Indítsd el az X-Window rendszert a "startx" paranccsal. Ha nem tudod elindítani, lásd alább a következő lépést.

```
bash# man startx
bash# startx
bash# cd /usr/src/linux
bash# make xconfig # Szebb, bár korlátozott; csak "X" alatt fut
```

Ha hibaizenetet kapsz, miszerint a "wish" nem található, telepítsd a tk-8.3.rpm csomagot.

- Ha nem tudod elindítani az "X"-et, akkor próbáld meg a következőt:

```
bash# export TERM=xterm
bash# make menuconfig # újabb, az ncurses/curses csomagot használja, nem működik, ha nincs telnet
Ha összezavarodott képernyőt kapsz, használj más terminál-emulációt,
mint a vt100, vt102, vt220 vagy ansi. A képernyő zavaros lesz és
összevissza karaktereket látsz, ha telnet-tel lépsz be egy távoli Linux
rendszerbe. Ebben az esetben a vt100, vt220 terminál-emulátorokat használd.
Például:
bash# export TERM=vt220
bash# export TERM=ansi
A VT alacsonyabb szintjein használd:
bash# export TERM=vt100
bash# make menuconfig # újabb, az ncurses/curses csomagot használja, nem működik, ha nincs telnet
Ha a "menuconfig" parancs nem működik, akkor próbáld meg:
bash# make config # régi, felhasználóbarát módszer !!
```

A "make xconfig" vagy "make menuconfig" egy felhasználóbarát GUI felületet indít el. A "make config" egy parancssoros, konzolos felületet. Betöltheted a beállítófájlt az /usr/src/linux/.config fájlból ("pont config" fájl. Figyelj a pontra a "config" előtt). Kattints a "Load configuration from file" (Beállítások betöltése fájlból) gombra. A "make xconfig"-on belül (a problémák kivédése érdekében) a következőket kell tenned:

- **NAGYON FONTOS !!!** : Válaszd a megfelelő CPU típust - Pentium 3, AMD K6, Cyrix, Pentium 4, Intel 386, DEC Alpha, PowerPC, egyébként a rendszermag fordítása nem sikerül, vagy ha igen, akkor nem fog elindulni!!
- Válaszd az SMP támogatást - akár egy, akár több processzorod van

- Fájlrendszerek - válaszd a Windows95 VFAT, MSDOS, NTFS fájlrendszereket mint a rendszermag részeit, és ne mint betölthető modulokat. (Nekem személy szerint előnyös, de szabadon válaszd a saját rendszerednek megfelelően).
- Engedélyezd a "Loadable kernel modules" (betölthető rendszermag modulok) támogatást! Ezzel az opcióval, a futó Linux rendszeren tudod az eszközmeghajtókat dinamikusan ki/be tölteni. Részletesen a Betölthető modulok (#loadable_modules) fejezetben olvashatsz erről.

Ments és lépj ki a "make xconfig"-ból. Minden kiválasztott beállítási opció elmentődött az `/usr/src/linux.config` (pont config) fájlba.

5. *Függőségek:* Most add ki a következő parancsot:

```
bash# make dep
```

6. *Adj egyedi nevet az új rendszermagodnak:* Nevet is adhatsz a rendszeragnak, így az egyedi lesz és nem akad össze a többivel.

```
bash# cd /usr/src/linux
bash# vi Makefile
```

Itt keresd meg az `EXTRAVERSION = -19.8.0_blabla` részt és változtasd meg például így: `EXTRAVERSION = -19.8.0MyKernel.26Jan2003`

7. *Add ki a "make" parancsot:* Olvasd el a következő fájlt (hogy információt gyűjts a rendszermag fordításáról/építéséről. Tipp: használd a színes `gvim` (<http://www.tldp.org/HOWTO/Vim-HOWTO.html>) szövegszerkesztőt a jobb olvashatóság érdekében.

```
bash# gvim -R /usr/src/linux/arch/i386/config.in
bash# man less
bash# less /usr/src/linux/arch/i386/config.in
Nyomd le a "h" billentyűt a sűgóért és navigálj a szövegben az i, j, k, l, h
vagy a nyíl-, ill. a page up/down billentyűkkel.
```

Most add ki a "make" parancsot:

```
bash# cd /usr/src/linux
bash# man nohup
bash# nohup make bzImage &
bash# man tail
bash# tail -f nohup.out (... a folyamat figyeléséhez)
Ez a rendszermagot az /usr/src/linux/arch/i386/boot/bzImage fájlba helyezi el.
```

8. **BETÖLTHETŐ MODULOK:** Most amíg a "make" végigzörög az előző rész szerint, elindíthatsz egy másik xterm-es parancsértelmezőt és kövesd a lépéseket: ez a lépés **CSAK AKKOR** szükséges, ha engedélyezted a "loadable module support"-ot a fenti "Beállítás" lépésnél. A betölthető modulok a `/lib/modules` alatt találhatóak. **EL KELL** végezned ezt a lépést, ha engedélyeztél vagy letiltottál bármely modult, egyébként "unresolved symbols" hibáüzeneteket fogsz kapni a rendszermag betöltődése közben vagy azután.

```
# indíts egy új xterm ablakot és ...
bash# cd /usr/src/linux
# irányítsd át a kimenetet, hogy ne írd felül a nohup.out-ot, amíg az fut...
bash# nohup make modules 1> modules.out 2> modules.err &
```

```
bash# make modules_install # csak akkor add ki, ha a fenti "make" parancs sikeresen lefutott
```

Ez bemásolja a modulokat a /lib/modules könyvtárba. Részletesen a Betölthető modulok (#loadable_modules) fejezetben olvashatsz erről.

9. *Most menj vacsorázni vagy feküdj le:* Amíg mindkét "make"-ablak elzörög magában, addig elmehetsz vacsorázni (pihenj picit) vagy bújj ágyba (szép linuxos álmokat), mire felébredsz és visszajössz, a rendszer kész lesz! Ellenőrizheted a kimenet naplóját a "less nohup.out" paranccsal.

```
bash# cd /usr/src/linux
bash# less nohup.out
bash# less modules.err
bash# less modules.out
Ha nem volt hiba:
bash# make modules_install
```

10. *bzImage:* Miután a bzImage sikerült, másold át a rendszermagot (kernel image) a /boot könyvtárba. Át kell másolnod az új fájlt, különben az új rendszermag *LEHET, HOGY NEM FOG* betöltődni. Ezen kívül a beállítófájlt is át kell másolnod a /boot területre, hogy tükrözze a rendszermag beállításait dokumentációs szempontból.

```
bash# cp /usr/src/linux/arch/i386/boot/bzImage /boot/bzImage.myker.26mar2001
# Át KELL másolnod a beállítófájlt, hogy tükrözze a megfelelő rendszermagot
# dokumentációs célból
bash# cp /usr/src/linux/.config /boot/config-<your_kernelversion_date>
# Példa: cp /usr/src/linux/.config /boot/config-2.4.18-19.8.0-26mar2001
```

FIGYELEM: Ha az initrd használatát tervezed a LILO vagy a GRUB rendszerbetöltőben, akkor lefordíthatod és elhelyezheted azt a /boot/initrd*.img fájlban. Részletesen az "A" függelék - initrd.img fájl készítése (#create_initrd) fejezetben olvashatsz erről.

11. *A LILO vagy a GRUB beállítása:* Két lehetőség van a betöltésre RedHat Linux alatt - a GRUB és a LILO. *A GRUB beállítása:* A GRUB újabb és sokkal jobb eszköz mint a LILO, elsődlegesen ennek a használatát ajánlom. A LILO régebbi technológia. A GRUB különbözik a többi betöltőtől (mint például a LILO) abban, hogy ez *"képes azt hazudni a MS Windowsnak (és elhítenni vele), hogy az első partícióra telepítették, akkor is ha nem így van!"* . Így megtarthatod a jelenlegi Linux rendszeredet ott, ahol van, és telepíthetsz Windowst mellé. Részletesen a "C" függelék - GRUB részletesen, grub.conf mintafájl (#grubconf) fejezetben olvashatsz erről. *A LILO beállítása:* A LILO régebbi eszköz, részletesen a "B" függelék - lilo.conf mintafájl (#liloconf) fejezetben olvashatsz róla.
12. Indítsd újra a gépet, és a lilo futásakor nyomd meg a TAB billentyűt, majd gépeld be: "myker". Ha elindul, jó munkát végeztél! Egyébként a lilo menüjében válaszd a régi rendszermagot, indítsd el és próbáld meg mindent előlről kezdeni. A régi rendszermagod még *mindig ÉRINTETLEN és BIZTONSÁGOS* , például a /boot/vmlinuz-2.0.34-0.6

13. Ha az új rendszermagod betöltődik, és jól is működik, elkészítheted a indítólemezed. Tégy egy üres hajlékonylemezt a meghajtóba, majd:

```
bash# cd /usr/src/linux
bash# make bzdisk
Használható még az mkbootdisk parancs -
bash# rpm -i mkbootdisk*.rpm
```

```
bash# man mkbootdisk
```

14. Opcionálisan, készíts RPM csomagot: ha több gépre is telepíteni akarsz az új rendszermagot, RPM csomagot is készíthetsz belőle.

```
make rpm # RPM csomag építéséhez
```

15. *Takarítás:* Opcionális: make clean (ha tárhelyet akarsz felszabadítani)

2.6. Hibaelhárítás

Problémád van? Olvasd el az "E" függelék - a gyakori hibák elhárítása (#troubleshoot) fejezetet.

2.7. A rendszermaghoz kapcsolódó utólagos fordítások

Olvasd el az "D" függelék - utólagos rendszermag fordítás (#post_kernel_building) fejezetet.

2.8. Hová küldjem a hibajelentést?

Ha a fenti lehetőségek mindegyikét kipróbáltad és úgy gondolod, hogy a rendszermag hibás. (Ez a fejezet eredetileg az "E" függelék - a gyakori hibák elhárítása () fejezet végén volt, tehát először az ott leírt lehetőségeket próbáljátok végig - a lektor) Ebben az esetben jelentheted a hibát, így (talán) javítva lesz. Olvasd el az /usr/src/linux/REPORTING-BUGS fájlt, majd látogass el a Reporting Bugs for the Linux Kernel (<http://www.kernel.org/pub/linux/docs/lkml/reporting-bugs.html>) (A Linux-rendszermag hibáinak bejelentése) honlapra.

3. Betölthető modulok

Mindenki, aki használta a "betölthető modulokat", igazán "megszerette" őket! Szeretem a modulokat, mivel agyafűrt dolgok, és a nagyobb feladatokat kicsi, kezelhető részekre szedik szét. Ha elkezded használni, fogadok, hogy megszereted őket! A modulok csak a Linuxra jellemzőek. Ez volt a világ első olyan operációs rendszere, ami bemutatta a betölthető modulok fogalmát. Egyetlen más operációs rendszer sem rendelkezik ezzel a fogalommal, de a közeljövőben más rendszerek is megvalósíthatják.

A betölthető modulok memóriát spórolhatnak, és megkönnyítik a beállítást. A modulok hatóköre magában foglalja a fájlrendszereket, ethernet kártyák, szalagos meghajtók, nyomtatók és mások eszközmeghajtóit.

A modulok a rendszermag kódjának darabjai, amelyeket nem csatoltak (foglaltak bele) közvetlenül a bele. Az ember lefordíthatja őket elkülönítve is, és beillesztheti vagy eltávolíthatja a futó rendszermagból majdnem bármikor. Rugalmasságából kifolyólag nem ajánlott bizonyos rendszermag-jellemzőket beléjük kódolni. Sok népszerű eszközmeghajtó, mint például a PCMCIA és a QIC-80/40 szalagos eszközmeghajtók, betölthető modulok.

Olvasd el a Module-HOWTO (Modul HOGYAN) doksit: "<http://www.tldp.org/HOWTO/Module-HOWTO>" (<http://www.tldp.org/HOWTO/Module-HOWTO>) .

Olvasd el ezeket a kézikönyv-oldalakat is:

```
bash# rpm -i /mnt/cdrom/Redhat/RPMS/modutils*.rpm
bash# man lsmod
bash# man insmod
bash# man rmmod
bash# man depmod
bash# man modprobe
```

Például, ha be akarsz tölteni a `/lib/modules/2.4.2-2/kernel/drivers/block/loop.o` modult, tedd ezt:

```
bash# man insmod
bash# modprobe loop
bash# insmod loop
bash# lsmod
```

Az `insmod` keresési útvonalát (`PATH`) az `/etc/modules.conf` fájlban állíthatod be.

3.1. A modulok segédprogramjainak telepítése

A modulok segédeszközeinek RPM csomagját így telepítheted:

```
bash# rpm -i /mnt/cdrom/Redhat/RPMS/modutils*.rpm
```

Az `insmod` beilleszti a modult a futó rendszerembe. A modulok általában a `.o` kiterjesztéssel rendelkeznek; a példaként említett neve `drv_hello.o`, így ennek beillesztéséhez kiadjuk az: `" insmod drv_hello.o "` parancsot. A rendszerem által jelenleg használt modulok listájához használd az `lsmod` parancsot. A kimenet valami ilyesmi: `blah# lsmod Module: #pages: Used by: drv_hello 1 " drv_hello "` a modul neve, egy lapnyi (4k) memóriát használ, és jelenleg nincs más modul, ami tőle függ. Eltávolításához használd az `" rmmod drv_hello "` parancsot. Figyeld meg, hogy az `rmmod` egy *modulnevet*, nem pedig egy fájlnevet igényel; ezt kinyerheted az `lsmod` listájából. A többi modul-segédeszköz céljait a kézikönyv-oldalaik írják le.

3.2. A rendszeremmel szállított modulok

A 2.0.30-as verziótól kezdve, majdnem minden elérhető betölthető modul formájában. A használatukhoz először bizonyosodj meg, hogy nem állítod be őket a rendes rendszeremben; tehát nem válaszolsz `y`-el ezekre a `" make config "` alatt. Fordíts egy új rendszerem és indítsd el. Aztán: `cd` az `/usr/src/linux` könyvtárba ismét, és: `" make modules "`. Ez lefordítja az összes modult, amit nem adott meg a rendszerem konfigurációjában, és hivatkozást (link) készít hozzájuk az `/usr/src/linux/modules` könyvtárba. Használhatod őket egyenesen abból a könyvtárból vagy hajtsd végre a `" make modules_install "` parancsot, ami telepíti őket a `/lib/modules/x.y.z` könyvtárba, ahol `x.y.z` a rendszerem verziója.

Ez főleg a fájlrendszereknél lehet kényelmes. Valószínűleg nem használod gyakran a minix vagy a msdos fájlrendszert. Például, ha belebotlok egy msdos (iszonyat!) hajlékonylemezbe, csak beírom: `insmod /usr/src/linux/modules/msdos.o`, és aztán `rmmod msdos`, amikor végeztem. Ez az eljárás körülbelül

50k RAM-ot spórol a rendszermagban a normál működés során. Egy kis megjegyzés kívánkozik ide a minix fájlrendszer kapcsán: *mindig* közvetlenül a rendszermagba állítsd be, ha "mentőlemezen" akarsz használni.

3.3. Hogyan telepítsek csak egyetlen modult?

Tegyük fel, hogy már megcsináltad a "make modules" és "make modules_install" lépéseket. Később pedig még a "make clean"-t is a helyfelszabadítás érdekében. Most pedig változtatni akarsz valamelyik modul egyetlen C fájlján, és újra akarsz építeni azt az egy modult, és a modulfájlt bemásolni a /lib/modules alá. Hogyan csinálod? Hiszen nem akarsz egy "make modules" parancsot kiadni, az mindent újraépít és 2-3 óráig tart!

Le tudsz fordítani egyetlen modulfájlt (mondjuk a foo.o-t) és telepítheted is. Ehhez egyszerűen szerkeszd meg a Makefile-t, és változtasd meg a SUBDIRS bejegyzést csakis azokra a könyvtárakra, amik érdekelnek.

Egy jó példa: azt vettem észre, hogy a rendszermagom nem támogatja az NTFS fájlrendszert (persze kiadtam a "make clean"-t a "make modules" után. A francba!). Így elhatároztam, hogy lefordítom az NTFS betölthető modulját. Nem akartam az egész készletet (mivel ez nálam 2 óráig tartana), ezért az alábbi módszert követtem, és csak a fs/ntfs modult fordítottam, majd kiadtam az "insmod ntfs" parancsot. Az egész 5 percig tartott!

Másik példa: ha csak az fs/autofs modul érdekel, akkor a következőket teszem:

```
cd /usr/src/linux
mv Makefile Makefile.original
cp Makefile.original Makefile.my
ln -s Makefile.my Makefile # mivel néhány fájlnak még szüksége lehet a "Makefile"-ra
vi Makefile.my
# Megjegyzésekkel láttam el a "SUBDIRS" sorokat, és hozzáadtam azokat a könyvtárakat, amik érdekelték
# például a fs/autofs-hez:
#SUBDIRS =kernel drivers mm fs net ipc lib abi crypto
SUBDIRS =fs/autofs
# Elmentettem a Makefile.my fájlt és kiadtam a következő parancsot:
make -f Makefile.my modules
# Ez megcsinálja az autofs.o modult
# Most bemásoltam a modul objektumfájlt a célkönyvtárba: /lib/modules
# VESZÉLYES: NE CSINÁLD EZT: "make -f Makefile.my modules_install" , mivel kitakaríthatja
# a többi jót is a /lib/modules könyvtárból !! Csak másold be, így:
cp autofs.o /lib/modules/2.4.18-19.8.0/kernel/fs/autofs

# Most állítsunk vissza mindent a normál helyzetbe
rm Makefile # This is a link to Makefile.my
ln -s Makefile.original Makefile

# Rögzítsd a változtatásokat a beállítófájlban a későbbi használat érdekében
# Szerkeszd az /usr/src/linux/.config fájlt és állítsd be modulként a szolgáltatást
cd /usr/src/linux
    mkdir /usr/src/kernelconfigs ;
    cp /usr/src/linux/.config /usr/src/kernelconfigs/.config.save;
cp /usr/src/linux/.config /usr/src/linux/configs/.config.save # Különösen biztonságos
cp /boot/config* /usr/src/linux/configs/ # Különösen biztonságos
vi /usr/src/linux/.config
# és változtasd meg a beállító paramétert. Például az én esetemben
# az ntfs modulnál CONFIG_NTFS_FS=m jelzi, hogy ez modul.
```

Többet is megtudhatsz a Makefile-ról és a make-ről, ha elolvasod a GNU make kézikönyv oldat:

- "<http://www.gnu.org/manual/make>" (<http://www.gnu.org/manual/make>) .
- University of Utah Makefile "http://www.math.utah.edu/docs/info/make-stds_toc.html" (http://www.math.utah.edu/docs/info/make-stds_toc.html)
- University of Hawaii Makefile "<http://www.eng.hawaii.edu/Tutor/Make>" (<http://www.eng.hawaii.edu/Tutor/Make>)
- A Linuxban - man make
- A Linuxban - info make

Ismerkedj meg a modulokat elkészítő Makefile-al. A fájlban van egy "module" sora, mint:

```
modules: $(patsubst %, _mod_%, $(SUBDIRS))
```

A patsubst függvény szintaktikája \$(patsubst minta,helyettesítés,szöveg). A százalékjelet ([percent]) ugyanúgy használja, mint a mintaillesztő szabályok - mint egy sztringet, ami illeszkedik a mintában és a helyettesítő szövegben is. Végigkeresi a "szöveg"-et elválasztójellel tagolt szavakért, amik illeszkednek a "mintá"-ra és kicseréli az előfordulásait a "helyettesítés"-re.

Ez a Makefile ugyanúgy tartalmaz szabványos shell-függvényeket, mint szabványos make függvényeket. A shell-függvények szintaktikája \$(shell parancs). Ez a függvény kimenetét adja vissza (az új sorok kihúzásával).

4. A rendszermag "klónozása"

Lehet, hogy készíteni akarsz egy rendszermagot egy adott rendszeren, majd ezt sok hasonló PC-re is akarsz nagyüzemben telepíteni. Az újonnan készített rendszermagnak a gépek százaira való egyszerűbb telepítése érdekében RPM (RedHat) vagy DEB (Debian) csomag formájában akarsz azt elkészíteni, vagy egyszerűen csak tar.gz fájl formájában.

1. RPM csomagot készíthetsz a rendszermagból az rpmbuild -ba kernel*.spec paranccsal
2. Ellenőrizd, hogy az elkészült kernel*.rpm minden, a /lib/modules/2.x.x-y könyvtárban lévő fájlt tartalmaz. Egyébiránt össze kell csomagolnod a tar és gzip segítségével ezt a könyvtárat, és átvinni a célgépekre.
3. Ellenőrizd, hogy a rendszermag-csomag tartalmazza a /boot/initrd-2.x.x-y.img fájlt, különben össze kell csomagolnod a tar és gzip segítségével, és átvinni a célgépekre.
4. Ezen felül a /boot könyvtár más fájljait is, amik nincsenek a kernel*.rpm csomagban.

5. Fontos kérdések és válaszok

5.1. Egyébként mit is csinál a rendszermag?

A Unix rendszermagja közvetítőként viselkedik a programok és a hardver között. Először is, elvégzi (vagy elrendezi) a memóriakezelést az összes futó program (folyamat) számára, és biztosítja, hogy mindegyik igazságosan (vagy éppen nem annyira igazságosan, ha engeded) osztozzon a processzor ciklusain. Ezen felül egy szép, meglehetősen hordozható felületet biztosít a programok felé, hogy "beszélgethessenek" a hardverrel.

Biztosan van még több minden is, amit el lehet mondani a működéséről, de ezek az alapvető funkciók a legfontosabbak amikről tudni kell.

5.2. Miért kéne frissítenem a rendszermagomat?

Az újabb rendszermagok általában több fajta hardver kezelését teszik lehetővé (vagyis több eszkövezérlőjük van), jobb folyamatvezérlésük lehet, gyorsabbak és stabilabbak lehetnek, mint a régi verziók és azok buta kis hibáit kijavíthatják. A legtöbben azért frissítenek, mert új eszkövezérlőket és hibajavításokat akarnak.

5.3. Milyen hardvert támogatnak az új verziójú rendszermagok?

Olvasd el a Hardware-HOWTO (<http://www.tldp.org/HOWTO/Hardware-HOWTO>) (Hardver HOGYAN) dokumentumban. Ezen kívül megnézheted a " `config.in` " fájlt a Linux forráskódjában, vagy egyszerűen keresd ki a " `make config` " kiadásával. Ez megmutatja az összes, standard rendszermag által támogatott hardvert, de nem az összeset, amit a Linux támogat; sok elterjedt eszkövezérlőt (mint a PCMCIA vezérlők és néhány szalagos meghajtó) betölthető modulként tartanak karban és külön terjesztenek.

5.4. Milyen verziójú gcc és libc kell?

Linus ajánl egy adott verziójú gcc-t a `README` fájlban, ami a Linux forráskód része. Ha nincs meg ez a verzió, az ajánlott verziójú gcc dokumentációja jelzi, ha frissíteni kell a libc programkönyvtárat. Ez nem bonyolult eljárás, de fontos az útmutatások követése.

5.5. Mi az a betölthető modul?

Részletesen a Betölthető modulok (`#loadable_modules`) fejezetben olvashatsz erről.

5.6. Mennyi lemezterületre van szükségem?

Ez függ az egyéni rendszer-beállításaidtól. Először is, a tömörített Linux forrás közel 14 MB méretű, legalábbis a 2.2.9 verzió. Sok helyen megtartják ezt is, miután kicsomagolták. Kicsomagolva és egy nem túl bonyolult beállítással, még további 67 MB-ot foglal.

5.7. Mennyi időt vesz igénybe?

Az újabb gépekkel a fordítás drámaian kevesebb időt igényel, mint a régieken; egy AMD K6-2/300 egy gyors merevlemezzel a 2.2.x rendszermagot elkészíti körülbelül 4 perc alatt. Viszont az öreg Pentiumokkal, 486-osokkal és 386-osokkal, ha úgy tervezed, hogy nekiállsz fordítani, várhatsz akár órákat, napokat...

Ha ez zavar téged, és éppen van egy gyorsabb gép a közelben, amin fordíthatsz, akkor lefordíthatod azon (feltéve, hogy jó adatokat adsz meg, a segédprogramjaid naprakészek stb.), majd átviheted a rendszermagot a lassabb gépre.

6. A rendszermag foltozása (patching)

6.1. Egy folt alkalmazása

A rendszermag lépésenkénti fejlesztésének eszköze a foltozás. Például, ha v1.1.45-ös Linuxod van, és észreveszed, hogy létezik egy " patch46.gz " fájl ehhez, ez azt jelenti, hogy frissítheted az 1.1.46 verzióra a folt alkalmazásával. Először biztonsági másolatot ajánlott készíteni a forrásfájról (" make clean " majd ezután " cd /usr/src; tar zcvf old-tree.tar.gz linux ", ami egy tömörített tar archívumot készít).

Tehát folytatva a fenti példát, tegyük fel, hogy van egy " patch46.gz " fájl az /usr/src könyvtárban. Lép be cd az /usr/src könyvtárba és add ki a " zcat patch46.gz [verbar] patch -p0 " (vagy " patch -p0 [lt] patch46 " parancsot, ha a folt nincs tömörítve). Látni fogod, ahogy a feliratok elzúgnak (vagy elcsattognak, ha lassabb géped van), jelezve, hogy megpróbálják a kóddarabokat beilleszteni, és hogy ez sikerült-e vagy sem. Általában ez a folyamat túl gyors ahhoz, hogy elolvashasd és nem lehetsz biztos benne, hogy működött-e, ezért használd a -s kapcsolót a patch programhoz, ami azt jelzi a patch programnak, hogy csak a hibáüzeneteket írja ki (nem fogsz sokat kapni a "hé, a komputerem éppen valami változtatást csinál!" érzésből, de lehet, hogy épp ezt értékeled...) Azon dolgok megtekintéséért, amik nem mentek simán, lépj be az

/usr/src/linux könyvtárba és keress .rej kiterjesztésű fájlokat. A patch régebbi verziói (azok a verziók, amiket alsóbbrendű fájlrendszeren fordítottak) a visszadobott dolgokat # kiterjesztéssel látják el. Használhatod a " find " parancsot, hogy keressen helyetted; a "find . -name '*.rej' -print" kiírja a standard kimenetre az összes .rej kiterjesztésű fájlt, ami az aktuális könyvtárban és alkönyvtáraiban van.

Ha minden rendben zajlott, adj ki egy " make clean ", " config ", and " dep " parancssorozatot, a 3. és 4. részben leírtak szerint.

Elég kevés kapcsolója van a patch parancsoknak. Ahogy fentebb említettem, a patch -s minden üzenetet elnyom, kivéve a hibáüzeneteket. Ha valahol másutt tartod a rendszermag forrását, nem az /usr/src/linux könyvtárban, a patch -p1 (abban a könyvtárban) rendben megfoltozza a dolgokat. Egyéb patch kapcsolókat a jól dokumentált kézikönyv oldalakban keress.

6.2. Ha valami nem sikerül

(Figyelem: ez a rész leginkább a meglehetősen régi rendszermagokra vonatkozik)

A leggyakrabban előforduló probléma az volt, amikor egy folt módosította a " config.in " fájlt és az nem jól nézett ki, mivel megváltoztattad a beállításokat, hogy megfeleljenek a gépednek. Ezt már javították, de még belefuthatsz, ha régi rendszermagot fordítasz. A kijavításához nézd meg a config.in.rej fájlt, hogy mi maradt az eredeti foltból. A változásokat általában " + " és " - " karakterek jelzik a sorok elején. Nézd meg a szomszédos sorokat, és jegyezd meg, hogy " y " -al vagy " n " betűvel vannak jelölve. Most szerkeszd a

`config.in` fájlt, és változtasd meg az `" y "-t" n "-re és az " n "-et" y "-ra`, amikor szükséges. Adj ki egy `"patch -p0 < config.in.rej"` parancsot és ha azt jelzi, hogy sikerült, akkor folytathatod a beállítást és fordítást. A `config.in.rej` fájl ottmarad, de törölhető.

Ha további problémákba ütközöl, akkor használaton kívüli foltot telepítettél. Ha azt mondja, hogy `" previously applied patch detected: Assume -R? "` (előzőleg már alkalmazott foltot észleltem), akkor valószínűleg olyan foltot próbáltál alkalmazni, ami a jelenlegi verziószám alatti; ha `" y "-t` válaszolsz, megpróbálja visszaállítani a forrásodat, és nagy valószínűséggel ez nem sikerül; ezért egy teljesen új forrásfárra van szükséged (ami nem is olyan rossz ötlet első nekifutásra).

A folt visszavonásához használd a `" patch -R "` parancsot az eredeti folton.

A legjobb dolog, ha a foltozás tényleg rossz irányba megy, hogy újratekedsz mindent egy tiszta, még nem használt forrásfával (például az egyik `linux-x.y.z.tar.gz` fájljal).

6.3. A .orig fájlaktól történő megszabadulás

Már néhány foltozás után a `.orig` fájlok elkezdnek szaporodni. Például, egy 1.1.51-es fátam valamikor még az 1.1.48 tájékaán takarítottam ki. A `.orig` fájlok törlésével majdnem fél MB hely szabadult fel. A `"find . -name '*.orig' -exec rm -f {} "; "` parancs gondoskodik erről. A `patch` azon verziói, amik a `#` karaktert használják a visszautasítások fájlneveiben, a tilde (`~`) karakterrel jelzik a `.orig` fájlokat.

Vannak jobb módszerek is a `.orig` fájlaktól történő megszabadulásra, amik függnek a GNU `xargs` parancstól: `"find . -name '*.orig' | xargs rm"` vagy az "elég biztonságos de kicsit bőbeszédűbb" módszer: `find . -name '*.orig' -print0 | xargs --null rm --`

6.4. Egyéb foltok

A Linus által terjesztett foltokon kívül léteznek mások is (én "nem szabványos" foltoknak hívom őket). Ha ezeket használod, a Linus-félék lehet, hogy nem működnek megfelelően. Elképzelhető, hogy vissza kell őket vonnod, kijavítani a forrást vagy a foltot magát, telepíteni egy új forrásfát vagy ezek kombinációját kell tenned. Ez nagyon bosszantó lehet, ezért ha nem akarod módosítani a forrást (egy nagyon rossz végeredmény lehetőségét vállalva), forgasd vissza a nem szabványos foltokat, mielőtt a Linus-féléket alkalmazod, vagy telepíts egy új forrásfát. Ezután láthatod, hogy a nem szabványos foltok működnek-e még. Ha nem, akkor vagy maradsz a régi rendszeragnál, játszatsz a folttal vagy a forrással, hogy működjön, vagy vársz (esetleg rimánkodszt érte) a folt új verziójáig.

Milyen gyakoriak azok a foltok, amik nincsenek a szabvány rendszeragnban? Valószínűleg hallottál róluk.

Valamikor használtam a "noblink" foltot a virtuális konzoljaimhoz, mert utálok a villogó kurzort. (Ezt a foltot (legalábbis régen) gyakran frissítették az új verziókhoz). Azonban mióta a legtöbb új eszközmeghajtót modulként fejlesztik, a "nem szabványos" foltok kibocsátási üteme jelentősen csökken.

7. Tippek és trükkök

7.1. A make vagy a patch parancs kimenetének átirányítása

Ha látni szeretnéd a naplókat, amiket a `" make "` vagy a `" patch "` parancsok csinálnak, akkor átirányíthatod a kimenetüket egy fájlba. Először nézd meg, milyen shellt használst: `" grep root /etc/passwd "` és keress

valami ehhez hasonlót: " /bin/csh ".

Ha sh-t vagy bash-t használsz, a (parancs) 2>&1 | tee (kimeneti fájl) utasítás sor készít egy másolatot a (parancs) kimenetéről a " (kimeneti fájl) "-ba.

A csh vagy tcsh esetén használd a (parancs) |& tee (kimeneti fájl) formát.

Az rc esetén (megjegyzés: valószínűleg nem ezt használod) : (parancs) >[2=1] | tee (kimeneti fájl)

7.2. Feltételes rendszermag-telepítés

A hajlékonylemezre történő telepítésen kívül még egyéb módszerek is léteznek az új rendszermag kipróbálására anélkül, hogy a régit bántatnánk. Sok egyéb Unix változattól eltérően, a LILO képes a rendszermagot a lemez bármely részéről betölteni (ha nagy méretű (500 MB vagy nagyobb) lemezed van, kérlek olvasd el a LILO dokumentációját, hogy ez mennyiben okozhat problémát). Ezért, ha az "image = /usr/src/linux/arch/i386/boot/bzImage label = new_kernel" sorokat berakod a LILO beállítófájljának végére, akkor választhatod az újonnan fordított rendszermagot is anélkül, hogy a régi /vmlinuz -hoz hozzányúlnál (természetesen miután futtatod a lilo parancsot). A legkönnyebben a betöltés közben a SHIFT lenyomásával lehet jelezni a LILO-nak, hogy az új rendszermagot töltsse be (amikor azt látod a képernyőn, hogy LILO , és semmi mást), ami ad egy készenléti jelet. Ennél a pontnál beírhatod: " new_kernel " az új rendszermag indításához.

Ha több különböző rendszermag-forrásfát akarsz tárolni egyszerre a gépen (ez azonban *rengeteg* lemezerületet fogyaszt; légy óvatos), a legtöbbször az /usr/src/linux-x.y.z könyvtárba kerülnek, ahol az x.y.z a rendszermag verziója. Ezután "kiválaszthatasz" egy forrásfát egy szimbolikus hivatkozás segítségével; például " ln -sf linux-1.2.2 /usr/src/linux ", ez az 1.2.2-t teszi aktuálissá. Mielőtt egy hasonló szimbolikus hivatkozást készítenél, bizonyosodj meg arról, hogy az ln utolsó paramétere nem egy létező könyvtár (régebbi szimbolikus hivatkozás elfogadható); különben az eredmény nem az lesz, amit várnál.

7.3. Rendszermag frissítések

Russell Nelson (nelson@crynwr.com) foglalja össze az új rendszermag kiadásoknál a változásokat. Ezek rövidek, és fejlesztés előtt átnézheted őket. Hozzáférhetőek anonymous FTP-vel az "ftp://ftp.emlist.com" (ftp://ftp.emlist.com) webhelyen, a pub/kchanges könyvtárban, vagy a "http://www.crynwr.com/kchanges" (http://www.crynwr.com/kchanges) webhelyen.

8. RPM csomagok felcsatolása (mount) az FTPFS segítségével

Mostanra lefordítottod az új rendszermagot és jól fut. Szükséged lehet arra a számtalan RPM csomagra, amiket lehet, hogy a közeljövőben telepítened kell majd. Ennek egyik útja, hogy fizikailag felcsatolod a LINUX CD-ROM-jait, de több mint 3 CD-ről van szó, kényelmetlen leválasztani és cserélni őket. Itt jön a képbe az FTPFS.

Ha automatizálni szeretnéd az RPM csomagok telepítését, fontold meg az "apt-get" parancs használatát. Az apt-get automatikusan feloldja a függőségeket, letölti és telepíti vagy frissíti a csomagokat. Részletesebben a Telepítés, frissítés fénysélességgel (#apt-get) fejezetben olvashatsz erről.

Az FTP fájlrendszer (FTPFS) egy Linux-rendszermag-modul, ami kibővíti a VFS hatókörét arra, hogy FTP köteteket is fel tudjon csatolni. Tehát fel tudsz csatolni FTP-n megosztott könyvtárakat a saját fájlrendszeredbe, és a helyi fájlok kezelésének előnyeit élvezheted. Megtalálható a "http://lufs.sourceforge.net/lufs" (http://lufs.sourceforge.net/lufs) és a "http://ftfps.sourceforge.net" (http://ftfps.sourceforge.net) webhelyen.

8.1. Az ftpfs használata

Töltsd le az ftpfs-t, és telepítsd a rendszeredre. Az ftpfs modulként települ: /lib/modules/2.4.18-19.8.0/kernel/fs/ftpfs/ftpfs.o. Ezen kívül az ftpmount parancs az /usr/bin/ftpmount könyvtárba. Ezután megteheted a következőt:

Jelentkezz be root felhasználóként (su - root) és futtasd ezt a szkriptet:

```
#!/bin/sh -x
# Használd ezt a szkriptet a redhat cdroms rpm-könyvtar (disk1,2,3) felcsatolásához
# Built rpm by name ftpfs.
# http://lufs.sourceforge.net/main/projects.html
# ftpmount --help
# Probáld ezt: ftpmount [user[:pass]@]host_name[:port][root_dir] mount_point [-o]
# [-uid=id] [gid=id] [fmask=mask] [dmask=mask]
#ftpmount anonymous:pass@ftp.kernel.org /mnt/ftpfs
#mkdir -p /mnt/ftpfs /mnt/ftpfs/updates /mnt/ftpfs/rpms /mnt/ftpfs/contrib
# Redhat ftp mirror sites - http://www.redhat.com/download/mirror.html
FTPSITE="csociety-ftp.ecn.purdue.edu"
USER="anonymous:pass"
ftpmount $USER@$FTPSITE/pub/redhat/redhat /mnt/ftpfs/site
ftpmount $USER@$FTPSITE/pub/redhat/redhat/linux/updates/8.0/en/os /mnt/ftpfs/updates
ftpmount $USER@$FTPSITE/pub/redhat/redhat/linux/8.0/en/os/i386/RedHat /mnt/ftpfs/rpms
ftpmount $USER@$FTPSITE/pub/redhat-contrib /mnt/ftpfs/contrib
```

8.2. Az ftpfs parancsai

Mielőtt egyáltalán elkezdenél az FTP kötetek felcsatolásának gondolatával játszani, győződj meg, hogy elegendő sávsebességed van-e, különben nem lesz nagy élvezet.

8.2.1. Az autofs-féle módszer - ki kell próbálnod!

Ha elég bölcs voltál, és telepítetted az autofs/automount szolgáltatást (nézd meg a telepítési feljegyzéseket), akkor nagyon fincsi módon használhatod az ftpfs-t: csak próbálg meg hozzáférni a kívánt szerveren bármelyik fájlhoz/könyvtárhoz az /mnt/ftpfs alatt.

```
cd /mnt/ftpfs/[user:pass@]ftp_server[:port]
```

Valami ilyesmit csinálj: cd /mnt/ftpfs/ftp.kernel.org. És mi történik? Már ott is vagy!

Normál esetben ezt csak anonymous FTP-vel használd, mivel nem akarhatod, hogy a felhasználói név/jelszó információid megjelenjenek az /mnt/ftpfs könyvtárában.

8.2.2. Az ftpmount-féle megoldás

```
ftpmount [lsqb ]user[lsqb ]:password[@]hostname[lsqb ]:port ][lsqb ]/root_dir mount_point [lsqb ]-own [lsqb ]-uid=id [lsqb ]-gid=id [lsqb ]-fmask=mask [lsqb ]-dmask=mask [lsqb ]-active
```

A paraméterek: [alapértékek]

- * user: A felhasználói név, amivel belépsz az FTP szerverre. [anonymous]
- * password: A felhasználói jelszó. [user@ftpf.s.sourceforge.net]
- * hostname: Az FTP szerver.
- * port: A port, amin a szerver figyel. [21]
- * root_dir: Az FTP szerver felcsatolandó könyvtára. Megadható a bevezető / jel nélkül is (vagyis a könyvtár).
- * mount_point: A helyi könyvtár, amibe az FTP könyvtárat fel akarod csatolni.
- * own: Jelzőbit minden távoli fájl tulajdonosi viszonyának jelzésére. Olyan FTP-knél hasznos, amiknél a távoli fájl tulajdonosa nem lehet a helyi felhasználó.
- * uid: Annak a helyi felhasználónak az UID-je, akié lesz a felcsatolt könyvtár.
- * gid: A helyi csoport azonosítója, aki birtokolja a felcsatolt könyvtárat.
- * fmask: A numerikus fájlmaszk, ami az összes felcsatolt fájllel VAGY kapcsolatban lesz.
- * dmask: A numerikus könyvtármask, ami az összes felcsatolt könyvtárral VAGY kapcsolatban lesz.
- * active: Jelzőbit az aktív módú FTP forgalom jelzésére. Hasznos, ha valamilyen tűzfal mögött van a szerver.

Például: `ftpmount mali@ftp.linuxnet.wox.org /mnt/ftpf -uid=500 -gid=500 -dmask=555`

Általában jó ötlet, ha nem adod meg paraméterként a jelszavadat, mivel az ftpmount úgyis kérdezni fogja.

8.2.3. A mount-féle módszer

Ha valamilyen okból nem az ftpmount-ot választod (valószínűleg valamilyen rendszermag-foltot telepítéssel és lusta vagy az ftpmount-ot is telepíteni), itt egy módszer a jó öreg mount paranccsal:

```
mount -n -t ftpfs none mount_point -o ip=server_ip [lsqb ],user=user_name [lsqb ],pass=password [lsqb ],port=server_port [lsqb ],root=root_dir [lsqb ],own [lsqb ],uid=id [lsqb ],gid=id [lsqb ],fmode=mask [lsqb ],dmode=mask [lsqb ],active
```

Figyeld meg, hogy a szerver IP címét kell megadnod és az egyetlen lehetőség a jelszó megadására, ha egyszerűen beírod a parancsorb. Például, mialatt teszteltem, a következő parancsot használtam:

```
mount -n -t ftpfs none /mnt/ftpf -o ip=127.0.0.1,user=mali,pass=my_pass
```

8.2.4. Néhány megjegyzés

A kötet leválasztására használható az

```
umount mount_point
```

parancs.

Az own opció (-o az ftpmount-nál) ráerőlteti az összes felcsatolt fájlra a felcsatolását végző felhasználó tulajdonosi jogát. Ez hasznos, ha ki akarjuk igazítani néhány szerver különös felhasználó/jogosultság beállítását (SERVU és hasonlók).

Néhány bölcs szó:

- Használd a -n felcsatolási kapcsolót! Gondolom, nem akarsz a felhasználó/jelszó információidat az mtab-ban vizsgálni.
- Ne "tép ki a falból"! (Kitépni a falból = egy tucat folyamat még olvassa a felcsatolási pont alatti dolgokat)
- Egyetlen folyamattal működik a legjobban! Amíg a konkurens hozzáférés (normál körülmények közt) nem okoz semmilyen problémát, a kimenetet egy olvasási folyamatra optimalizálták (a TCP kapcsolat életben marad). Tehát ha mozit akarsz nézni, nem akarhatod, hogy másik folyamat is hozzáférjen a felcsatolási ponthoz és agyonvágja a teljesítményt (bízz bennem!).
- Az IP címformátum nem kóser - használd az ftpmount-ot.

9. A rendszermagról szóló könyvek és dokumentumok

A Linux-rendszermagról szóló könyveket találsz a következő helyeken:

- A Linux-rendszermag működésének megértéséhez ajánlott elolvasni az /usr/src/linux/Documentation könyvtárban lévő kernel-docs.txt fájlt, amely az interneten megtalálható a "<http://www.dit.upm.es/~jmseyas/linux/kernel/hackers-docs.html>" (<http://www.dit.upm.es/~jmseyas/linux/kernel/hackers-docs.html>) honlapon is. Ez számos fontos hivatkozást tartalmaz Linux-rendszermag tankönyvekre és dokumentációkra. El kell látogatni erre a webhelyre.
- Kernel book (Rendszermag könyv) a "<http://kernelbook.sourceforge.net>" (<http://kernelbook.sourceforge.net>) és a "<http://sourceforge.net/projects/kernelbook>" (<http://sourceforge.net/projects/kernelbook>) honlapon.
- A rendszermagról szóló könyvek, mint a "The Linux Kernel Module Programming Guide", "Linux Kernel 2.4 Internals", "The Linux System Administrators Guide", "The Linux Network Administrator's Guide" és mások megtalálhatók a "<http://www.tldp.org/guides.html>" (<http://www.tldp.org/guides.html>) honlapon.
- FreeTech könyvek: "<http://www.tcfb.com/freetechbooks/booklinuxdev.html>" (<http://www.tcfb.com/freetechbooks/booklinuxdev.html>)
- Rusty-féle doksik: "<http://www.netfilter.org/unreliable-guides>" (<http://www.netfilter.org/unreliable-guides>)
- Linux-rendszermag hivatkozások: "<http://www.topology.org/soft/lkernel.html>" (<http://www.topology.org/soft/lkernel.html>)
- Linux Kernel Internals: "<http://www.moses.uklinux.net/patches/lki.html>" (<http://www.moses.uklinux.net/patches/lki.html>)
- Könyvekre mutató hivatkozások: "<http://linux-mm.org/kernel-links.shtml>" (<http://linux-mm.org/kernel-links.shtml>)

A rendszermag készítésének bemutatása: Presentation of Kernel building process NBLUG Linux Seminars - Kernel Building Presentation Talks (<http://mike.passwall.com/nblug/kernel-talk>)

Hivatkozások egyéb, a témába vágó HOGYANokra:

- Bootdisk-HOWTO (<http://www.tldp.org/HOWTO/Bootdisk-HOWTO/index.html>) (Linux indítólemez HOGYAN)
- Sound-HOWTO : (<http://www.tldp.org/HOWTO/Sound-HOWTO/index.html>) hangkártyák és segédprogramjaik
- SCSI-HOWTO: (<http://www.tldp.org/HOWTO/SCSI-Generic-HOWTO/index.html>) minden, amit tudni kell az SCSI vezérlőkről és eszközökről. Olvasd a SCSI-2.4-HOWTO (<http://www.tldp.org/HOWTO/SCSI-2.4-HOWTO/index.html>) doksiban.
- NET-2-HOWTO: (<http://www.tldp.org/HOWTO/Net-HOWTO/index.html>) hálózatkezelés
- PPP-HOWTO: (<http://www.tldp.org/HOWTO/PPP-HOWTO/index.html>) PPP hálózatkezelés általában
- PCMCIA-HOWTO: (<http://www.tldp.org/HOWTO/PCMCIA-HOWTO.html>) a notebook géped meghajtóiról
- ELF-HOWTO: (<http://www.doclib.org/Linux/docs/HOWTO/other-formats/html/ELF-HOWTO-html/ELF-HOWTO.html>) ELF: mi is ez, konvertálás... Tüköroldalak: ELF-HOWTO-mirror (http://cs.mipt.ru/docs/comp/eng/os/linux/howto/howto_english/elf/elf-howto.html) . Lásd még: GCC-HOWTO (<http://www.tldp.org/HOWTO/GCC-HOWTO>)
- Hardware-HOWTO: (<http://www.tldp.org/HOWTO/Hardware-HOWTO/index.html>) a támogatott hardverek áttekintése
- Module mini-HOWTO: (<http://tldp.org/HOWTO/mini/Modules/index.html>) még több információ a rendszermag-modulokról
- Kernel mini-HOWTO: (<http://tldp.org/HOWTO/mini/Kernel/index.html>) a kernel démonról
- BogoMips mini-HOWTO: (<http://tldp.org/HOWTO/mini/BogoMips.html>) (BogoMips mini-HOGYAN: (<http://tldp.fsf.hu/HOWTO/mini/BogoMips-hu.html>)) ha csak kíváncsi vagy.

10. Információk a rendszermag fájljairól

Ez a rész "rövid áttekintést" és "bemutatást" tartalmaz a Linux-rendszermag egyes részeiről. Ha van időd, olvasd el.

Figyelmeztetés: nagyon elővigyázatosnak kell lenned ezekkel a fájlokkal, és nem szabad szerkeszteni vagy mozgatni/törölni/átnevezni őket.

10.1. vmlinuz és vmlinux

A "vm" azt jelenti "Virtuális Memória" ("Virtual Memory"). A Linux támogatja a virtuális memória használatát, szemben az olyan régi rendszerekkel mint a DOS. Annál a 640 kByte egy komoly korlát volt. A Linux képes virtuális memóriaként használni a merevlemezt, ezért "vm" a neve. A vmlinuz a rendszermag végrehajtható fájlja. Helye a /boot/vmlinuz könyvtár. Ez lehet egy szimbolikus hivatkozás valamire, például /boot/vmlinuz-2.4.18-19.8.0. A "make zImage" parancs készíti el a vmlinuz fájlt, és a "cp /usr/src/linux/arch/i386/linux/boot/zImage /boot/vmlinuz" paranccsal rakhatod a helyére. A vmlinuz a vmlinux tömörített változata. A zImage ezért visszamenőleg kompatibilis (a kisebb rendszermagok esetében). Megjegyzendő, hogy a közeljövőben megszűnhet a zImage, és előnyben részesül a "make bzImage" (big zImage; nagy zImage). A zImage (vmlinuz) nem csak egy tömörített fájl, de van benne egy beépített gzip-kicsomagoló is (a fájl elejében). Tehát nem lehet használni a gzip -dc és gunzip parancsokat a vmlinuz kicsomagolására.

A zImage és a bzImage egyaránt tömörített a gzip programmal. A rendszermagban van egy mini-gunzip, ami a rendszermag kicsomagolására és indítására szolgál. A különbség az, hogy a régi zImage az alsó memóriába (az első 640 kByte-ra), míg a bzImage a rendszermagot a felső memóriába csomagolja ki (1 MByte fölé).

A vmlinux a tömörítetlen rendszermag-fájl, a vmlinuz a tömörített, amit betölthetővé tettek. (Figyeld meg, hogy mindkét név hasonlóan néz ki, kivéve az utolsó z betűt). Általában nem kell törődnöd a vmlinux fájljal, ez csak egy közbenső lépés.

A rendszermag általában egy bzImage fájlt készít, eltárolja az arch/i386/boot könyvtárban, és a felhasználónak kell átmásolni azt a /boot könyvtárba, majd beállítani a GRUB vagy a LILO rendszerbetöltőt.

10.2. Rendszerbetöltő (bootloader) fájlok

A .b fájlok a rendszerbetöltő fájlok. Ezek szükségesek a rendszermag memóriába való betöltéséhez. Lehetőleg NE bántsd őket.

```
ls -l /boot/*.b
-rw-r--r-- 1 root root 5824 Sep 5 2002 /boot/boot.b
-rw-r--r-- 1 root root 612 Sep 5 2002 /boot/chain.b
-rw-r--r-- 1 root root 640 Sep 5 2002 /boot/os2_d.b
```

10.3. Üzenetfájl (message file)

A "message" fájl tartalmazza a bootloader által megjeleníthető üzenetet, ami az operációs rendszer kiválasztására szólít fel. Ezért NE nyúlj hozzá.

```
ls -l /boot/message*
-rw-r--r-- 1 root root 23108 Sep 6 2002 /boot/message
-rw-r--r-- 1 root root 21282 Sep 6 2002 /boot/message.ja
```

10.4. initrd.img

Részletesen az "A" függelék - initrd.img fájl készítése (#create_initrd) fejezetben olvashatsz erről.

10.5. bzImage

A bzImage a "make bzImage" parancs által készített tömörített rendszermag-fájl, ami a fordítás során jött létre. Fontos megjegyezni, hogy a bzImage nincs tömörítve a *bzip2* programmal!! A bz a bzImage nevében félrevezető!! A valódi jelentése "Big Zimage". A "b" jelentése a bzImage szóban "big". A zImage és a bzImage egyaránt a gzip módszerrel van tömörítve. A rendszermagban van egy mini-gunzip, ami a rendszermag kicsomagolására és indítására szolgál. A különbség az, hogy a régi zImage az alsó memóriába (az első 640 kByte-ra), míg a bzImage a rendszermagot a felső memóriába csomagolja ki (1 MByte fölé). Az egyetlen ismert probléma az lehet, hogy néhány

gépen nem működik a bzImage (mert a gép bugyuta). A bzImage jelenleg gyorsabban elindul mint a zImage, de nincs különbség a rendszer *futásának* sebességében. A szabály az, ha az összes meghajtóprogram (driver) nem fér bele a zImage fájlba, akkor moduláris rendszermagra van szükség.

Ha a rendszermag kicsi a zImage és bzImage is használható, az elindított rendszer ugyanúgy fut. A nagy rendszermag mint bzImage fog futni, nem mint egy zImage. Mindkét rendszerfájl a gzip módszerrel tömörített (a bzImage nem a bzip módszerrel van tömörítve, mint azt a neve sugallja), de különböző módon töltődnek be a memóriába. A rendszermag a felső memóriaterületre is betölthető, így nem korlátozza a memóriaméret a gyagya intel architektúrán. Miért létezik két módszer? Néhány régebbi lilo és loadlin rendszerbetöltő nem kezeli a bzImage formátumot. Megjegyzendő, hogy a *betöltés* különböző, de a *futás* azonos. Sok tévinformáció származott abból, hogy mi is a bzImage fájl (a legtöbb szerint bzip2 módszerrel tömörített fájl).

10.6. module-info

A "module-info" fájl egy szimbolikus hivatkozás:

```
$ uname -r
2.4.18-19.8.0custom
```

```
# ls -l /boot/module-info*
```

```
lrwxrwxrwx    1 root    root                25 Jan 26 10:44 /boot/module-info -> module-info-2.4.18-19.8
-rw-r--r--    1 root    root           15436 Sep  4 2002 /boot/module-info-2.4.18-14
-rw-r--r--    1 root    root           15436 Jan 26 01:29 /boot/module-info-2.4.18-19.8.0
```

A fentiekhez hozzátéve megjegyzendő, hogy nem kötelező module-info szimbolikus hivatkozást készíteni egy rendszermaghoz kapcsolódó fájlhoz, mint amilyen a System-map és vmlinuz fájlhoz szükséges. Ez csak egy szöveges fájl, amely akkora, mint az aktuális module-info lista. Mielőtt eltávolítanád az összes RH rendszermaghoz szükséges "alapanyagot" a rendszeredről, készítened kellene egy mentést erről a fájlról:

```
# cp /boot/module-info-2.4.20-19.9 /boot/module-info-2.4.20-19.9.backup
```

Ezt biztonságosabb a module-info fájlal megtenni, ritkán változnak ugyanazon RH rendszermagok változatán belül.

Ez a "module-info" fájl az anaconda/utills/modlist (speciális RedHat Linux Anaconda telepítőhöz) programmal készült. Egyéb Linux összeállításokban létezhet ennek megfelelő parancs. Tájékozódj a Linux disztribútorod kézikönyv oldalairól.

Nézd meg a szkriptet és keress rá a "module-info" szóra updmodules (<http://www.mit.edu/afs/sipb/system/rhlinux/redhat-6.2/misc/src/trees/updmodules>) .

Következik egy részlet a szkriptből:

```
#!/bin/bash
# updmodules.sh
MODLIST=$PWD/./anaconda/utills/modlist
MODINFO=$KERNELROOT/boot/module-info-$version
-- kivágás
bla-bla-bla
-- kivágás
```

```
# module-info fajl elkeszítése
$MODLIST --modinfo-file $MODINFO --ignore-missing --modinfo \
$(ls *.o | sed 's/\.$//') > ../modinfo
```

Az anaconda/Utils/modlist program az anaconda-runtime*.rpm csomagban van a RedHat CD-ROM-on:

```
cd /mnt/cdrom/RedHat/RPMS
rpm -i anaconda-8.0-4.i386.rpm
rpm -i anaconda-runtime-8.0-4.i386.rpm
ls -l /usr/lib/anaconda-runtime/modlist
```

Beszerezheted a forráskódját (anaconda/Utils/modlist.c) az anaconda*.src.rpm-ből a "<http://www.rpmfind.net/linux/rpm2html/search.php?query=anaconda>" (<http://www.rpmfind.net/linux/rpm2html/search.php?query=anaconda>) webhelyről. Egyből olvashatod is: modlist.c (<http://www.trustix.net/pub/Trustix/trustix-1.5/i586/misc/src/anaconda/Utils/modlist.c>) .

A "module-info" a fordítás során készül el. Ez egy információs fájl, amit legalábbis akkor használnak, mielőtt kitöltésre kerülnek a rendszermag megfelelő Oops jelentései. Ez egy lista a modulok belépési pontjairól. Ezen kívül a depmod is használja azon táblák felépítésénél, amiket az insmod és rokonsága használ. Függőségi információkat tartalmaz azokról a modulokról, amiket egy adott modul előtt be kell tölteni stb.

A lényeg az, hogy "Ne távolítsd el a module-info fájlt."

Néhány információ a module-info fájlról:

- A rendszermag rpm fájlok tartalmazzák (az anaconda-runtime*.rpm építi fel)
- Egy hivatkozás a module-info-{kernel-version} fájlhoz
- Az összes hozzáférhető modulról tartalmaz információt (legalábbis azokról, amik benne vannak az alapértelmezett rendszermag beállításban).
- Fontos az anaconda számára - az anaconda/Utils/modlist parancsban.
- A kudzu is használhatja, hogy felderítse a modulok alapértelmezett paramétereit, amikor elkészíti a bejegyzéseket az /etc/modules.conf fájlban. Ha használaton kívül helyezed a module-info fájlt, leállítod a gépet, beraksz egy új hálózati kártyát és újraindítasz, a kudzu hangosan reklamálni fog. Nézd meg a kudzu forráskódját.

10.7. config

Minden alkalommal, ha új rendszermagot fordítasz és telepíted a rendszermag-fájlt a /boot könyvtárba, a megfelelő beállítófájlt szintén át kell másolnod a /boot könyvtárba, dokumentációs célból és későbbi hivatkozás miatt. Ezeket a fájlokat NE változtasd vagy szerkeszd!

```
ls -l /boot/config-*
-rw-r--r-- 1 root root 42111 Sep 4 2002 /boot/config-2.4.18-14
-rw-r--r-- 1 root root 42328 Jan 26 01:29 /boot/config-2.4.18-19.8.0
-rw-r--r-- 1 root root 51426 Jan 25 22:21 /boot/config-2.4.18-19.8.0BOOT
-rw-r--r-- 1 root root 52328 Jan 28 03:22 /boot/config-2.4.18-19.8.0-26mar2003
```

10.8. grub

Ha a GRUB rendszerbetöltőt használod, akkor lesz ott egy "grub" könyvtár is.

```
ls /boot/grub
device.map      ffs_stagel_5  menu.lst      reiserfs_stagel_5  stage2
e2fs_stagel_5  grub.conf    minix_stagel_5  splash.xpm.gz      vstafs_stagel_5
fat_stagel_5   jfs_stagel_5  stagel        xfs_stagel_5
```

Részletesen a "C" függelék - GRUB részletesen, grub.conf mintafájl (#grubconf) fejezetben olvashatsz erről.

10.9. System.map

A System.map egy "telefonkönyv-szerű" függvénylistája egy bizonyos lefordított rendszeragnak. Tipikusan egy szimbolikus hivatkozás az éppen futó rendszermag System.map fájljára. Ha rossz (vagy semmilyen) System.map fájlt használasz, az összeomlások nyomon követése nehezebb, de más hatása nincs. A System.map nélkül kisebb zavaró üzenetekkel kell szembesülnöd.

NE bánts a System.map fájlokat.

```
ls -ld /boot/System.map*
lrwxrwxrwx    1 root    root          30 Jan 26 19:26 /boot/System.map -> System.map-2.4.18-19.8.0
-rw-r--r--    1 root    root        501166 Sep  4 2002 /boot/System.map-2.4.18-14
-rw-r--r--    1 root    root        510786 Jan 26 01:29 /boot/System.map-2.4.18-19.8.0
-rw-r--r--    1 root    root        331213 Jan 25 22:21 /boot/System.map-2.4.18-19.8.0BOOT
-rw-r--r--    1 root    root        503246 Jan 26 19:26 /boot/System.map-2.4.18-19.8.0custom
```

Hogyan készül a rendszermag szimbólumtáblája (Kernel Symbol Table)? A System.map fájlt az "nm vmlinux" készíti el, és a nem fontos vagy érdeklődésre számot nem tartó szimbólumokat kiszedi a grep. Amikor lefordítod a rendszermagot, a System.map fájl az /usr/src/linux/System.map fájlba kerül. Valahogy így:

```
nm /boot/vmlinux-2.4.18-19.8.0 > System.map
# Ez egy sor az /usr/src/linux/Makefile fájlból
nm vmlinux | grep -v '\(compiled\)|\(\.o\$\)\|(\ [aUw] \)\|\(\.\.ng\$\)\|\(LASH[RL]DI\)' | sort > S
cp /usr/src/linux/System.map /boot/System.map-2.4.18-14 # For v2.4.18
```

Forrás: "<http://www.dirac.org/linux/systemmap.html>" (<http://www.dirac.org/linux/systemmap.html>)

10.9.1. System.map

Úgy néz ki, hogy információhiány van a System.map fájlról. Tényleg semmi rendkívüli nincs benne, és a dolgok állása szerint tényleg nem olyan fontos. De az információ hiánya homályossá teszi. Olyan, mint a fülcimpa: mindenkinek van, de senki sem tudja igazán, miért. Ez egy kis weboldal, amit a "miért" leírására hoztam össze.

Megjegyzem, nem vagyok 100%-ig korrekt. Példának okáért lehetséges, hogy egy rendszerben nincs /proc fájlrendszer támogatás, de a legtöbben van. Feltételezem, hogy "úszol az árral" és egy meglehetősen tipikus rendszered van.

A dolgok egy része az Oops-okról az Alessandro Rubini-féle "Linux Device Drivers" (Linux meghajtóprogramok) leírásból származik, amiből a legtöbbet tanultam a rendszermag programozásáról.

10.9.2. Mik a szimbólumok?

Programozási környezetben a szimbólum a program építőeleme: változónév vagy függvéynév. Nem meglepetés, hogy a rendszermagban is vannak szimbólumai, ugyanúgy, mint az általad írt programoknak. A különbség persze ott van, hogy a rendszermag nagyon bonyolult darab kódolási szempontból, és sok-sok globális szimbóluma van.

10.9.3. Mi a rendszermag szimbólumtáblája (Kernel Symbol Table)?

A rendszermag nem használ szimbólumneveket. Sokkal jobban szereti tudni a változó vagy függvény nevét azok címei által. Ahelyett, hogy a size_t BytesRead formát használná, előnyben részesíti azt, ha erre a változóra (példának okáért) c0343f20 formában hivatkozhat.

Másrésztől, az emberek nem szeretik a c0343f20 kinézetű neveket. Jobban kedveljük azt, hogy size_t BytesRead. Normál esetben ez nem jelent problémát. A rendszermagot főleg C nyelven írták, ezért a fordító/szerkesztő megengedi, hogy szimbólumneveket használjunk kódolás közben, a rendszermagban pedig engedi, hogy címeket használjon futás közben. Mindenki boldog lehet.

Azonban vannak olyan szituációk, amikor tudnunk kell egy szimbólum címét (vagy egy címhez tartozó szimbólumot). Ez a szimbólumtábla által valósul meg, és nagyon hasonló ahhoz, ahogy a gdb (GNU debugger - a ford.) visszaadja a függvényneveket egy címről (vagy egy címet a függvéynévből). A szimbólumtábla egy lista az összes szimbólumról, a címekkel együtt. Íme egy példa:

```
c03441a0 B dmi_broken
c03441a4 B is_sony_vaio_laptop
c03441c0 b dmi_ident
c0344200 b pci_bios_present
c0344204 b pirq_table
c0344208 b pirq_router
c034420c b pirq_router_dev
c0344220 b ascii_buffer
c0344224 b ascii_buf_bytes
```

Látható, hogy a dmi_broken nevű változó a c03441a0 rendszermag-címen van.

10.9.4. Mi is a System.map fájl?

Két fájl használatos szimbólumtáblaként:

1. /proc/ksyms
2. System.map

Na mármost. Már tudod, mi is a System.map fájl.

Minden alkalommal, ha új rendszermagot fordítasz, a különböző szimbólumnevek címei megváltoznak.

A /proc/ksyms egy "folyamatfájl" és a rendszermag indulásakor menet közben készül el. Valójában ez nem fájl: egyszerűen a rendszermag adatainak megjelenítése, ami azt az illúziót adja, mintha lemezn lévő fájl lenne. Ha nem hiszel nekem, próbáld megállapítani a /proc/ksyms fájl méretét. Ezért mindig az aktuálisan futó rendszermaghoz képest lesz korrekt.

A System.map azonban egy létező fájl a fájlrendszeredben. Amikor új rendszermagot fordítasz, ennek régi verziója rossz szimbólum-információkat tartalmaz. Egy új verzió készül minden egyes új fordításkor, és ki kell cserélned a régit az újjal.

10.9.5. Mi az Oops?

Mi a leggyakoribb hiba a házilag készült programjaiddal? A szegmentációs hiba (segfault). A jó öreg signal 11.

Mi a Linux-rendszerem leggyakoribb hibája? A segfault. Itt azonban a segfault fogalma sokkal összetettebb, és ahogy az várható sokkal komolyabb. Amikor a rendszermag egy hibás mutatóra hivatkozik, azt nem segfault-nak hívjuk - ezt hívják "oops"-nak. Egy ilyen oops rendszermag-hibát jelez, mindig jelteni és javítani kell.

Figyeld meg, hogy az oops nem ugyanaz a dolog, mint a segfault. A programod nem tud kijönni egy segfault-ból. A rendszermag viszont nem szükségszerűen kerül instabil állapotba, ha egy oops fordul elő. A rendszermag nagyon robusztus; az oops csak az aktuális folyamatot öli meg, a rendszermag többi részét megfelelően jó állapotban hagyhatja.

Az oops nem egyenlő a rendszermag pánikkal (kernel panic). Pánik alkalmával a rendszermag nem tud tovább futni; a rendszer halt állapotba zuhan és újra kell indítani. Egy oops akkor okozhat pánikot, ha a rendszer egy életfontosságú része semmisül meg. Egy oops valamely eszközevezlőben például majdnem sosem okoz pánikot.

Amikor egy oops előfordul, a rendszer a hibakereséshez elengedhetetlen információt nyomtat ki, mint például a CPU összes regiszterének tartalmát és az oldalleíró táblák (page descriptor tables) helyét. Főleg az EIP (utasítás mutató) tartalma íródik ki. Mint ez itt:

```
EIP: 0010:[<00000000>]  
Call Trace: [<c010b860>]
```

10.9.6. Mit köze egy oops-nak a System.map fájlhoz?

Egyetérthetsz azzal, hogy az EIP-ben adott információ és a nyomkövetési adatok nem valami információgazdagok. Ennél is fontosabb, hogy még a rendszermag fejlesztőinek sem azok. Mivel a szimbólumnak nincs fix címe, a c010b860 mutathat bárhova.

Ahhoz, hogy használhassuk ezeket a titkosított oops-kimeneteket, a Linux egy klogd nevű démon használ, a rendszermag naplózó démon. A klogd elfogja a rendszermag oops-ait és a syslog segítségével naplózza, kicserélve néhány haszontalan információt, mint a c010b860 olyanra, amit ember is tud használni. Más szóval, a klogd egy rendszermag-üzenet naplózó, ami név-cím feloldást tud végezni. Amint átalakítja a rendszermag üzeneteit, egy olyan naplózót használ, ami a rendszerszintű üzeneteket tudja naplózni, általában a syslogd démon.

A név-cím feloldáshoz a klogd a System.map fájlt használja. Most már tudod, mi az oops és mi köze a System.map fájlhoz.

Megjegyzések: Jelenleg kétféle címfeloldást végez a klogd.

- Statikus fordítást, ami használja a System.map fájlt.
- A dinamikus fordítást, amit a betölthető modulokkal használnak, nem használja

a System.map fájlt, ezért nem fontos ennek tárgyalásánál, de azért röviden ismertetem.

A klogd dinamikus fordítása

Tegyük fel, hogy betöltöttél egy rendszermag-modult, ami oops-ot idézett elő. Egy oops üzenet készült és a klogd elfogta. Azt találta, hogy az oops a d00cf810-nál fordult elő. Mivel ez a cím egy dinamikus betöltött modulhoz tartozik, nincs bejegyzés hozzá a System.map fájlban. A klogd keresi, de nem talál semmit így arra következtet, hogy egy betölthető modul generálta az üzenetet. A klogd ezután lekérdezi a rendszermagot olyan szimbólumokért, amiket a betölthető modulok exportáltak. Még ha a modul szerzője nem exportálta is a szimbólumokat, legalább a klogd tudni fogja, melyik modul idézte elő az oops-ot, ami jobb, mint semmit sem tudni az oops-ról.

Más programok is használják a System.map fájlt és rövidesen ezzel is foglalkozom.

10.9.7. Hol kellene lennie a System.map fájlnak?

A System.map bárhol lehet, ahol az őt használó szoftverek keresik. Most beszéljünk arról, hogy a klogd hol keresi. Az induláskor, ha a klogd nem kapta meg argumentumként a System.map helyét, akkor három helyen keresi a következő sorrendben:

1. /boot/System.map
2. /System.map
3. /usr/src/linux/System.map

A System.map ezenkívül verzió-információkat is tartalmaz, és a klogd intelligens módon a megfelelő map (térkép)fájlt keresi meg. Például, ha a 2.4.18-as rendszermagot futtatod és a hozzá társított fájl a /boot/System.map. Most fordítasz egy új 2.5.1-es rendszermagot az /usr/src/linux fán belül. A fordítási folyamat közben elkészül az /usr/src/linux/System.map fájl. Amikor elindítod az új rendszermagot, a klogd először megnézi a

/boot/System.map-et, megállapítja, hogy ez nem a futó rendszer~~mag~~ megfelelő térkép~~é~~fájl, ezután megnézi az /usr/src/linux/System.map-et, megállapítja, hogy ez a megfelelő, és elkezd olvasni a szimbólumokat.

Néhány megjegyzés:

- Valahol a 2.5.x szérián belül, a Linux-rendszer~~mag~~ elkezdett Linux-verziószám formában kicsomagolódni a tar archívumból, a sima linux helyett (kezeket fel: hányan vártuk már, hogy ez megtörténjen?) Nem tudom, hogy a klogd démont módosították-e már úgy, hogy az /usr/src/linux-verzió/System.map fájl keresse. TENNIVALÓ: nézd meg a klogd forrását. ?? FIXME ?? Ha valaki megver is ezért, kérlek küldj e-mailt, és tudasd, hogy módosították-e a klogd-t úgy, hogy a forráskód új nevű könyvtárában keressen. ??FIXME??
- A kézikönyv oldal nem ír le mindent. Nézd meg ezt:

```
# strace -f /sbin/klogd | grep 'System.map'
31208 open("/boot/System.map-2.4.18", O_RDONLY|O_LARGEFILE) = 2
```

Kétségtelen, hogy a klogd nemcsak a 3 keresési könyvtárban nézi meg a térkép~~é~~fájl megfelelő verzióját, de tudja azt is, hogy a "System.map" nevet követő "-kernelverző"-t nézze, mint a System.map-2.4.18. Ez a klogd egy nem dokumentált képessége.

Néhány meghajtónak szüksége van a System.map-re a szimbólumok feloldásához (mivel a rendszer~~mag~~ fejléc~~é~~fájljaihoz lettek linkelve, és nem mondjuk a glibc-éihez). Ezek nem fognak jól működni a futó rendszer~~mag~~hoz készült megfelelő System.map fájl nélkül. Ez NEM ugyanaz a dolog, mint hogy egy modul nem töltődik be a rendszer~~mag~~ verziószámának eltérése miatt. Ezt a rendszer~~mag~~-verzió jelzésével kell megoldani, nem a szimbólumtáblával, ami változik az ugyanolyan verziójú rendszer~~mag~~okon belül is!

10.9.8. Mi használja még a System.map fájlt?

Ne gondold, hogy a System.map csak a rendszer~~mag~~ oops-ok számára hasznos. Bár a rendszer~~mag~~ maga nem igazán használja, más programok, mint a lsof,

```
satan# strace lsof 2>&1 1> /dev/null | grep System
readlink("/proc/22711/fd/4", "/boot/System.map-2.4.18", 4095) = 23
```

és a ps :

```
satan# strace ps 2>&1 1> /dev/null | grep System
open("/boot/System.map-2.4.18", O_RDONLY|O_NONBLOCK|O_NOCTTY) = 6
```

valamint számos egyéb szoftver is, mint a dosemu igényli a megfelelő System.map-et.

10.9.9. Mi történik, ha nem megfelelő System.map fájlom van?

Tegyük fel, hogy több rendszermagod van ugyanazon a gépen. Minden egyes rendszermaghoz más-más System.map-re van szükség! Ha olyan rendszermagot indítasz, amihez nem tartozik System.map, rendszeresen látsz majd olyan üzeneteket, hogy: System.map does not match actual kernel (A System.map nem felel meg az aktuális rendszermagnak). Nem végzetes hiba, de bosszantó lehet mindig ezt látni, ha kiadsz egy "ps ax" parancsot. Néhány program, mint a dosemu, lehet, hogy nem működik megfelelően (bár nem tudok semmi biztosat erről). Végül, a klogd vagy a ksyoops kimenete nem lesz megbízható egy rendszermag oops esetén. Olvasd el a kézikönyv oldalakat, a "man ksyoops" és "man klogd" parancsok kiadása után.

10.9.10. Hogyan orvosoljam a fenti szituációt?

A megoldás, hogy az összes System.map fájlt a /boot-ban tárolod és átnevezed a rendszermag verziószámára. Tegyük fel hogy több rendszermagod van, mint:

- /boot/vmlinuz-2.2.14
- /boot/vmlinuz-2.2.13

Ezután csak nevezd át a térképállományokat a rendszermag verziójának megfelelően és rakd őket a /boot könyvtárba:

```
/boot/System.map-2.2.14
/boot/System.map-2.2.13
```

Mi van, ha két másolatod van ugyanabból a rendszermagból? Mint itt:

- /boot/vmlinuz-2.2.14
- /boot/vmlinuz-2.2.14.nosound

A legjobb válasz az lehet, ha az összes szoftver a következő fájlokat nézi meg:

```
/boot/System.map-2.2.14
/boot/System.map-2.2.14.nosound
```

Használhatsz szimbolikus hivatkozásokat is:

```
System.map-2.2.14
System.map-2.2.14.sound
ln -s System.map-2.2.14.sound System.map # Itt System.map -> System.map-2.2.14.sound
```

11. Linux rendszer-adminisztrációs eszközök

Két igazán jó adminisztrációs eszköz létezik Linuxra, ezek a Linuxconf és a Webmin.

Linux rendszer-adminisztrációs eszközök:

- A Linuxconf megtalálható a <http://www.solucorp.qc.ca/linuxconf> (<http://www.solucorp.qc.ca/linuxconf>) webhelyen. A Linuxconf egy különösen nagy projekt. A program félelmetesen jól használható, sok beállítási lehetőség van benne. A Linuxconf a LEGISMERTEBB rendszer-adminisztrációs eszköz Linuxra. Ez az összes disztribúcióban megtalálható, mint a Redhat, Debian, Suse stb. A Linuxconf egyenértékű a HPUX SAM, valamint az AIX SMITTY programjával. Egyaránt fut konzolon és az X-Window GUI felületén. Debian Linux esetén a letöltéshez és telepítéshez elég az "apt-get linuxconf".
- A Webmin a legjobban használható és nagyon népszerű rendszer-adminisztrációs eszköz a Linuxhoz. Elég elterjedt és díjazott, mint legjobb rendszer-adminisztrációs eszköz Solaris BSD, Linux, HPUX, AIX, SCO és más rendszerekhez. A Webmin évről-évre szervezetek, konferenciák és szerkesztők által odaítélt díjakat nyert az elmúlt 5 évben. A Webmin más rendszerek alatt is használható mint például AIX, Solaris, HPUX, IRIX, BSD, SCO Unix, OSF, Darwin, Apple Macintosh Mac OS X és Cygwin. Jelenleg a leginkább támogatott rendszerek a Solaris, Linux (különösen a Redhat) és FreeBSD. Perl nyelven írták, ez hordozható akárcsak a "C", de ellentétben azzal szkript-nyelv. A Perl "unokatestvére" a "C" nyelvnek. A legfőbb hátránya, hogy szüksége van az X-Window rendszerre, az Apache webszerver és a Perl telepítése és futtatása után használható a Webmin. Ezt a hézagot tömi be a Linuxconf, nincs szükség az X-Window vagy más program telepítésére a használatához. A Linuxconf akár egy buta karakteres terminálon is futhat az ncurses-al. A Webmin megtalálható a <http://www.webmin.com> (<http://www.webmin.com>) webhelyen. Itt letölthető a Webmin csomag.
- Google könyvtár http://directory.google.com/Top/Computers/Software/Operating_Systems/Unix/Administration/ (http://directory.google.com/Top/Computers/Software/Operating_Systems/Unix/Administration/)
- Íme a RedHat adminisztrációs eszközök listája:

```
# ls /usr/sbin/redhat-con*
/usr/sbin/redhat-config-bind           /usr/sbin/redhat-config-packages
/usr/sbin/redhat-config-bind-gui      /usr/sbin/redhat-config-printer
/usr/sbin/redhat-config-kickstart     /usr/sbin/redhat-config-printer-gui
/usr/sbin/redhat-config-network       /usr/sbin/redhat-config-printer-tui
/usr/sbin/redhat-config-network-cmd   /usr/sbin/redhat-config-proc
/usr/sbin/redhat-config-network-druid /usr/sbin/redhat-config-services
```

A program indításához -

```
# /usr/sbin/redhat-config-bind &
# /usr/sbin/redhat-config-network &
```

Majd próbálkozz más parancsokkal.

- Vizuális valamint Samba segédprogramok és egyebek megtalálhatók a <http://www.ibiblio.org/pub/Linux/system/admin/frontends/!INDEX.html> (<http://www.ibiblio.org/pub/Linux/system/admin/frontends/!INDEX.html>) és <http://www.ibiblio.org/pub/Linux/system/admin/!INDEX.html> (<http://www.ibiblio.org/pub/Linux/system/admin/!INDEX.html>) honlapokon.
- Rendszer, hálózati és adminisztrációs segédprogramok találhatók a <http://gd.tuwien.ac.at/opsys/linux/applications/text1.html#sysad> (<http://gd.tuwien.ac.at/opsys/linux/applications/text1.html#sysad>) honlapon.

- Képernyő-alapú adminisztrációs segédprogramok a Linux Refresher webhelyről:
<http://linuxrefresher.com/maintenance/admgui/guiconf.htm> (
<http://linuxrefresher.com/maintenance/admgui/guiconf.htm>)
- SuSE YAST adminisztrációs segédprogram, Redhat /usr/sbin/redhat-conf* és egyéb segédprogramok:
http://freshmeat.net/browse/253/?topic_id=253 (http://freshmeat.net/browse/253/?topic_id=253)
http://freshmeat.net/browse/201/?topic_id=201 (http://freshmeat.net/browse/201/?topic_id=201)

Linux rendszer-adminisztrációval kapcsolatos felhasználói kézikönyvek:

- <http://tldp.org/guides.html> (<http://tldp.org/guides.html>)
- <http://tldp.org/LDP/lame/LAME/linux-admin-made-easy/index.html> (
<http://tldp.org/LDP/lame/LAME/linux-admin-made-easy/index.html>)
- <http://tldp.org/LDP/nag2/index.html> (<http://tldp.org/LDP/nag2/index.html>)
- Linux rendszer-adminisztrációval kapcsolatos gyakorlatok:
<http://www.yolinux.com/TUTORIALS/LinuxTutorialSysAdmin.html> (
<http://www.yolinux.com/TUTORIALS/LinuxTutorialSysAdmin.html>)
- IBM Linux gyakorlatok (<http://www-106.ibm.com/developerworks/views/linux/tutorials.jsp>)
- <http://tariqnazir.tripod.com/unix.html> (<http://tariqnazir.tripod.com/unix.html>)
- <http://www.unixtools.com> (<http://www.unixtools.com>)

Rendszer-adminisztrációs System Administration magazin:

- <http://www.samag.com/> (<http://www.samag.com/>)

Linux rendszer-adminisztrációs tanfolyamok:

- Online India, Pune: <http://www.nixcraft.com/services/education/redhat/> (
<http://www.nixcraft.com/services/education/redhat/>)
- Online UK : <http://www.firstalt.co.uk/courses/la2.html> (<http://www.firstalt.co.uk/courses/la2.html>)
- Online UK : http://training.gbdirect.co.uk/courses/linux/running_linux_in_the_enterprise.html (
http://training.gbdirect.co.uk/courses/linux/running_linux_in_the_enterprise.html)
- Online USA : http://www.sgi.com/support/custeducation/courses/linux/sys_admin.html (
http://www.sgi.com/support/custeducation/courses/linux/sys_admin.html)
- Online USA : HOTT Training (http://www.traininghott.com/Courses/Linux-System-Admin-Hands-On-Training-Course-Class-Seminar-NIS-DNS-DHCP-LILO.htm?source=findwhat_keyword=linux-admin-group)
- Google directory: Google Linux System Admin education (
<http://www.google.com/search?q=courses+Linux+system+admin+&hl=en&lr=&ie=UTF-8&oe=UTF-8>)

12. Telepítés, frissítés fénysebességgel, az apt-get segítségével (Redhat, Debian, Suse, Mandrake, egyéb)

Automatizálható a Linux rendszer karbantartása egy olyan használható segédprogrammal, mint az apt-get. Az apt-get roppant hatékony, világszerte milliónyi linuxos gépen használják. Az apt-get RedHat és Debian Linux alapokon működik. Az RPM és Deb csomagokat támogató disztribúciókon használható. A disztribúciók legtöbbjén használható; ha mégsem, nagyon könnyen átszabhatod a forráskódját. A Mandrake-ben használt eszköznek urpmi a neve.

- Ha RedHat Linuxot használasz, töltsd le az apt-get csomagot az [Apt for Redhat Linux \(http://apt.freshrpms.net\)](http://apt.freshrpms.net) (Apt a RedHat Linuxhoz) honlapról.
- A további részletekről az [Apt for Redhat Linux \(http://freshrpms.net/apt\)](http://freshrpms.net/apt) (Apt a RedHat Linuxhoz) honlapon olvashatsz.
- Alaposan olvasd át a dokumentációt mielőtt használnád. Olvasd el a [Debian Docs \(http://www.debian.org/doc/ddp\)](http://www.debian.org/doc/ddp) és [User's Manuals \(http://www.debian.org/doc/user-manuals\)](http://www.debian.org/doc/user-manuals) (Felhasználói kézikönyvek) továbbá [APT HOWTO \(http://www.debian.org/doc/user-manuals#apt-howto\)](http://www.debian.org/doc/user-manuals#apt-howto) honlapokat és válaszd ki a megfelelő nyelvet, például [English: HTML \(http://www.debian.org/doc/manuals/apt-howto/index.en.html\)](http://www.debian.org/doc/manuals/apt-howto/index.en.html)
- [Apt+RPM howto \(http://bazar.conectiva.com.br/~godoy/apt-howto\)](http://bazar.conectiva.com.br/~godoy/apt-howto)
- [APT for RPM Based Linux Distributions \(http://apt4rpm.sourceforge.net\)](http://apt4rpm.sourceforge.net) (APT az RPM-alapú Linux terjesztésekhez)
- [List of Distributions which support APT \(http://www.debian.org/doc/manuals/apt-howto/ch-distros.en.html\)](http://www.debian.org/doc/manuals/apt-howto/ch-distros.en.html) (Az APT programot támogató terjesztések listája)
- Nézd meg a [YUM segédprogramot a YUM - Yellowdog Updater, Modified \(http://www.linux.duke.edu/projects/yum\)](http://www.linux.duke.edu/projects/yum) honlapon.

Többet szeretnél tudni az apt-get parancsról? Csak nézd meg a súgóját a következő bash shell-ben kiadott paranccsal:

```
$ apt-get -h | more
```

```
apt 0.5.5cnc6 for linux i386 compiled on Jul 19 2003 21:23:24
Usage: apt-get [options] command
    apt-get [options] install|remove pkg1 [pkg2 ...]
    apt-get [options] source pkg1 [pkg2 ...]
```

```
apt-get is a simple command line interface for downloading and
installing packages. The most frequently used commands are update
and install.
```

Commands:

```
update - Retrieve new lists of packages
upgrade - Perform an upgrade
install - Install new packages (pkg is libc6 not libc6.rpm)
remove - Remove packages
source - Download source archives
build-dep - Configure build-dependencies for source packages
```

dist-upgrade - Distribution upgrade, see apt-get(8)
clean - Erase downloaded archive files
autoclean - Erase old downloaded archive files
check - Verify that there are no broken dependencies

Options:

-h This help text.
-q Loggable output - no progress indicator
-qq No output except for errors
-d Download only - do NOT install or unpack archives
-s No-act. Perform ordering simulation
-y Assume Yes to all queries and do not prompt
-f Attempt to continue if the integrity check fails
-m Attempt to continue if archives are unlocatable
-u Show a list of upgraded packages as well
-b Build the source package after fetching it
-D When removing packages, remove dependencies as possible
-c=? Read this configuration file
-o=? Set an arbitrary configuration option, eg -o dir::cache=/tmp

See the apt-get(8), sources.list(5) and apt.conf(5) manual pages for more information and options.

This APT has Super Cow Powers.

A kézikönyv oldal a következő módon tekinthető meg:

man apt-get

In the bottom of manual page, look at "See Also" section

man apt.conf

man apt-cache

man apt-cdrom

Egy példa az apt-get használatára:

```
apt-get -f upgrade postgresql-contrib
apt-get upgrade postgresql-contrib
apt-get install postgresql-contrib
apt-get install gaim
apt-get dist-upgrade
```

12.1. Az "apt-get" forráskódja

Letöltheted és újrafordíthatod az apt-get forráskódját a saját terjesztésedhez. Az apt-get forráskódja letölthető a <http://ftp.freshrpms.net/pub/freshrpms/redhat/9/apt> (<http://ftp.freshrpms.net/pub/freshrpms/redhat/9/apt>) honlapról.

12.2. Az "rpmfind" segédprogram

Az apt-get programtól függetlenül az rpmfind is használható a RedHat-ban. Az rpmfind néhány dologban hasonlít az apt-get segédprogramra. Látogass el az rpmfind honlapjára: <http://rpmfind.net/linux/rpmfind> (<http://rpmfind.net/linux/rpmfind>). Az rpmfind egy olyan segédprogram, amely megkeresi, telepíti és önműködően frissíti a helyi gépeden lévő RPM fájlokat. Az rpmfind webhelyét megtalálod a <http://rpmfind.net> (<http://rpmfind.net>) címen.

13. Haladóknak szóló témák - A Linux rendszerindítási folyamata

Ez a rész nem igazán érdekes az átlagos "Józsi, otthoni PC-felhasználó" számára, inkább irányul azok felé, akik számítógép-tudományos háttérrel rendelkeznek.

A betöltési folyamat a következő: CPU-> VGA-> Power-On-Self-Test-> SCSI-> Boot Manager-> Lilo boot loader-> kernel-> init-> bash. A firmware és szoftver programok különböző üzeneteket adnak, amikor a számítógép és a Linux életre kel.

A Linux betöltési folyamatának részletes bemutatása:

1. Az alaplapon lévő BIOS előidézi a videokártya BIOS inicializálását
2. Az alaplapon lévő BIOS inicializálja önmagát
3. Az SCSI vezérlőn lévő BIOS inicializálja önmagát
4. Hardveres összegzés: az alaplapi BIOS ezután kiírja a következő összegzést a hardver tartalmáról. Majd futtatja a vírusellenőrző kódját, ami megváltozott indítószektorokat (boot sector) keres (ha engedélyezve van - a lektor).
5. BootManager menü : a Master Boot Record (MBR) kiolvasásra kerül az első merevlemezen, a DOS hagyományaihoz híven, a 0x00007c00 címre, és a processzor elkezd végrehajtani az ott leírt utasításokat. Ez az MBR betöltőkód beolvassa az aktív DOS partíció első szektorában lévő kódot.
6. A Lilo elindul: ha a Linuxot választottad és ha azt a LILO-val telepítetted, akkor betöltődik a 0x00007c00 címre. A Lilo kiírja a LILO üzenetet az előrehaladásáról, egyenként írva ki a betűket. Az első "L" betűt akkor írja ki, miután a Lilo átmozgatta magát egy jobb helyre, mint például a 0x00009A000 címre. Az "I" jelenik meg, mielőtt elkezdene a másodszintű betöltőkódját. A másodszintű betöltő írja ki a következő "L"-t, betölti a rendszermag részeire mutató leírókat, és végül kiírja a végső "O" betűt. A leírók a 0x00009d200 címen helyeződnek el. A rendszerindítási üzenet és a parancssor, ha úgy adták meg, kiíródik. A "tab" megnyomása a promptnál teszi lehetővé, hogy a felhasználó rendszert válasszon, és parancssori opciókat adjon át a rendszernek, a meghajtóinak és az "init" programnak. Ezen kívül környezeti változók is megadhatók ennél a pontnál.

A következő sor a /boot/message fájlból származik:

```
>
>
>
```

Press to list available boot image labels.

(Nyomj Tab-ot a kiválasztható rendszerindító fájlok (boot image) címkéinek megtekintéséhez)

A következő sor az /sbin/lilo promptjából származik:

```
boot:
```

Figyelem: ha a Lilo-t nem használjuk, akkor a rendszermag elejébe, a linux/arch/i386/boot/bootsect.S-be épített kód kiírja a "Loading"

üzenetet és folytatja.

A Lilo kiírja a következőt, amint tölti be a rendszermag kódját.

A "Linux-2.2.12" szöveget a "label=..." megadásából veszi a lilo.conf-ból.

```
Loading linux-2.2.12.....
```

7. A /linux/arch/i386/boot/setup.S fájlban található rendszermag-kód végzi el az átkapcsolást a processzor valós (DOS) módjából a védett (teljes 32 bites) üzemmódba. A Trampoline.S és Trampoline32.S nevű kódrészek segítik az átkapcsolást. A kisebb rendszermag fájlok (zImage) kitömörítődnek, és betöltődnek a 0x00010000 címre. A nagyobb fájlok (bzImage) ehelyett a 0x00100000 címre töltődnek. Ez a kód állítja be a regisztereket, kicsomagolja a tömörített rendszermagot (aminek az elején a linux/arch/i386/head.S található), kiírja a következő két sort a linux/arch/i386/boot/compressed/misc.c fájlból:

```
Uncompressing Linux... Ok.
```

```
Booting the kernel.
```

Az i386-specifikus setup.S most már bevégezte dolgát, és elugrik a 0x00010000 (vagy 0x00100000) címre, hogy elindítsa az általános Linux kódot.

- Processzor, konzol és memória inicializálása: ez futtatja a linux/arch/i386/head.S-et, ami azonnal elugrik a linux/init/main.c fájlban lévő start_kernel(void) függvényhez, ahol a megszakításokat újradefiniálják. A linux/kernel/module.c ezután betölti a konzol és a PCI busz meghajtóit. Ettől a ponttól kezdve a rendszermag üzenetei a memóriában is tárolódnak, és hozzáférhetőek a /bin/dmesg használatával. Általában később átkerülnek a /var/log/message fájlba is, végső tárolásra.
- A PCI busz inicializálása: a linux/init/main.c fájlban lévő mpci_init() függvény váltja ki a következő, a linux/arch/i386/kernel/bios32.c fájlban lévő sorok kiírását:
- Hálózat inicializálása: a linux/init/main.c fájlban lévő socket_init() függvény végzi el a hálózat inicializálását:

```
linux/net/socket.c prints:
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society NET3.039
linux/net/unix/af_unix.c prints:
NET4: Unix domain sockets 1.0 for Linux NET4.0.
linux/net/ipv4/af_inet.c prints:
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
linux/net/ipv4/ip_gre.c prints:
GRE over IPv4 tunneling driver
linux/net/core/dev.c prints:
early initialization of device gre0 is deferred
linux/net/core/rtnetlink.c prints:
Initializing RT netlink socket
```

- A Kernel Idle Thread (üresjárat) indulása: ennél a pontnál egy rendszermag-folyamat indul el, ami az init() függvényt futtatja, ami a linux/init/main.c fájlban definiált rutinok egyike. Ez az init() nem összetévesztendő az /sbin/init programmal, ami a rendszermag elindulása után kezd el futni. A linux/init/main.c fájlban lévő mkswapd_setup() függvény váltja ki a következő, a linux/mm/vmscan.c fájlból származó sorok kiírását: Starting kswapd v1.5
- Eszközmeghajtók inicializálása: a linux/arch/i386/kernel/setup.c rendszermag-rutin ezek után inicializálja az eszközöket és a fájlrendszereket (beépítve a rendszermagba??). A következő sorokat állítja elő, majd elágazik (fork) az /sbin/init-re:

- Általános párhuzamos port inicializálás: a `linux/drivers/misc/parport_pc.c` párhuzamos port inicializáló rutin írja ki a következőket:
 - Karakteres eszközök inicializálása: a következő 3 sor a `linux/drivers/char/serial.c` fájlból származik:
 - Block Device Initializations : `linux/drivers/block/rd.c` prints: RAM disk driver initialized: 16 RAM disks of 8192K size `linux/drivers/block/loop.c` prints: loop: registered device at major 7 `linux/drivers/block/floppy.c` prints: Floppy drive(s): fd0 is 1.44M, fd1 is 1.44M FDC 0 is a post-1991 82077
 - SCSI busz inicializálása: a következő sorok a `linux/drivers/scsi` alkönyvtárban lévő `aic7xxx.c`, `scsi.c`, `sg.c`, `sd.c` vagy `sr.c` fájljokból származnak:
 - A rendszerem Point-To-Point protokoll támogatásának inicializálása: a következő inicializálást a `linux/drivers/net/ppp.c` végzi.
 - A merevlemez-elrendezés vizsgálata: a következő sorok a `linux/drivers/block/genhd.c` fájlból származnak:
8. Init Program (Process 1) Startup : The program `/sbin/init` is started by the "idle" process (Process 0) code in `linux/init/main.c` and becomes process 1. `/sbin/init` then completes the initialization by running scripts and forking additional processes as specified in `/etc/inittab`. It starts by printing: INIT: version 2.76 booting and reads `/etc/inittab`.
9. A Bash parancsértelmező indulása: a bash shell, a `/bin/bash` indul el ezek után. A feléledése az `/etc/profile` szkript végrehajtásával kezdődik, ami beállítja a rendszerszintű környezeti változókat:

13.1. Hivatkozások a betöltési folyamat témájában

Nézd meg a következő forrásokat:

- The Linux Boot Process (<http://www.tldp.org/HOWTO/Bootdisk-HOWTO/x1440.html>) (A Linux betöltési folyamata)
- Bootdisks and Boot Process (<http://www.tldp.org/HOWTO/Bootdisk-HOWTO/x88.html>) (Indítólemezek és a betöltési folyamat)
- Linux Boot Process - by San Gabreil LUG (<http://ourworld.compuserve.com/homepages/KanjiFlash/SGVLUG.htm>) (Linux betöltési folyamat - írta San Gabreil LUG)
- Boot Process (Netmag) (<http://www.linuxnetmag.com/en/issue4/m4boot1.html>) (Betöltési folyamat (Netmag))
- Boot Process (LUG Victoria) (<http://oldfield.wattle.id.au/luv/boot.html>) (Betöltési folyamat (LUG Victoria))

14. Eme dokumentum más formátumai

Ezt a fejezetet Al Dev ([mailto:alavoor\[AT\]yahoo.com](mailto:alavoor[AT]yahoo.com)) írta, (a "<http://milkyway.has.it>" (<http://milkyway.has.it>) és a "<http://www.milkywaygalaxy.freesevers.com>" (<http://www.milkywaygalaxy.freesevers.com>) webhelyen, tükrözései megtalálhatók a [angelfire](http://www.angelfire.com/country/aldev0) (<http://www.angelfire.com/country/aldev0>) , [geocities](http://www.geocities.com/alavoor/index.html) (<http://www.geocities.com/alavoor/index.html>) , [virtualave](http://aldev0.virtualave.net) (<http://aldev0.virtualave.net>) , [Fortunecity](http://www.fortunecity.com)

(<http://members.fortunecity.com/aldev>) , Freewebsites (<http://aldev.freewebsites.com>) , Tripod (<http://members.tripod.lycos.com/aldev>) , 101xs (<http://www.101xs.com/101xs/aldev>) és 50megs (<http://aldev0.50megs.com>) webhelyeken)

Ezt a dokumentumot 14 különböző formátumban terjesztik, ezek a következők: DVI, Postscript, Latex, Adobe Acrobat PDF, LyX, GNU-info, HTML, RTF (Rich Text Format), egyszerű szöveg, Unix man oldal, egyoldalas HTML fájl, SGML (Linuxdoc formátum), SGML (Docbook formátum), MS WinHelp formátum.

A dokumentum megtalálható a

- "<http://www.tldp.org>" (<http://www.tldp.org>) webhelyen, itt kattints a "HOWTO" hivatkozásra és keresd a CTRL-f vagy ALT-f segítségével a dokumentum nevét a honlapon.

A következő tüköroldalakon is megtalálhatod ezt a dokumentumot:

- "<http://www.caldera.com/LDP/HOWTO>" (<http://www.caldera.com/LDP/HOWTO>)
- "<http://www.linux.ucla.edu/LDP>" (<http://www.linux.ucla.edu/LDP>)
- "<http://www.cc.gatech.edu/linux/LDP>" (<http://www.cc.gatech.edu/linux/LDP>)
- "<http://www.redhat.com/mirrors/LDP>" (<http://www.redhat.com/mirrors/LDP>)
- Egyéb, hozzád közeli tüköroldalak (hálózati címtől függ) találhatóak a "<http://www.tldp.org/mirrors.html>" (<http://www.tldp.org/mirrors.html>) honlapon. Válassz egy webhelyet és lépj be az /LDP/HOWTO/xxxxx-HOWTO.html könyvtárba.

- A dokumentumot egyszerű tar csomagként is letöltheted HTML, DVI, Postscript vagy SGML formátumban a "<ftp://www.tldp.org/pub/Linux/docs/HOWTO/other-formats/>" (<ftp://www.tldp.org/pub/Linux/docs/HOWTO/other-formats/>) és "<http://www.tldp.org/docs.html#howto>" (<http://www.tldp.org/docs.html#howto>) honlapokról.
- A sima szöveges formátum megtalálható az "<ftp://www.tldp.org/pub/Linux/docs/HOWTO/>" (<ftp://www.tldp.org/pub/Linux/docs/HOWTO/>) tárhelyen illetve a "<http://www.tldp.org/docs.html#howto>" (<http://www.tldp.org/docs.html#howto>) honlapon.
- Egyoldalas HTML fájlként letöltheted a "<http://www.tldp.org/docs.html#howto>" (<http://www.tldp.org/docs.html#howto>) honlapról. Az egyoldalas HTML fájlt a következő paranccsal állítható elő (olvasd az `sgml2html` kézikönyvoldalát): `sgml2html -split 0 xxxxhowto.sgml`
- Más nyelvi (mint például francia, német, spanyol, kínai, japán) fordítások találhatóak a "<ftp://www.tldp.org/pub/Linux/docs/HOWTO/>" (<ftp://www.tldp.org/pub/Linux/docs/HOWTO/>) és "<http://www.tldp.org/docs.html#howto>" (<http://www.tldp.org/docs.html#howto>) honlapokon. Jól jön bármely segítség töletek a más nyelvre való fordításhoz (íme :) - a ford.)

(Ez a rész mára elavult - a lektor) A dokumentum az "SGML-Tools" nevű eszköz segítségével készült, amely letölthető a "<http://www.sgmltools.org>" (<http://www.sgmltools.org>) webhelyről. A forrás lefordítása után a következő parancsot használhatod a konvertáláshoz:

- `sgml2html xxxxhowto.sgml` (többoldalas HTML fájlt készít)

- `sgml2html -split 0 xxxxhowto.sgml` (egyoldalas HTML fájlt készít)
- `sgml2rtf xxxxhowto.sgml` (RTF fájlt készít)
- `sgml2latex xxxxhowto.sgml` (LaTeX fájlt készít)

14.1. Acrobat PDF formátum

PDF fájlt a postscript fájl felhasználásával lehet készíteni, vagy az `acrobat distill` illetve `Ghostscript` segítségével. A postscript fájlt DVI-ből lehet elkészíteni, ami viszont közvetlenül LaTeX fájlból készül. A `distill` szoftver letölthető a "<http://www.adobe.com>" (<http://www.adobe.com>) webhelyről. Alább egy egyszerű példa látható:

```
bash$ man sgml2latex
bash$ sgml2latex filename.sgml
bash$ man dvips
bash$ dvips -o filename.ps filename.dvi
bash$ distill filename.ps
bash$ man ghostscript
bash$ man ps2pdf
bash$ ps2pdf input.ps output.pdf
bash$ acroread output.pdf &
```

Esetleg használhatod a következő Ghostscript parancsot: `ps2pdf`. A `ps2pdf` hasonlóan működik mint az Adobe Acrobat Distiller program, és majdnem minden funkcióját tudja: konvertálja a PostScript fájlokat Portable Document Format (PDF) formátumba. A `ps2pdf` konvertálást egy nagyon kicsi parancsszkriptben (batch fájlban) valósították meg, ami meghívja a GhostScript programot, kiválasztva egy speciális "kimeneti eszközt", amely a `pdfwrite`. A `ps2pdf` használatához a `pdfwrite` eszközt be kell szúrni a Makefile-ba, a GhostScript programfordításakor; olvasd a fordítási dokumentációt további részletekért.

14.2. Konvertálás Linuxdoc-ról Docbook formátumra

(Ez a fejezet mára elavult, bár bizonyos részei használhatók - a lektor) Ez a dokumentum Linuxdoc SGML formátumban készült. A Docbook SGML formátum helyettesíti a Linuxdoc formátumot, mivel sokkal több szolgáltatása van, mint annak. A Linuxdoc formátum nagyon egyszerű és könnyű használni. A formátumának Docbook SGML-re való konvertálásához használd a `ld2db.sh` szkriptet, valamint néhány perl szkriptet. Az `ld2db` kimenete nem 100%-ig tiszta, ezért használd a `clean_ld2db.pl` perl szkriptet. Lehet, hogy kézzel is kell javítani néhány sort a doksiban.

- Töltsd le a `ld2db` programot a "<http://www.dcs.gla.ac.uk/~rrt/docbook.html>" (<http://www.dcs.gla.ac.uk/~rrt/docbook.html>) vagy a "<http://milkyway.has.it>" (<http://milkyway.has.it>), Milkyway Galaxy site (<http://www.milkywaygalaxy.freesevers.com>) webhelyről.
- A `cleanup_ld2db.pl` perl szkriptet letöltheted a "<http://milkyway.has.it>" (<http://milkyway.has.it>) és Milkyway Galaxy site (<http://www.milkywaygalaxy.freesevers.com>) webhelyekről.

Az `ld2db.sh` nem 100%-ig korrekt, sok hibát fogsz kapni, mikor kiadod a

```
bash$ ld2db.sh file-linuxdoc.sgml db.sgml
bash$ cleanup.pl db.sgml > db_clean.sgml
```

```
bash$ gvim db_clean.sgml
bash$ docbook2html db.sgml
```

parancsokat. Valószínűleg kézzel kell szerkesztened néhány kisebb hibát, miután futtattad a szkriptet. Például a záró `</Para>` tag-eket neked kell kitenni minden `< Listitem>` után.

14.3. Konvertálás MS WinHelp formátumra

Átalakíthatod a Microsoft Windows Help formátumára is az SGML dokumentumot, először konvertáld át HTML formátumba a következő parancsok használatával:

```
bash$ sgml2html xxxxhowto.sgml      (ez HTML fájlt készít)
bash$ sgml2html -split 0 xxxxhowto.sgml (to generate a single page html file)
```

Majd használd a `HtmlToHlp` (<http://javadocs.planetmirror.com/htmltohlpe.html>) segédprogramot. Ezen kívül használható az `sgml2rtf` is, majd az RTF fájlokat lehet a WinHelp legyártásához felhasználni.

14.4. A különböző formátumok olvasása

A dokumentum dvi formátumban történő olvasásához használd az `xdvi` programot. Az `xdvi` a `tetex-xdvi*.rpm` csomagban van a RedHat Linuxban, amit meg lehet találni a Control Panel | Applications | Publishing | TeX menüpontokon keresztül. Az olvasáshoz add ki a

```
xdvi -geometry 80x90 howto.dvi man xdvi
```

parancsot. Méretezd át az ablakot egérrel. A navigáláshoz használd a nyílbillentyűket, a Page Up/Down gombokat, de az "f", "d", "u", "c", "l", "r", "p", "n" billentyűket is a fel/le mozgáshoz, középre igazításhoz, következő/előző oldal kéréséhez stb. A haladó menü kikapcsolásához nyomj "x"-et.

A postscript fájlt a "gv" (ghostview) vagy a "ghostscript" programmal nézheted meg. A ghostscript program a `ghostscript*.rpm` csomagban van, a gv pedig a `gv*.rpm`-ben a RedHat Linux alatt, ami elérhető a ControlPanel | Applications | Graphics menüpontokon keresztül. A gv program sokkal felhasználóbarátabb, mint a ghostscript. Ezenkívül a ghostscript és a gv is elérhető más platformokon, mint OS/2, Windows 95 és NT, még ezeken is megnézheted ezt a doksit.

- A Windows 95, OS/2 és egyéb operációs rendszerekhez beszerezhető a "<http://www.cs.wisc.edu/~ghost>" (<http://www.cs.wisc.edu/~ghost>) webhelyről.

A postscript dokumentum olvasásához add ki a

```
gv howto.ps ghostscript howto.ps
```

parancsot.

A HTML formátumú doksit olvashatod a Netscape Navigator, Microsoft Internet explorer, Redhat Baron és vagy 10 másik böngészőprogram segítségével.

A latex, LyX kimenet olvasásához használd a LyX X-Window előtétprogramot a latex programmal.

15. "A" függelék - initrd.img fájl készítése

Az *initrd* az "initial ramdisk" (kezdeti RAM-lemez) rövidítése. Egy RAM-lemezen elegendő fájl fér el a szükséges eszközmeghajtók tárolásához. Ezek a meghajtók kellene ahhoz, hogy a rendszermag fel tudja csatolni a / könyvtárat, és el tudja indítani az init-et. Az *initrd*-t tipikusan a hardver átmeneti "beindítására" használjuk, ezután a valódi rendszermag (*vmlinuz*) folytatja a betöltési folyamatot. Például, ha a rendszermag nem tudja olvasni az scsi merevlemezt az scsi eszközmeghajtó betöltése előtt. (Megoldás: tölts be egy *initrd* rendszermagot, amely majd betölti az igazi rendszermagot és használd az *initrd*-t az scsi betöltési problémájának megoldására.)

Megszabadulhatsz az "*initrd.img*" fájltól és nem szükséges akkor, ha az SCSI eszközvezérlőket fixen a rendszermagba fordítod, nem pedig modulba rakod. (Sokan ajánlják ezt).

15.1. Az mkinitrd használata

Az *mkinitrd* segédprogram készíti el az *initrd* fájlt, egyetlen paranccsal. Ez a parancs a RedHat jellegzetessége. A Linux többi terjesztésénél hasonló parancsok létezhetnek. Nagyon kényelmes program.

Olvasd el az *mkinitrd* kézikönyv oldalát.

```
/sbin/mkinitrd --help # vagy egyszerűen: "mkinitrd --help"
usage: mkinitrd [--version] [-v] [-f] [--preload <module>]
      [--omit-scsi-modules] [--omit-raid-modules] [--omit-lvm-modules]
      [--with=<module>] [--image-version] [--fstab=<fstab>] [--nocompress]
      [--builtin=<module>] [--nopivot] <initrd-image> <kernel-version>
      (example: mkinitrd /boot/initrd-2.2.5-15.img 2.2.5-15)
# olvasd el a kezikönyv oldalt .....
man mkinitrd
su - root
# Az alábbi parancs elkészíti az initrd fájlt
mkinitrd ./initrd-2.4.18-19.8.0custom.img 2.4.18-19.8.0custom
ls -l initrd-2.4.18-19.8.0custom.img
-rw-r--r-- 1 root root 127314 Mar 19 21:54 initrd-2.4.18-19.8.0custom.img
cp ./initrd-2.4.18-19.8.0custom.img /boot
```

A következő fejezetekben olvashatsz arról, hogyan kell kézzel elkészíteni az *initrd* fájlt.

15.2. Rendszermag-dokumentációk

A */boot/initrd.img* elkészítéséről szól az */usr/src/linux/Documentation/initrd.txt* leírás, valamint a Loopback-Root-mini-HOWTO (<http://www.tldp.org/HOWTO/mini/Loopback-Root-FS-3.html#ss3.3>).

15.3. Linuxman Book

Egy részlet a "<http://www.linuxman.com.cy/rute/node1.html>" (<http://www.linuxman.com.cy/rute/node1.html>) 31.7. fejezetéből.

Az SCSI telepítésének buktatói és az initrd

A következő leírás némelyike nehezen lesz érthető anélkül, hogy tudnál valamit a rendszermag-modulokról, amiket a 42. fejezetben mutat be. Később térj vissza erre a fejezetre.

Képzeld el egy rendszert, amiben egyetlen IDE lemez sincs, csak egy SCSI lemez tartalmazza a telepített Linuxot. Vannak az SCSI lemezt olvasó BIOS megszakítások, ugyanúgy, mint az IDE-hez, tehát a LILO vígan hozzá tud férni egy SCSI partíción lévő rendszermaghoz. Azonban a rendszermag "el fog veszni" a rendszermag-modulok nélkül [lásd 42. fejezet. A rendszermag magától nem tudja az összes létező hardver elemet támogatni. Ez általában egy fő részre (a rendszermag fájlra, amit ebben a fejezetben tárgyalunk) és modulok százaira (betölthető részek, amik a /lib/modules alatt található) tagolható szét [amik támogatják a sokféle típusú SCSI, hálózati, hangeszközt stb.], amely képes vezérelni azt a bizonyos SCSI meghajtót. Így bár a rendszermag be tud tölteni és el tud indulni, de nem tudja felcsatolni a gyökér fájlrendszert az SCSI modul előzetes betöltése nélkül. Viszont a modul maga a gyökér fájlrendszeren foglal helyet a /lib/modules-ban. Ez egy trükkös helyzet, és kétféle módon lehet megoldani: (a) vagy előre engedélyezett (preenabled) SCSI támogatású rendszermagot, vagy (b) egy initrd előzetes gyökér fájlrendszernek nevezett fájlt használunk.

Az első módszert ajánlom. Magától értetődő (bár időigényes) folyamat egy olyan rendszermag elkészítése, ami beépített támogatást nyújt az SCSI kártyához (és nem egy külön modulban). A beépített SCSI és hálózati meghajtók legtöbbször automatikus detektálást is végeznek, lehetővé téve az eszközhöz való időközbeni hozzáférést - mindenféle kapcsoló megadása nélkül működnék. [lsqb] A 42. fejezet tárgyalja.] és ami még fontosabb, anélkül, hogy olvasnod kéne a beállításokról. Ennek befördített hardvertámogatás a neve (a modulok támogatással szemben). Az így készült rendszermag a modul méretével lesz nagyobb. A 42. fejezet taglalja az ilyen fajta programfordítást.

A második módszer gyorsabb, de trükkösebb. A Linux támogat egy olyan dolgot, amit initrd (initial RAM disk) néven ismernek. Ez egy kicsi, +1.5 MB méretű fájlrendszer, amit a LILO tölt be, és a rendszermag a valódi gyökér fájlrendszer helyett csatol fel. A rendszermag RAM-lemezként csatolja fel ezt a fájlrendszert, végrehajtja a /linuxrc fájlt, majd felcsatolja a valódi fájlrendszert.

31.6 fejezet initrd fájl készítése

Kezdjük egy kis fájlrendszer elkészítésével. Készíts egy könyvtárat [nbsp]/initrd néven és másold át a következő fájlokat oda:

```
drwxr-xr-x    7 root    root          1024 Sep 14 20:12 initrd/
drwxr-xr-x    2 root    root          1024 Sep 14 20:12 initrd/bin/
-rwxr-xr-x    1 root    root        436328 Sep 14 20:12 initrd/bin/insmod
-rwxr-xr-x    1 root    root        424680 Sep 14 20:12 initrd/bin/sash
drwxr-xr-x    2 root    root          1024 Sep 14 20:12 initrd/dev/
crw-r--r--    1 root    root           5,   1 Sep 14 20:12 initrd/dev/console
crw-r--r--    1 root    root           1,   3 Sep 14 20:12 initrd/dev/null
brw-r--r--    1 root    root           1,   1 Sep 14 20:12 initrd/dev/ram
crw-r--r--    1 root    root           4,   0 Sep 14 20:12 initrd/dev/systty
crw-r--r--    1 root    root           4,   1 Sep 14 20:12 initrd/dev/tty1
crw-r--r--    1 root    root           4,   1 Sep 14 20:12 initrd/dev/tty2
crw-r--r--    1 root    root           4,   1 Sep 14 20:12 initrd/dev/tty3
crw-r--r--    1 root    root           4,   1 Sep 14 20:12 initrd/dev/tty4
drwxr-xr-x    2 root    root          1024 Sep 14 20:12 initrd/etc/
drwxr-xr-x    2 root    root          1024 Sep 14 20:12 initrd/lib/
```

```
-rwxr-xr-x    1 root    root          76 Sep 14 20:12 initrd/linuxrc
drwxr-xr-x    2 root    root        1024 Sep 14 20:12 initrd/loopfs/
```

Az én rendszeremen, a `initrd/bin/insmod` fájl statikusan linkelt [ami azt jelenti, hogy nem igényel megosztott könyvtárakat], a `/sbin/insmod.static` fájlból - ez a `modutils-2.3.13` csomag része. Az `initrd/bin/sash` egy statikusan linkelt parancsértelmező a `sash-3.4` csomagból. Az `insmod`-ot újra is fordíthatod forrásból, ha nincs statikus verziód. Esetleg másold be a szükséges DLL-eket a `/lib`-ből az `initrd/lib` könyvtárba. (A szükséges DLL-ek listáját megkapod, ha az `"ldd /sbin/insmod"` parancsot kiadod. Ne felejtse el átmásolni a szimbolikus linkeket sem, valamint futtatni a `"strip -s {lib}"` parancsot, a DLL-ek méretének csökkentéséhez.

Most másold át az `initrd/lib/` könyvtárba a szükséges SCSI modulokat. Például, ha egy Adaptec AIC-7850 SCSI adaptered van, akkor az `aic7xxx.o` modul kell a `/lib/modules/{version}/scsi/aic7xxx.o` fájllokból. Ezek után rakd be ezt a `initrd/lib/` könyvtárba.

```
-rw-r--r--    1 root    root        129448 Sep 27 1999 initrd/lib/aic7xxx.o
```

Az `initrd/linuxrc` fájlban egy szkriptet kell tartalmaznia, hogy betöltse az összes modult a rendszerem számára, hogy hozzáférhessen az SCSI partícióhoz. Ebben az esetben, csak az `aic7xxx` modul kell [az `insmod` az eszköz IRQ-ira és IO-címeire vonatkozó paramétereket is megkaphat, lásd a 42. fejezetet.]:

```
#!/bin/sash

aliasall

echo "Loading aic7xxx module"
insmod /lib/aic7xxx.o
```

Alaposan ellenőrizd az összes jogosultságot, majd futtasd a `chroot` parancsot a fájlrendszer teszteléséhez.

```
chroot ~/initrd /bin/sash
/linuxrc
```

Ezután készíts egy "fájlrendszer-fájlt" (file system image), hasonlóan a 19.9 fejezetben leírtakhoz:

```
dd if=/dev/zero of=~/file-inird count=2500 bs=1024
losetup /dev/loop0 ~/file-inird
mke2fs /dev/loop0
mkdir ~/mnt
mount /dev/loop0 ~/mnt
cp -a initrd/* ~/mnt/
umount ~/mnt
losetup -d /dev/loop0
```

Végül tömörítsd össze a fájlrendszert a gzip programmal, valamilyen néven:

```
gzip -c ~/file-inird > initrd-<kernel-version>
```

31.7 fejezet A lilo.conf módosítása az initrd használatához

A lilo.conf fájlt annak megfelelően kell megváltoztatni, hogy az initrd fájlrendszert töltsse be. Egyszerűen add hozzá az initrd kapcsolót. Például:

```
boot=/dev/sda
prompt
timeout = 50
compact
vga = extended
linear
image = /boot/vmlinuz-2.2.17
        initrd = /boot/initrd-2.2.17
        label = linux
        root = /dev/sda1
        read-only
```

Figyeld meg a "linear" kapcsoló használatát. Ez egy BIOS trükk, amiről a lilo(5) kézikönyv oldalán olvashatsz. Ez gyakran szükséges, viszont azt eredményezheti, hogy az SCSI lemezek nem lesznek átvihetők különböző BIOS-al rendelkező gépekre (azt jelenti, hogy újra le kell futtatni a lilo-t, ha egy másik gépbe rakod át a lemezt).

16. "B" függelék - lilo.conf mintafájl

16.1. LILO forrásanyagok

A következő dokumentumokban olvashatsz a témáról:

- <http://www.tldp.org/HOWTO/mini/LILO.html> (<http://www.tldp.org/HOWTO/mini/LILO.html>)
- Olvasd a rendszeren lévő LILO dokumentációt:

```
# Használd a kghostview, ghostview vagy gv parancsot
kghostview /usr/share/doc/lilo-21.4.4/doc/user.ps
```

```
# A HTML formátum olvasásához tedd ezt:
mkdir $HOME/lilodocs
cd $HOME/lilodocs
cp /usr/share/doc/lilo-21.4.4/doc/user.tex .
```

```
latex2html user

# Ez létrehozza a html fájlokat az usr könyvtárban
```

- 17 GRUB beállítófájl
- "<http://www.tldp.org/HOWTO/LILO-crash-rescue-HOWTO.html>" (<http://www.tldp.org/HOWTO/LILO-crash-rescue-HOWTO.html>).

16.2. LILO hibaelhárítás

A csipogó (beeper) hibakód-táblázata:

Táblázat 1. Csipogó hibakód-táblázat

Kód	Leírás
0	PC-hangszóró hiba
1	DRAM frissítési hiba
2	Paritáshiba (Paritykring defect)
3	Hiba az alap 64K RAM-ban
4	Rendszeridőzítő hiba
5	Processzor hiba
6	Billentyűzet-vezérlő hiba
7	Virtuális mód hiba
8	A videomemória tesztje nem sikerült
9	hibás ROM-BIOS ellenőrző összeg

2 rövid csipogás: a POST nem megfelelő. Hiba a hardver tesztben. 1 rövid és 2 hosszú csipogás: video hiba. 1) Video ROM BIOS, paritás hiba. 2) Probléma a videokártya vízszintes visszatérésével. 1 hosszú és 3 rövid csipogás: video hiba. 1) videokártya hiba. 2) a monitor hibás detektálása. 3) Video RAM hiba. 1 hosszú csipogás: a POST rendben lefutott. Ha a POST-ban hiba lép fel, akkor hardverprobléma van. Ellenőrizd a bővítőkártyák érintkezését

A témáról olvashatsz a <http://www.preggers.easynet.be/lilo.html> (<http://www.preggers.easynet.be/lilo.html>) honlapon.

Ha a LILO-val vannak problémák, olvasd el a következő tippeket. Ha a betöltés közben a "L01010101010101" hibával találkozol, akkor a következőket tedd:

- A kedvenc szövegszerkesztőddel nyisd meg a /etc/lilo.conf fájlt:

```
# keresd meg a következő sort:
linear
# tegyél elé megjegyzés jelet (kettős keresztet - a lektor), hogy így nézzen ki:
# linear
Mentsd el és futtasd a lilo-t.
```

- A boot-partíciónak a 8GB-os határon belül kell lennie. Ha problémád van, olyan programokkal, mint a Partition Magic vagy a Mandrake-féle DiskDrake, könnyen orvosolhatod.
- A 01 a DRAM frissítési hiba. Ha csak szimplán "L 01"-et látsz, indíts újra a gépet a CTRL+ATL+DEL kombinációval (viszont ne kapcsold ki a gépet, csak CTRL+ATL+DEL-t nyomj). Ez javíthatja a problémát.

16.3. Egy LILO beállítófájl minta

Mindig adj egy dátumra vonatkozó kiterjesztést a fájlnevhez, mert ez jelzi, mikor készítetted a rendszermagot, amint alább is látható:

```
bash# man lilo
bash# man lilo.conf
Szerkeszd az /etc/lilo.conf fájlt és rakd be ezeket a sorokat:
image=/boot/bzImage.myker.26mar2001
label=myker
root=/dev/hda1
read-only
Az eszköznevet a "root=" számára a következő paranccsal ellenőrizheted:
bash# df /
Most add ki a következő parancsokat:
bash# lilo
bash# lilo -q
```

Akkor is újra kell futtatnod a lilo-t, ha a "myker" bejegyzés létezik, valahányszor új bzImage fájlt készítesz.

Alább egy példa /etc/lilo.conf látható. Az elnevezési szabályokat is követheted, mint ker2217 (a 2.2.17-es rendszermaghoz), ker2214 (a 2.2.14-hez). Sok rendszermag fájlod lehet ugyanabban a /boot rendszerben. Az én gépemem valami hasonló van:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=firewall
image=/boot/vmlinuz-2.2.14-5.0
label=ker2214
read-only
root=/dev/hda9
image=/boot/vmlinuz-2.2.17-14
label=ker2217
read-only
root=/dev/hda9
#image=/usr/src/linux/arch/i386/boot/bzImage
# label=myker
# root=/dev/hda7
# read-only
```

```
image=/boot/bzImage.myker.11feb2001
label=myker11feb
root=/dev/hda9
read-only
image=/boot/bzImage.myker.01jan2001
label=myker01jan
root=/dev/hda9
read-only
image=/boot/bzImage.myker-firewall.16mar2001
label=firewall
root=/dev/hda9
read-only
```

17. "C" függelék - GRUB részletesen, grub.conf mintafájl

17.1. GRUB forrásanyagok

Olvasd a

- ["http://www.tldp.org/HOWTO/Linux+Win9x+Grub-HOWTO/intro.html"](http://www.tldp.org/HOWTO/Linux+Win9x+Grub-HOWTO/intro.html)
(<http://www.tldp.org/HOWTO/Linux+Win9x+Grub-HOWTO/intro.html>)
(["http://tldp.fsf.hu/HOWTO/Linux+Win9x+Grub-HOWTO-hu.html"](http://tldp.fsf.hu/HOWTO/Linux+Win9x+Grub-HOWTO-hu.html)
(<http://tldp.fsf.hu/HOWTO/Linux+Win9x+Grub-HOWTO-hu.html>)) dokumentumot.
- GNU GRUB ["http://www.gnu.org/software/grub"](http://www.gnu.org/software/grub) (<http://www.gnu.org/software/grub>)
- Redhat Manual (<http://www.redhat.com/docs/manuals/linux/RHL-7.2-Manual/ref-guide/ch-grub.html>) (RedHat kézikönyv).
- Multiboot-with-GRUB minihowto (<http://www.tldp.org/HOWTO/mini/Multiboot-with-GRUB.html>)
- Grub Manual (http://www.mcc.ac.uk/grub/grub_toc.html) (Grub kézikönyv)

```
bash# man grub
bash# man grubby # (parancssoros eszköz a grub, lilo, és elilo beállításához)
bash# man grub-install
```

Szerkeszd az `/etc/grub.conf` fájlt, az új rendszermagok bejegyzéseinek létrehozásához. Lásd az alábbi mintafájlt:

17.2. GRUB tippek

A RedHat Linuxban a grub képernyőjénél állva nyomj egy `c`-t a parancssori kapcsolók megtekintéséhez:

```
A Linux betöltéséhez tedd ezt:
grub> help
```

```
grub> root
      (hd1,1): Filesystem is type ext2fs, partition type 0x83
grub> root (hd1,0)

grub> kernel / <Nyomj-TAB-ot>
Kilistázza az összes fájlt.

grub> kernel /boot <Nyomj-TAB-ot>
Kilistázza a /boot könyvtár tartalmát.

grub> kernel /boot/vmlinuz

grub> boot
```

Erről a GRUB Manual (http://www.gnu.org/manual/grub/html_mono/grub.html) (GRUB kézikönyv) dokumentációban is olvashatsz. A MS Windows 95/2000 stb. betöltéséhez tedd ezt: ha nem támogatott operációs rendszert (például Windows 95) akarsz betölteni, csinálj betöltési láncot (chain-load) az operációs rendszer betöltőjéhez. Normál esetben a rendszerbetöltőt annak a partíciónak a boot szektora tartalmazza, amelyikre az operációs rendszert telepítettük.

```
grub> help
grub> help rootnoverify
grub> rootnoverify (hd0,0)
grub> makeactive
grub> chainloader +1
grub> boot
```

17.3. Egy GRUB beállítófájl minta

```
# grub.conf generated by anaconda
#
# Figyeld meg, hogy nem kell ujrafuttatni a grub-ot, ha változtatsz a fajlon
# FIGYELEM: nincs /boot partíciód. Ez azt jelenti, hogy minden rendszermag es
# initrd utvonal a / könyvtárhoz viszonyított, például:
#       root (hd0,8)
#       kernel /boot/vmlinuz-version ro root=/dev/hda9
#       initrd /boot/initrd-version.img
#boot=/dev/hda
# Alapertelmezesben a masodik bejegyzes inditasa.
default=1
# Az else bejegyzes a biztonsagi tartalek.
fallback 0
# Automatikus betoltes 2 perc elteltevel.
timeout=120
splashimage=(hd0,8)/boot/grub/splash.xpm.gz
```

```
title Windows 2000
unhide (hd0,0)
hide (hd0,1)
hide (hd0,2)
rootnoverify (hd0,0)
chainloader +1
makeactive
title Red Hat Linux (2.4.18-19.8.0.19mar2003)
root (hd0,8)
kernel /boot/bzImage.2.4.18-19.8.0.19mar2003 ro root=LABEL=/ hdd=ide-scsi
initrd /boot/initrd-2.4.18-19.8.0custom.img.19mar03
title Red Hat Linux (2.4.18-19.8.0custom)
root (hd0,8)
kernel /boot/vmlinuz-2.4.18-19.8.0custom ro root=LABEL=/ hdd=ide-scsi
initrd /boot/initrd-2.4.18-19.8.0custom.img
title Red Hat Linux (2.4.18-14)
root (hd0,8)
kernel /boot/vmlinuz-2.4.18-14 ro root=LABEL=/ hdd=ide-scsi
initrd /boot/initrd-2.4.18-14.img
title MyKernel.26jan03 (Red Hat Linux 2.4.18-14)
root (hd0,8)
kernel /boot/bzImage.myker.26jan03 ro root=LABEL=/ hdd=ide-scsi
initrd /boot/initrd-2.4.18-19.8.0.img
title Windows 98
hide (hd0,0)
hide (hd0,1)
unhide (hd0,2)
rootnoverify (hd0,2)
chainloader +1
makeactive
title DOS 6.22
hide (hd0,0)
unhide (hd0,1)
hide (hd0,2)
rootnoverify (hd0,1)
chainloader +1
makeactive
title Partition 2 (floppy)
hide (hd0,0)
unhide (hd0,1)
hide (hd0,2)
chainloader (fd0)+1
title Partition 3 (floppy)
hide (hd0,0)
hide (hd0,1)
unhide (hd0,2)
chainloader (fd0)+1
```

18. "D" függelék - utólagos rendszermag fordítás

Miután sikeresen felépítettük (build) és betöltöttük a Linux-rendszermagot, a következő kiegészítő lépésekre azért van szükség, hogy bizonyos eszközöket működésre bírjunk Linux alatt. (Az alábbi lépéseket RedHat Linuxon teszteltük, de más terjesztésekkel is működni kell.)

Videokártya/Monitor beállítása:

- Nézd meg a videokártya leírását, amit általában együtt szállítanak a PC-vel. Keresd meg a "Technical Specifications" ("Technikai paraméterek") oldalt.
- Nézd meg a monitor leírását, és keresd meg a "Technical Specifications" ("Technikai paraméterek") oldalt.

Ha a legújabb Linuxot (2.4 vagy későbbi) használod, kattints a KDE/GNOME munkaasztalon belül a Start->"System Settings"->Display menüpontra.

A Linux régebbi verzióinál kövesd az alábbi lépéseket:

A videokártyát és a monitort a következő parancsokkal állíthatod be:

```
bash$ su - root
bash# man Xconfigurator
bash# /usr/bin/X11/Xconfigurator --help
bash# /usr/bin/X11/Xconfigurator
bash# /usr/bin/X11/Xconfigurator --expert
See also:
bash# man xf86config
bash# /usr/bin/X11/xf86config
```

Ha a kártyádat nem ismerte fel automatikusan a rendszer, akkor használhatod a --expert opciót, és válaszd az "Unlisted card" (fel nem sorolt kártya) menüpontot. Ha a monitorod nincs felsorolva, akkor válaszd az általános SVGA 1024x768 típust.

Hangkártya beállítása:

- Kösd be a külső hangszórókat a hangkártya kimentére.
- Kösd össze a CD-ROM audiokábelét a hangkártya 4 tűs audio-aljzatával. (Egyébként a zenei CD-ket nem fogod hallani a kártyával.)
- Olvasd a hanggal foglalkozó HOGYANokat a "<http://www.tldp.org>" (<http://www.tldp.org>) webhelyen (továbbá a "HOGYAN-INDEX - MIDI, hangkártyák" (<http://tldp.fsf.hu/HOWTO/HOWTO-INDEX/hardware.html#HWSOUND>) honlapon - a lektor).

Ha a legújabb Linuxot (2.4 vagy későbbi) használod, kattints a KDE/GNOME munkaasztalon belül a Start->"System Settings"->Soundcard Detection menüpontra.

A régebbi Linux verzióknál kövesd az alábbi lépéseket:

```
bash$ su - root
bash# man sndconfig
bash# /usr/sbin/sndconfig
```

Majd indítsd el az X-Window rendszer "KDE-munkaasztalát" a "startx" paranccsal. Kattints a "K Start->ControlCenter->SoundServer->General->Test Sound" menüpontra. Ennek le kell játszania a teszthangot. Aztán kattints a "K Start->MultiMedia->SoundMixer->SoundVolumeSlider" menüpontra és állítsd be a hangerőt.

Hálózati kártya beállítása: Ha a legújabb Linuxot (2.4 vagy későbbi) használod, kattints a KDE/GNOME munkaasztalon belül a Start->"System Settings"->Network menüpontra.

A régebbi Linux verzióknál kövesd az alábbi lépéseket:

- Használd az /sbin/linuxconf programot
- vagy a KDE vezérlőpultot
- Olvasd a hálózattal foglalkozó HOGYANokat a "<http://www.tldp.org>" (<http://www.tldp.org>) webhelyen (továbbá a "HOGYAN-INDEX - Hálózat" (<http://tldp.fsf.hu/HOWTO/HOWTO-INDEX/networking.html>) honlapon - a lektor).

Tűzfal és IP-álcázás beállítása: A rendszermag 2.4-es és a feletti verzióinál a tűzfal és az IP-álcázás a NetFilter csomaggal lett megvalósítva. Ezért a rendszermag beállítása közben engedélyezned kell a Netfilter-t és futtatni a tűzfal/IPmaszk szkripteket. Töltsd le a szkripteket a Firewall-IPMasq scripts (<http://www.BoingWorld.com/workshops/linux/iptables-tutorial>) honlapról, a NetFilter címlapját a "<http://netfilter.samba.org>" (<http://netfilter.samba.org>) webhelyen találod. Idevágó anyagokat találsz a firewaling-matures (http://www.linuxsecurity.com/feature_stories/kernel-netfilter.html) és Netfilter-FAQ (<http://netfilter.filewatcher.org/netfilter-faq.html>) honlapokon (valamint magyarul, a Firewall LDP Team (<http://doc.skylinux.org/>) webhelyén - a lektor).

A 2.4 alatti verziókhöz telepítheted a tűzfal rpm csomagjait a rpmfind.net (<http://rpmfind.net/linux/rpm2html/search.php?query=firewall>) vagy a [firewall.src.rpm](http://rpmfind.net/linux/RPM/contrib/noarch/SRPMS/firewall-2.2-3.src.html) (<http://rpmfind.net/linux/RPM/contrib/noarch/SRPMS/firewall-2.2-3.src.html>) honlapról.

Egyéb eszközök beállítása: Olvasd az ezzel foglalkozó HOGYANokat a "<http://www.tldp.org>" (<http://www.tldp.org>) webhelyen (továbbá a "HOGYAN-INDEX - Hardver" (<http://tldp.fsf.hu/HOWTO/HOWTO-INDEX/hardware.html>) honlapon - a lektor).

19. "E" függelék - a gyakori hibák elhárítása

19.1. A rendszermag rendben elkészül, de a "make modules" nem sikerül

Jelenség: A rendszermag elkészül, és elkészül a bzImage, de a "make modules" már nem sikerül.

Megoldás: Ez a legtrükkösebb probléma, számos oka lehet. Ilyenek például a Linux terjesztés maga, nem frissített csomagfüggőségek. Ez nagyon jellemző a RedHat terjesztésre, de másikon is előfordulhat. Okozhatja továbbá néhány "ott felejtett" fájl is, amelyek felfüggesztik a programfordítási folyamatot, ezáltal problémát okoznak. Ennek ellenszere a "make mrproper" és "make clean", majd a "make modules" parancs kiadása. Szükség lehet a beállítási fájlok mentésére, alább látható ennek menete:

```
bash# cd /usr/src/linux
bash# mkdir /usr/src/kernelconfigs ;
bash# cp /usr/src/linux/.config /usr/src/kernelconfigs/.config.save;
bash# cp /usr/src/linux/.config /usr/src/linux/configs/.config.save # Különösen biztonságos
```

```
bash# cp /boot/config* /usr/src/linux/configs/ # Különösen biztonságos
bash# make clean
bash# make mrproper # "EL KELL VÉGEZNEDE ezt az mrproper-t", különben pokolian sok problémával
# kell szembenézned
bash# make clean
bash# cp /usr/src/kernelconfigs/.config.save .config # abban az esetben, ha újra fel akarod
# használni a beállítófájlt ??
```

19.1.1. Nem megfelelő beállítófájl

Ha az előző bekezdésben ajánlott "make mrproper" nem oldja meg a problémát, egyéb körmönfont probléma ejtett csapdájába. Talán valami komoly hiba van a beállítófájlban. Talán nem a processzorodnak megfelelő beállítófájlból indultál ki (talán ATHLON CPU típust választottál Pentium vagy Cyrix CPU típust Athlon processzoros gépedhez) Kezdd előlről az egészet, ha Athlon CPU-s az athlon.config vagy ha Intel 696 CPU-s géped géped van az i686.config fájlt másold le. Töltsd le az aktuális beállítófájlt az /usr/src/linux/configs könyvtárból

```
bash# cp /usr/src/linux/configs/kernel-2.4.18-i686.config /usr/src/linux/.config
Vagy Athlon processzorok esetében
bash# cp /usr/src/linux/configs/kernel-2.4.18-athlon.config /usr/src/linux/.config
```

Kövessd a Gyors lépések - a rendszermag fordítása (#quick_steps) fejezetben leírtakat.

19.1.2. A csomagok nincsenek szinkronban

Még mindig problémáid vannak? Ha a fenti bekezdésben leírtak nem oldották meg a problémát, egyéb körmönfont probléma ejtett csapdájába. Biztos vagy benne, hogy az összes csomagfüggőség rendben van? Az összes függő csomag szinkronban van a többivel? Telepítettél csomagot a "--nodeps" kapcsolóval? Automatizálhatod a csomagfüggőségek feloldását egy olyan nagyszerű eszközzel, mint az apt-get (Olvasd el a Telepítés, frissítés fénysebességgel (#apt-get) fejezetet.) Kézzel szinkronban tartani csomagok és programozói könyvtárak százait fene nagy munka, használd az apt-get programot.

19.2. A programfordítás rendben megtörténik, de a rendszermag nem indul

Jelenség: ha a rendszermag rendben lefordul, de nem töltődik be és mindig rendszermag pánikra panaszkodik valahol az /sbin/modprobe körül.

Megoldás: nem készítetted el az initrd fájlt. Olvasd el az "A" függelék - initrd.img fájl készítése (#create_initrd) fejezetet.

Az initrd elkészítésén kívül, ki kell adnod egy "make modules" és "make modules_install" parancsot. Még ha ki is adtad a "make modules" parancsot előtte, próbáld meg másodszor is lefuttatni (nem árthat). Add ki a "make modules" és "make modules_install" parancsokat még egyszer, hogy teljesen megbizonyosodj arról, miszerint a betölthető modulok a helyükre kerültek.

19.3. A rendszer működése felfüggesztődik a LILO-nál

Jelenség: Miután felépítetted a rendszermagot és újraindítottál, a rendszer várakozik épp a LILO előtt.

Ok: Valószínűleg nem állítottad be a BIOS-ban a megfelelő elsődleges mester IDE és másodlagos szolga IDE merevlemez partíciót.

Megoldás: Kapcsold be a gépet és nyomd meg a DEL gombot a BIOS (Basic Input Output System) Setup menübe történő belépéshez. Válaszd az IDE beállításokat és állítsd be a megfelelő elsődleges merevlemez partíciót és a szolga-meghajtókat. Amikor a rendszer indul, megkeresi az elsődleges IDE merevlemezt és a Master Boot Record partíciót. Kiolvassa az MBR-t és elkezd betölteni a Linux-rendszermagját a merevlemez partíciójáról.

19.4. No init found (nem található init)

A következő hibát gyakran követik el a kezdő felhasználók.

Ha az új rendszermagod nem indul el, és a következőt hibaüzenetet kapod:

```
Warning: unable to open an initial console
Kernel panic: no init found. Try passing init= option to kernel
```

Az a probléma, hogy *nem állítottad be megfelelően* a "root=" paramétert az /etc/lilo.conf fájlban. Az én esetemben, a root=/dev/hda1 partíciót használom, amin a gyökér "/" partíció van. Jól kell beállítanod a root-eszközt, olyasminek kell lennie, mint /dev/hdb2 vagy /dev/hda7.

Lehetnek hibák ez előtt a rendszermag pánik előtt is. Nézd meg és olvasd el a lehetséges hibaüzeneteket a "Kernel panic:" felirat előtt. A hibát okozhatja bármely ez előtti hiba is (összegződő hatás). Például a "Kernel panic:" hibaüzenet előtt láthatsz olyat is, hogy "kernel-module version mismatch" (a rendszermag-modul verzió nem egyezik) vagy "ilyen-olyan-egyéb-hibaüzenet"-eket is. Próbáld meg az ELSŐ, rendszer által jelzett hibát kijavítani.

A rendszermag az init parancsot az /sbin/init alatt keresi. Az /sbin könyvtár pedig a gyökér-partíción van. További részleteket a

```
bash# man init
```

parancs kiadásával, valamint a "C" függelék - GRUB részletesen, grub.conf mintafájl (#grubconf) és "B" függelék - lilo.conf mintafájl (#liloconf) fejezetek olvasásával tudhatsz meg.

19.5. Csomó fordítási hiba (compile error)

A "make", "make bzImage", "make modules" vagy "make modules_install" fordítási hibákat jelez. Add ki a "make mrproper" parancsot a "make" parancs kiadása előtt.

```
bash# make clean && make mrproper # "KÖTELEZŐ KIADNOD AZ mrproper parancsot", egyébként problémák sz
```

Ha a probléma változatlanul fennáll, próbáld ki a "menuconfig"-ot az "xconfig" helyett. Néha a GUI adott verziója problémázik az "xconfig"-al:

```
bash# export TERM=VT100
bash# make menuconfig # Újabb, az "ncurses"/"curses"-t használja, ha nincs telepítve nem működik
```

19.6. A "depmod" parancs "Unresolved symbol error messages" hibaüzenetet ír ki

A `depmod` parancs futásakor "Unresolved symbols" hibaüzenetet ír ki. Az alábbi példa mutatja be az esetet:

```
bash$ su - root
bash# man depmod
bash# depmod
depmod: *** Unresolved symbols in /lib/modules/version/kernel/drivers/md/linear.o
depmod: *** Unresolved symbols in /lib/modules/version/kernel/drivers/md/multipath.o
depmod: *** Unresolved symbols in /lib/modules/version/kernel/drivers/md/raid0.o
depmod: *** Unresolved symbols in /lib/modules/version/kernel/drivers/md/raid1.o
depmod: *** Unresolved symbols in /lib/modules/version/kernel/drivers/md/raid5.o
```

Ok: Nem fordítottad és telepítetted a modulokat az új rendszermag elkészítése (`"make bzImage"`) után.

Megoldás: Az új rendszermag elkészítése után muszáj ezt tenned:

```
bash$ su - root
bash# cd /usr/src/linux
bash# make modules
bash# make modules_install
```

19.7. A rendszermag nem tölti be a modult, "Unresolved symbols" hibaüzenetet ír ki

Amikor betöltöd a rendszert, és az bármely modult megpróbálva betölteni a "Unresolved symbol : `__some_function_name`" üzenetet ír ki, akkor ez azt jelenti, hogy nem "tisztá helyzetből kiindulva" fordítottad a modulokat és a rendszermagot. Elengedhetetlen, a `make clean` parancs kiadása, majd a modulok fordítása. Ezt az alábbi parancsok kiadásával teheted meg:

```
bash# cd /usr/src/linux
bash# make dep
bash# make clean
bash# make mrproper # "MUST DO THIS mrproper", otherwise you will face hell lot of problems !!
bash# make clean
bash# nohup make bzImage &
bash# tail -f nohup.out      (... to monitor the progress)
bash# make modules
bash# make modules_install
```

19.8. A rendszermag nem tud betölteni egy modult

Ha a rendszermag nem tud betölteni egy modult (mondjuk egy hálózati kártyáét vagy más eszközt), akkor megpróbálhatod az eszközt közvetlenül a rendszermagba fordítani. Néha a *betölthető modul NEM működik* és a meghajtót fixen a rendszermagba kell fordítani. Például - néhány hálózati kártya nem támogatja a betölthető modul szolgáltatást - egyből a rendszermagba KELL fordítanod. Ezért a "make xconfig"-ban NEM SZABAD a betölthető modul opciót választani ehhez az eszközhöz.

19.9. Betölthető modulok

Az alapértelmezett betölthető modulokat telepítheted így:

Az alább megadott lépés nem szükséges, de *HIBA ESETÉN SZÜKSÉG LEHET RÁ*, amikor a /lib/modules fájlljai megsérültek. Ha már létezik a /lib/modules könyvtár, és ki akarsz cserélni a tartalmát, használd a "--force" kapcsolót a csomag lecseréléséhez és válaszd a megfelelő CPU architektúrát.

A RedHat Linux új verzióiban (mint a 6.0 vagy későbbi) a rendszermag-modulokat a kernel-2.2*.rpm tartalmazza. Telepítsd a rendszermagot és a modulokat:

Ez kilistázza a már telepített csomagokat.

```
bash# rpm -qa | grep -i kernel
```

```
bash# rpm -U --force /mnt/cdrom/Redhat/RPMS/kernel-2.2.14-5.0.i686.rpm
(or)
```

```
bash# rpm -U --force /mnt/cdrom/Redhat/RPMS/kernel-2.2.14-5.0.i586.rpm
(or)
```

```
bash# rpm -U --force /mnt/cdrom/Redhat/RPMS/kernel-2.2.14-5.0.i386.rpm
```

Ez csak a régi, 5.2 és az előtti verziókhoz szól. Indíts az új rendszermaggal és telepítsd a betölthető modulokat a RedHat "contrib" CD-ROM-ról: cdrom

```
bash# rpm -i /mnt/cdrom/contrib/kernel-modules*.rpm
```

...(A régi Linux rendszerekhez, amikben nincs előre telepítve az insmod)

19.10. Olvasd el a dokumentációt

Ha további problémák vannak, elolvashatod az /usr/src/linux/README (legalább egyszer), valamint az /usr/src/linux/Documentation fájlt is.

```
bash [/] # cd /usr/src/linux/Documentation
```

```
bash [/usr/src/linux/Documentation] # ls *.txt
```

```
binfmt_misc.txt      ioctl-number.txt      nbd.txt               serial-console.txt
cachetlb.txt         IO-mapping.txt        nfsroot.txt           sgi-visws.txt
cciss.txt             IRQ-affinity.txt      nmi_watchdog.txt     smart-config.txt
```

computone.txt	isapnp.txt	oops-tracing.txt	smp.txt
cpqarray.txt	java.txt	paride.txt	sonypi.txt
devices.txt	kernel-doc-nano-HOWTO.txt	parport-lowlevel.txt	specialix.txt
digiboard.txt	kernel-docs.txt	parport.txt	spinlocks.txt
digiepca.txt	kernel-parameters.txt	pci.txt	stallion.txt
DMA-mapping.txt	kmod.txt	pcwd-watchdog.txt	svga.txt
dnotify.txt	locks.txt	pm.txt	swsusp.txt
exception.txt	logo.txt	ramdisk.txt	sx.txt
floppy.txt	magic-number.txt	riscom8.txt	sysrq.txt
ftape.txt	mandatory.txt	rtc.txt	unicode.txt
hayes-esp.txt	mca.txt	SAK.txt	VGA-softcursor.txt
highuid.txt	md.txt	sched-coding.txt	watchdog-api.txt
i810_rng.txt	memory.txt	sched-design.txt	watchdog.txt
ide.txt	modules.txt	scsi-generic.txt	zorro.txt
initrd.txt	mtrr.txt	scsi.txt	

19.11. make clean

Ha az új rendszermag valóban furcsa dolgokat csinál egy rutinszerű frissítés után, esélye van annak, hogy elfelejtetted kiadni a `make clean` parancsot az új rendszermag fordítása előtt. A jelenségek kiterjedhetnek bármire az egyből lefagyástól, a különös I/O hibákon át a csapnivaló teljesítményig. Bizonyosodj meg, hogy kiadtad a `make dep` parancsot is.

19.12. Óriási vagy lassú rendszermag

Ha a rendszermagod sok memóriát zabál, túl nagy, és/vagy örökké tart a programfordítása még az új Quadbazillium-III/4400-as gépeden is, akkor valószínűleg sok felesleges cuccot (eszközmeghajtót, fájlrendszert stb.) tettél bele. Ha nem használod, ne állítsd be, mivel memóriát foglal. A legkézenfekvőbb jelenség a rendszermag felfűvódására a memória állandó oda- visszatöltése a lemezzről; ha a lemezed sok zajt bocsát ki, és nem egyike a régi Fujitsu Eagles-eknek (ami kikapcsoláskor olyan hangot ad ki, mint egy leszálló repülőgép), akkor nézd át a beállításaidat.

Megállapíthatod, hogy mennyi memóriát használ a rendszermag, ha veszed a teljes memóriamennyiséget és kivonod belőle a "total mem" értékét, amit a `/proc/meminfo` mutat, vagy a `free` parancs.

19.13. A párhuzamos port/nyomtató nem működik

A PC-k beállítási lépései: először is, a "General Setup" kategóriában válaszd a "Parallel port support" és "PC-style hardware" beállításokat. Aztán a "Character devices" alatt, válaszd a "Parallel printer support"-ot.

Ezután következnek a nevek. A Linux 2.2 máshogy nevezi a nyomtatóeszközöket, mint a korábbi kiadások. Ennek az a következménye, hogy ha `lp1` eszközöd van a régi rendszermag alatt, ez valószínűleg `lp0` az új verziónál. Használj a `dmesg` parancsot, vagy nézd meg a naplókat a `/var/log` könyvtárban, hogy kiderítsd az eszköz nevét.

19.14. A rendszermag nem fordul le

Ha nem fordul le, akkor lehetséges, hogy egy foltozás nem sikerült, vagy a forrásod valamiért nem jó. A gcc verzió szintén nem biztos, hogy megfelelő, vagy szintén nem jó (például az include fájlok lehetnek hibásak). Győződj meg, hogy a Linus által leírt szimbolikus linkek, amiket a README fájlban ír le, jól vannak beállítva. Általánosan, ha egy hagyományos rendszermag nem fordul le, akkor valami komoly gond van a rendszerben, és bizonyos eszközök újbóli telepítése válhat szükségessé.

Néhány esetben, a gcc hardver problémák miatt szállhat el. A hibaüzenet olyasmi lehet, hogy "xxx exited with signal 15" és ez általában nagyon rejtélyesen néz ki. Valószínűleg nem említettem volna, de megtörtént velem, egyszer volt egy kevés rossz cache-memóriám, és időnként a fordító véletlenszerűen elhányta magát. Először próbáld meg a gcc-t kicserélni, ha problémád van. Kisakkozhatod, hogy lefordul-e a rendszermag a külső gyorsítótár kikapcsolásával, csökkentett méretű RAM-mal stb.

Fel tudja zaklatni az embereket, ha azt mondják nekik, hogy rossz a hardverük. Nos, én nem adom fel. Létezik egy GYIK is erről, ezen a "<http://www.bitwizard.nl/sig11>" (<http://www.bitwizard.nl/sig11>) webhelyen.

19.15. A rendszermag új verziója nem töltődik be

Nem futtattad a LILO-t, vagy nincs rendesen beállítva. Egy dolog, ami "megfogott" engem egyszer, egy konfigurációs fájl problémája volt: ez volt benne: " boot = /dev/hda1 " ahelyett, hogy " boot = /dev/hda " lett volna. (Ez először tényleg zavaró lehet, de ha egyszer van egy működő beállítófájlod, nem kell megváltoztatnod).

19.16. Elfelejtetted futtatni a LILO-t, vagy a rendszered egyáltalán nem indul el

Hoppá! A legjobb dolog, amit ekkor tehetsz, hogy hajlékonylemezről vagy CD-ROM-ról indítasz, és készítesz egy másik indítólemezt (amit a " make zdisk " paranccsal is megtehetsz). Tudnod kell, hol van a gyökér (/) fájlrendszered és milyen típusú (tehát például ext2, minix). Az alábbi példában azt is tudnod kell, milyen fájlrendszeren van a /usr/src/linux forrásfád, ennek típusát, és normál esetben hova csatolódik fel.

A következő példában a / a /dev/hda1 , és a fájlrendszer, ami tartalmazza a /usr/src/linux könyvtárat, a /dev/hda3 , normál esetben a /usr alá van felcsatolva. Mindkettő second extended (ext2) fájlrendszer. A működő rendszermag helye a /usr/src/linux/arch/i386/boot könyvtár, és bzImage a neve.

Az ötlet az, hogy ha van egy működő bzImage , akkor azt használhatod egy új hajlékonylemez készítéséhez. Egy másik lehetőséget, ami vagy jobban működik, vagy nem (attól az egyedi módszertől függ, amivel szétbarmoltad a rendszered) beszélünk meg a példa után.

Először indíts egy boot/root lemezpárosról vagy mentőlemezezz, és csatold fel a működő rendszermagot tartalmazó fájlrendszert:

```
mkdir /mnt mount -t ext2 /dev/hda3 /mnt
```

Ha az mkdir azt írja ki, hogy a könyvtár már létezik, ne törődj vele. Most a cd paranccsal lépj be arra a helyre, ahol a működő rendszermag van. Figyeld meg, hogy /mnt + /usr/src/linux/arch/i386/boot - /usr = /mnt/src/linux/arch/i386/boot. Helyezz egy formázott lemezt az "A" meghajtóba (ne a boot vagy root lemezed!), másold ki a fájlt a lemeze, és állítsd be a gyökér fájlrendszeredhez:

```
cd /mnt/src/linux/arch/i386/boot dd if=bzImage of=/dev/fd0 rdev /dev/fd0 /dev/hda1
```

A cd paranccsal lépj be a / könyvtárba és válaszd le a normál /usr fájlrendszert:

```
cd / umount /mnt
```

Most már képesnek kell lenned normál módon újraindítani a rendszert erről a hajlékonylemezeződől. Ne felejtse el futtatni a lilo-t (vagy bármi volt, amit elrontottál) az újraindítás után!

Amint fentebb említettük, van egy másik általános lehetőség. Ha véletlenül van egy működő rendszermag a / könyvtárban (/vmlinuz például), használhatod azt is a indítólemezehez. Feltéve, hogy teljesül az összes fenti feltétel, és a rendszermagod a /vmlinuz , csak ezeket a változásokat tedd meg a fenti példához képest: változtasd meg a /dev/hda3 -at /dev/hda1 -re (a / fájlrendszerre), az /mnt/src/linux -ot a /mnt -re, és a if=bzImage -et if=vmlinuz -ra. A jegyzet arról, hogyan származtatjuk a /mnt/src/linux -ot, figyelmen kívül hagyható.

A LILO használata nagy meghajtókkal (több mint 1024 cilinderrel) problémákat okozhat. Olvasd a LILO mini-HOWTO (LILO mini HOGYAN) , vagy egyéb dokumentációt ennek a kivédéséről.

19.17. Azt írja ki: "warning: bdflush not running"

Ez komoly probléma lehet. A rendszermag v1.0-ás verziójától kezdve (1994. ápr. 20-körül), az " update " nevű programot, ami rendszeresen üríti a fájlrendszer puffereit, fejlesztették/kicserélték. Szerezd meg a " bdflush " forrását (ott találsz, ahol a rendszermag forrását), és telepítsd fel (valószínűleg a régi rendszermaggal futtatod a rendszert, amíg ezt teszed.) Ez önmagát " update " néven telepíti, és miután újraindítottad a rendszert, az új rendszermagnak már nem szabad panaszkodnia.

19.18. Nem tudom működéskére bírni az IDE/ATAPI CD-ROM-ot

Különös módon sokan nem tudják működéskére bírni az ATAPI meghajtóikat, valószínűleg mert több dolog sem stimmel.

Ha a CD-ROM az egyetlen eszköz egy bizonyos IDE csatolón, akkor "master"-ként és nem "slave"-ként kell beállítani. Meglepő, de ez a legáltalánosabb hiba.

A Creative Labs (elsőként) IDE csatolót rakott a hangkártyáira. Ez azonban ahhoz az érdekes problémához vezetett, hogy míg néhányan csak egy csatolóval rendelkeztek, sokan viszont kettővel, beépítve az alaplapon (általában a 15-ös IRQ-n), így egy általános megoldás lett a SoundBlaster csatolóját a harmadik IDE portnak venni (IRQ11, legalábbis így mondták).

Ez a régi, 1.3-as és az alatti Linux-verzióknál gondot okozott. Ezekben a verziókban a Linux nem támogatta a harmadik IDE csatolót. Ennek megkerülésére kevés lehetőség van.

Ha már van egy második IDE port, van esély rá, hogy nem használod, vagy nincs még rajta két eszköz. Vedd le az ATAPI meghajtót a hangkártyáról és rakd a második csatolóra. Ezek után le tudod tiltani a hangkártya csatolóját, ami egyébként egy IRQ-t is megspórol.

Ha nincs második csatolód, jumperrel állítsd be a hangkártya csatolóját (ne a kártya hang-részét) az IRQ15-re, a második csatolóra. Ennek működni kell.

19.19. Furcsa dolgokat jelez elavult útválasztási kérésekről (routing requests)

Szerezd be a route program újabb verzióját és bármely egyéb programét is, ami útválasztási információkat kezel. Az /usr/include/linux/route.h (ami igazából egy fájl a /usr/src/linux könyvtárban) megváltozott.

19.20. "Not a compressed kernel image file" (nem tömörített rendszer-mag)

Ne használj betöltőfájlként a `vmlinux` fájlt, ami a `/usr/src/linux` könyvtárban van, a `[..]/arch/i386/boot/bzImage` a megfelelő.

19.21. Problémák a konzolos terminállal, miután 1.3.x-re frissítettünk

Változtasd meg a `dumb` szót `linux`-ra a konzol termcap bejegyzésében, a `/etc/termcap` fájlban. Ezen kívül egy `terminfo` bejegyzést is létre kell hoznod.

19.22. Úgy néz ki, hogy nem fordíthatók le dolgok a rendszer-mag frissítése után

A Linux-rendszer-mag forrása tartalmaz számos ún. `include` fájlt (azok a dolgok, amik `.h`-vel végződnek), amikre a standard `/usr/include` könyvtárban lévő fájlok is hivatkoznak. Általában a következő módon hivatkoznak rájuk (ahol a `xyzy.h` valamilyen fájl a `/usr/include/linux` könyvtárban): `#include <linux/xyzy.h>` Normál esetben van egy `linux` nevű link a `/usr/include` könyvtárban az `include/linux` könyvtárra, ami a forráson belül van (`/usr/src/linux/include/linux` egy tipikus rendszeren). Ha ez a link nincs ott, vagy rossz helyre mutat, a legtöbb dolog egyáltalán nem fordítódik le. Ha úgy döntesz, hogy a rendszer-mag forrása túl sok helyet foglal és letörlöd, ez bizony probléma lehet. Egy másik dolog, ami rossz lehet, a fájlok tulajdonjogai; ha a `root` felhasználónak olyan fájlmaszkja van, ami nem engedi meg alapértelmezésben a többi felhasználónak, hogy lássa a fájljait, és a rendszer-mag forrását a `p` (`preserve filemodes`) opció nélkül csomagoltad ki, ezek a felhasználók nem tudják használni a C fordítót sem. Bár használhatod a `chmod` parancsot ennek kijavítására, valószínűleg egyszerűbb újra kicsomagolni az `include` fájlokat. Ezt ugyanúgy teheted meg, ahogy a teljes forrást az elején, csak egy kiegészítő argumentummal:

```
blah# tar zxvpf linux.x.y.z.tar.gz linux/include Figyelem: a " make config " újra létrehozza a
/usr/src/linux linket, ha az nincs ott.
```

19.23. Korlátok kitolása

A következő néhány *példa* parancs hasznos lehet azoknak, akik kíváncsiak arra, hogyan kell megemelni néhány változtatható korlátot, amit a rendszer-mag ránk kényszerít:

```
echo 4096 > /proc/sys/kernel/file-max
echo 12288 > /proc/sys/kernel/inode-max
echo 300 400 500 > /proc/sys/vm/freepages
```

19.24. Hová küldjem a hibajelentést?

Részletesen a Gyors lépések - Rendszer-mag-fordítás (#quick_steps) fejezet Hová küldjem a hibajelentést? (#reportbugs) alfejezetében olvashatsz erről.