

Linux + XFS HOGYAN

Szteroiddal fűtött Linux

Russel Ingram

ringram@gargoylecc.com

Ez a dokumentum leírja, hogyan építs fel egy SGI XFS naplózó fájlrendszeren futó Linux rendszert.

Tartalomjegyzék

1. Bevezetés	2
1.1. Az XFS naplózó fájlrendszer bemutatása	2
1.2. Előszó, visszajelzés, készítőik	2
1.3. Szerzői jog és a felhasználás feltételei	2
1.4. Magyar fordítás	2
2. Előkészületek az XFS támogatás telepítéséhez.....	3
2.1. A Linux 2.4.x-XFS forráskód letöltése	3
2.2. XFS támogatás: kernelbe fordítva vagy modulként betöltve?.....	4
3. Kernel beállítás és telepítés	5
3.1. A kernel beállítása az XFS támogatáshoz	5
3.2. A kernel és a modulok fordítása.....	5
3.3. Az új kernel és modulok telepítése.....	6
3.4. Egy új bejegyzés hozzáadása a lilo beállításokhoz, a beállítások érvényesítése.....	6
3.5. Az XFS segédprogramok fordítása és telepítése	6
3.6. Az új kernel indítása.....	7
4. A fájlrendszer lecserélése	7
4.1. A gyökerkönyvtár átmásolása	7
4.2. Utolsó simítások	9

1. Bevezetés

1.1. Az XFS naplózó fájlrendszer bemutatása

Ez a dokumentum leírja, hogyan lehet egy Linux rendszert felépíteni és futtatni az SGI XFS naplózó fájlrendszeren. Idézet az XFS GYIK-ból: "Az XFS naplózó fájlrendszert az SGI fejlesztette ki az SGI IRIX operációs rendszerhez történő alkalmazásra, és most a GPL keretein belül a Linux számára is használhatóvá vált. A fájlrendszer különösen jól méretezhető, kihasználja a B-fa adta lehetőségeket a nagyméretű, több részletben tárolt fájlok, és a különösen nagyméretű könyvtárak tárolásához. A naplózó képesség azt jelenti, hogy búcsút mondhatunk az fsck hibajavítására történő várakozásra, vagy a fájlbejegyzési hiba miatti aggodalomnak." Az XFS naplózó fájlrendszert az SGI alapvetően a nagyteljesítményű kiszolgáló rendszerek számára tervezte, ezért kapta a jelen leírás a "Sztteroiddal fűtött Linux" alcímet.

1.2. Előszó, visszajelzés, készítőik

Az Irix rendszeradminisztrátorok közösségének egészen friss tagjaként valósággal szerelmese lettem az Irix számára kifejlesztett megbízható fájlrendszernek (amely természetesen az XFS). Mondanom sem kell, figyelemmel kísértem a fájlrendszer Linux-ra átírásáról szóló híreket, és már a kezdetektől fogva arról álmodoztam, hogy Linux rendszeremet csakis XFS fájlrendszeren futtatom. A projekt már elérkezett arra a pontra, amikor ez valósággá válhat (valójában ez már közel egy évvel a jelen írás megszületése előtt megtörtént). A nagyszámú programkészítési/programátültetési projekt ellenére a dokumentációk nem mindig olyan könnyen fellelhetők, vagy érthetőek mint szeretnénk. Ez a leírás ezt a helyzetet próbálja meg orvosolni.

A jelen leírásban található információk Jason Walker és Russel Cattelan által az XFS fejlesztői listára küldött levelek gyűjteményén, az SGI Open Source Development weboldalakon összegyűjtött információkon, valamint saját (Russel Ingram) beállítási tapasztalataimon alapul.

Küldj nyugodtan egy e-mailt erre a címre: <ringram@gargoylecc.com> ha javítani valót találsz benne, vagy valamilyen információ/URL illetve egyéb hiányzik belőle. Minél több visszajelzést kapok erről HOGYANról, annál több segítséget tudok nyújtani mindenkinek.

A leírás legfrissebb angol nyelvű változatát a következő helyen találod: Gargoyle Computer Consulting (<http://www.gargoylecc.com/Linux+XFS-HOWTO/t1.html>).

1.3. Szerzői jog és a felhasználás feltételei

A szerzői jogokat Russel Ingram birtokolja; copyright (c) 2001; ez egy INGYENES leírás. A GPL feltételei szerint te is szabadon terjesztheted.

Ennek a leírásnak a tartalma Russel legjobb tudása szerint megfelel a valóságnak. Mivel azonban az XFS linuxos átíratát emberek készítették el, így lehetnek benne tévedések, hibák ezek időnként előfordulnak.

Semmilyen személy, csoport vagy egyéb szervezet nem felelős a számítógépe(i)dben esett bármilyen kárért, amit az ebben a leírásban közölt információk felhasználása okozott. Azaz:

A SZERZŐ NEM VÁLLAL FELELŐSSÉGET SEMMILYEN KÁRÉRT, AMELYET A JELEN LEÍRÁSBAN KÖZÖLT INFORMÁCIÓKRA ALAPOZOTT TEVÉKENYSÉG OKOZOTT.

1.4. Magyar fordítás

A magyar fordítást Daczi László (mailto:dacas@freemail.hu_NO_SPAM) készítette (2002.05.30). Bármilyen fordítással kapcsolatos észrevételt a linuxhowto@sch.bme.hu (mailto:linuxhowto@sch.bme.hu_NO_SPAM) címre küldjétek. A lektorálást Varga Zsolt (mailto:tech.translation@axelero.hu_NO_SPAM) végezte el (2002.06.27). A fordítás javítva 2002.10.29.-én. A dokumentum legfrissebb változata megtalálható a Magyar Linux Dokumentációs Projekt (<http://tldp.fsf.hu/index.html>) honlapján.

2. Előkészületek az XFS támogatás telepítéséhez

2.1. A Linux 2.4.x-XFS forráskód letöltése

Az SGI Open Source Development oldalai jelenleg az egyetlen olyan hely, ahol elérhető az XFS támogatás Linux alatt használható forráskódja. Innen CVS-en keresztül töltheted le az anyagokat

Megjegyzés: jelenleg két különböző változat érhető el:

- linux-2.5-xfs: fejlesztői változat
- linux-2.4-xfs: stabil, hibamentes változat

Az én tapasztalataim a 2.4 verzió alapulnak, de úgy gondolom, a fejlesztői változatban is minden ugyanúgy működik. Mindkét változatot szinkronizálják a saját rendszermag-ágával, legalábbis annak fő változataival.

A kernel forrás letöltéséhez szükséges lépések a következők:

A. A Linux kernel forrása általában az /usr/src könyvtárban van, tehát először lépj be ebbe a könyvtárba (vagy ahová a kernel forrást telepítetted - a ford.)

```
$ cd /usr/src
```

B. Állítsd be a CVSROOT környezeti változót, hogy az a megfelelő CVS kiszolgálóra mutasson

- ha sh, bash, ksh, stb-t használasz...:

```
$ export CVSROOT=:pserver:cvs@oss.sgi.com:/cvs'
```

- Csh vagy tcsh használata esetén:

```
$ setenv CVSROOT :pserver:cvs@oss.sgi.com:/cvs
```

Ha gyakran szeretnéd frissíteni a kernelt, érdemes ezt beletenni a login scriptbe.

C. Jelentkezz be a CVS kiszolgálóra.

```
$ cvs login (a jelszó: "cvs")
```

Ezt csak EGYSZER kell megtenned, nem minden alkalommal, amikor ezt a CVS-t használod.

D. Töltsd le a linux-2.4-xfs forráskódot. Az első alkalommal ez valahogy így néz ki:

```
$ cvs -z3 co linux-2.4-xfs
```

Legközelebb csak le kell ellenőrizni a forrást, valahogy így...

```
$ cvs -z3 update linux-2.4-xfs
```

...majd frissíteni a nálad lévő forráskódot a CVS kiszolgálón lévő legfrissebb változatra.

2.2. XFS támogatás: kernelbe fordítva vagy modulként betöltve?

Az XFS támogatást modulként is lehetséges kernelbe fordítani, ami aztán RAM diszk és néhány lilo konfigurációs beállítás segítségével működik is (nekem legalábbis így mondták). Én nem próbáltam ezt ki egyelőre, tehát nem tudom leírni ennek a menetét, ezért idézek Russel Cattelan fejlesztői levelezőlistára beküldött leveléből:

Russell Cattelan

Valójában nem nehéz az xfs támogatást modulként futtatni. A cmd/xfs/misc könyvtárban van egy módosított mkinitrd, amely mindig előállít egy RAM diszket az xfs és xfs_support számára.

Elég egyszer hozzáadni az initrd betöltésére vonatkozó bejegyzést a lilo.conf állományban ÉS:

```
append = "ramdisk_size=25000"
```

Az alapértelmezett méret 4096, ami közel sem elég az xfs támogatáshoz.

Részlet a hordozható gépem beállításainak listájából:

```
punch[12:57am]==>mount
/dev/ide/host0/bus0/target0/lun0/part8 on / type xfs (rw,noatime)
none on /proc type proc (rw)
/dev/ide/host0/bus0/target0/lun0/part6 on /boot type ext2 (rw,noatime)
none on /dev/pts type devpts (rw,mode=0620)
/dev/ide/host0/bus0/target0/lun0/part1 on /mnt/windows type vfat (rw,nosuid,nodev,umask=0)
/dev/ide/host0/bus0/target0/lun0/part9 on /blam type xfs (rw)

punch[12:57am]==>lsmod
Module                Size Used by
autofs                 13180   1 (autoclean)
usb-uhci               24918   0 (unused)
```

```

usbcore          35339    0 [usb-uhci]
3c59x            25149    1 (autoclean)
maestro          29757    0 (unused)
soundcore        6085     2 [maestro]
vfat             13075    1 (autoclean)
fat              37733    0 (autoclean) [vfat]
xfs              447888    2
xfs_support      13954    0 [xfs]
pagebuf          39935    2 [xfs]

```

```

image=/boot/vmlinuz-2.4.0-XFS-test13-pre4
label=t13p4
root=/dev/hda8
initrd=/boot/initrd-2.4.0-XFS-test13p4.img
append="ramdisk_size=25000"
read-only

```

Ez alapján azt gondolom nem túl nehéz a támogatást modulként a kernelbe fordítani, ez az, amit én is tenni fogok. Megpróbálom majd így használni később, és további részletekkel szolgálok ennek megvalósításáról. Ha valakinek van ideje leírni ennek menetét mielőtt én megtenném ezt, kérem, küldje el e-mailben, én majd beleteszem a leírásba, a szerzőt megillető jutalékkal együtt :-))

3. Kernel beállítás és telepítés

3.1. A kernel beállítása az XFS támogatáshoz

Megjegyzés: Ha még soha nem állítottál be és fordítottál új linux kernelt, ajánlott elolvasni a Linux Kernel HOGYANt, mielőtt ezeket a lépéseket végrehajtod. A Linux Kernel HOGYANt megtalálod a Linux Documentation Project (LDP) (<http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html>) oldalon vagy valamelyik tükrözésén.

Miután letöltötted a CVS könyvtárból az aktuális kernel forrását, annak a `/usr/src/linux-2.4-xfs(-beta)/linux` könyvtárban kell lennie. Szóval lépj be ebbe a könyvtárba mielőtt kiadnád a `make config` (méginkább a `make menuconfig` - a ford.) parancsot a beállítások elvégzéséhez. Az XFS támogatáshoz engedélyezned kell a "Prompt for development and/or incomplete code/drivers" beállítást a "Code maturity level options" fejezetben majd a "Page Buffer support" és "SGI XFS filesystem support" menüpontokkal kell ezt megtenned (mindkettő a "File systems" fejezetben található). Lehetőség van még kiválasztani az "Enable XFS Debug mode" és "Enable XFS Vnode Tracing" beállításokat az "SGI XFS filesystem support" fejezet alatt. Ezek lassíthatják az XFS fájlrendszered használatát, de hasznosak lehetnek egy összeomlás esetének kivizsgálásában, ha ilyen előfordulna.

3.2. A kernel és a modulok fordítása

Minden kernel fordításkor ki kell adni az alábbi parancsokat az új kernel (és modulok - a ford.) létrehozása érdekében:

```
$ make dep
$ make bzImage
$ make modules
```

3.3. Az új kernel és modulok telepítése

Ezt minden kernel telepítéskor újra meg kell csinálni:

```
$ make modules_install
$ cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.0-XFS
```

3.4. Egy új bejegyzés hozzáadása a lilo beállításokhoz, a beállítások érvényesítése

```
$ vi /etc/lilo.conf
```

Hozz létre egy új kernelindító bejegyzést a lilo.conf állományban ehhez hasonlóan:

```
image=/boot/vmlinuz-2.4.0-XFS label=xfs read-only root=/dev/hda2
```

A "root=" szöveg utáni részt másold át egy szintén linuxot indító bejegyzésből, a lilo.conf állományból. Ne felejtse el elindítani a lilo-t a lilo.conf fájl szerkesztése után, hogy a módosítások érvénybe lépjenek (a parancssorba géped be "lilo" majd üss egy "enter"-t - a ford.)

3.5. Az XFS segédprogramok fordítása és telepítése

Az XFS támogatással számos eszközt is kapsz, amikkel felépítheted és karbantarthatod a fájlrendszert. Ezek a /usr/src/linux-2.4-xfs(-beta)/cmd/xfspgms könyvtárban vannak.

Megjegyzés: Ezek az eszközök a /usr/lib/libuuid.a osztott programozói könyvtárra épülnek. Ha a könyvtárat még nem telepítetted, akkor az XFS eszközök lefordítása előtt feltétlenül tedd ezt meg. Az ezt tartalmazó rpm csomagot megtalálod a Rpmfind.net (<http://www.rpmfind.net>) címen, ha rákérselek a "/usr/lib/libuuid.a." szövegre, a Debian csomag neve pedig uuid-dev. Más disztribúciókban ez a programozói könyvtár ezektől eltérő csomagban is lehet. Egy jó módszer a megfelelő csomag megkeresésére, ha a Google Linux (<http://www.google.com/linux>) keresőmotort használod.

Lépj be ebbe a könyvtárba:

```
$ cd ../cmd/xfsprogs
```

Fordítsd le, és telepítsd az XFS eszközöket:

```
$ make install
```

3.6. Az új kernel indítása

```
$ reboot
```

Megjegyzés: Ha nem állítod be az alapértelmezett kernelnek az XFS támogatással fordított kernelt, akkor bootoláskor be kell gépelned az "xfs" szöveget, majd leütnöd egy enter-t, amikor megjelenik a "LILO Boot:" szöveg a képernyőn.

4. A fájlrendszer lecserélése

Az eljárás befejező része talán a legtrükkösebb, és ez vezethet a legkönnyebben adatvesztéshez. Ajánlott egy biztonsági másolatot készítése a teljes rendszerről (vagy legalább az összes fontos adatról), mielőtt megkísérléd azt az XFS fájlrendszerre másolni. Még példát írni is nehéz erről, mivel több száz lehetséges módszer létezik. Megadom az alapvető parancsokat, amikkel általánosságban új fájlrendszert hozhatsz létre, valamint átadom a saját fájlrendszerem átmásolásából származó tapasztalatokat.

4.1. A gyökérkönyvtár átmásolása

Az XFS rendszernek a gyökérkönyvtár átmásolása talán a legtrükkösebb része az eljárásnak, mivel ezen van a rendszer és nem tudod leválasztani (értsd: umount - a ford.) mialatt a linux fut. Ha van olyan tartalék fájlrendszered, amit fel tudsz csatlakoztatni (értsd: mount - a ford.), és elfér rajta a gyökérkönyvtár, akkor csinálhatsz valami ehhez hasonlót (nekem a /dev/hda4 a tartalék, és a /dev/hda2 a root partíció):

```
$ mkfs -t ext2 /dev/hda4
$ mkdir /mnt/temp
$ mount -t ext2 /dev/hda4 /mnt/temp
$ cd /
$ tar lcf - . |(cd /mnt/temp; tar xpvf - )
```

Én a "tar"-ral másoltam át a fájlokat, esetleg használhatod a "cp -dpR" parancsot is, de ha a "tar"-t használod -- mint én a példában tettem az "-l" kapcsolóval -- akkor csak az ezen a *partíción* lévő fájlok kerülnek átmásolásra. Pl.: ha van egy másik partíciód, amit a /usr pontra csatlakoztatsz, akkor az azon lévő fájlokat *nem* másolja át (pedig a gyökérkönyvtárból kiindulva ugyebár a /usr könyvtárba is eljuthatunk... - a ford.).

(Figyelem!!! A szöveg további részében a "/dev/hda2" és a "/dev/hda4" partíciók nem biztos, hogy megfelelnek a Te partícióid hivatkozásainak - a ford.) Következő lépésként cseréld ki az /etc/fstab és /etc/lilo.conf állományokban a /dev/hda2 bejegyzéseket /dev/hda4 -re, majd futtasd a "lilo"-t és indítsd újra a rendszert.

A rendszer újraindítása után a /dev/hda4 partíció csatlakozik a gyökérkönyvtár pontjára, az eredeti gyökérkönyvtárat pedig nem csatlakoztatja a rendszer. Most egy új XFS fájlrendszert hozhatsz létre a /dev/hda2 helyén.

```
$ mkfs -t xfs /dev/hda2
```

Csatlakoztasd az új XFS fájlrendszert:

```
$ mount -t xfs /dev/hda2 /mnt/temp
```

És másold vissza a gyökérkönyvtárat az eredeti helyére:

```
$ cd /
$ tar lcf - . |(cd /mnt/temp; tar xpvf -)
```

Cseréld ki a /etc/fstab és /etc/lilo.conf állományokban a /dev/hda4 hivatkozásokat és futtasd a "lilo"-t. A fájlrendszer típusát is módosítanod kell a /dev/fstab állományban. Ennek valahogy így kell kinéznie:

```
/dev/hda2          /                  xfs      defaults 1 1
```

Megjegyzés: Néhány linux disztribúcióban az fstab állományban nem csak alapértelmezett beállítások szerepelnek. Például a Debian rendszerekben használatos a "defaults,errors=remount-ro." A csatlakozás (mount /a ford.) beállításai különbözőek minden fájlrendszerre nézve, kivéve a "default" kulcsszót. Ha nem ismered a specifikus XFS beállításokat, elégedj meg az alapértelmezettel. A Debianban például nem használható az XFS hibakezelési beállítása, és a fájlrendszer nem lesz felcsatlakoztatva.

Ráadásul a fájlrendszer-címkék egyre népszerűbbek, ezért az fstab állományban ilyen bejegyzéseket is láthatsz, mint például ez:

```
LABEL=/            /                  ext2     defaults      1 1
```


A legegyszerűbb módja a fenti probléma megoldásának az, ha kicseréled a címkét a valódi eszköz állománynevére (például, ha a "/dev/hda1"-re címkével hivatkozol / cseréld a "LABEL=/" bejegyzést "/dev/hda1"-re).

Most indítsd újra a rendszered az új XFS partíción lévő gyökérkönyvtárral.

Természetesen számos egyéb lehetőség van a gyökérkönyvtár átmásolására. Ha úgy gondolsz, hogy tudsz egy jobbat, feltétlenül szeretnék hallani róla és berakni ebbe a leírásba, ha egyszerűbbnek néz ki a fent leírt ötletnél. Nekem nincs másik felcsatlakoztatható partícióm, viszont van CD íróm, tehát felírtam a gyökérkönyvtáram CD-re, felcsatlakoztattam, mint gyökérkönyvtárat, miközben készítettem egy új XFS partíciót. Minden esetben ugyanazokat az alapvető parancsokat kell használni egy új fájlrendszer készítésére és csatlakoztatására.

4.2. Utolsó simítások

Az utolsó lépés a partíciók kicserélése. Ez lényegében ugyanaz, mint amit a gyökérkönyvtár átmásolásakor csináltam. (Normális rendszerben van egy partíció a / könyvtárnak, és egy-egy a boot, usr, tmp, root, stb. könyvtáraknak - a ford.) Azt ajánlom, ezt egyfelhasználós üzemmódban hajtsd végre, így a gyökérkönyvtáron kívül bármit lecsatlakoztathatsz, és az egész csereberét milliányi újraindítás nélkül elvégezheted. Egyfelhasználós módba úgy kapcsolhatsz a rendszert, ha kiadsz egy futási szintet átállító parancsot, például így:

```
$ telinit 1
```

vagy újraindításkor a Lilo promptnál kérd ezt:

```
LILO Boot: xfs single
```

Ez elindítja a rendszert, és belépsz egy root parancssorba minden külső kapcsolat és virtuális terminál nélkül, így nem fordulhat elő, hogy valaki, vagy valamilyen program használjon egy fájlrendszert (ez azt eredményezné, hogy az foglalt lenne és nem tudnád leválasztani). Most felcsatlakoztathatod a tartalék fájlrendszered úgy, ahogy előzőleg már leírtuk, és átmásolhatod rá a következő könyvtárat (először talán töröld le róla a gyökérkönyvtárból felmásolt anyagot). Válaszd le a régi fájlrendszert, készíts rajta egy új xfs fájlrendszert, majd csatlakoztasd fel, és másold rá vissza a régi tartalmát stb... Mondjuk ha a /dev/hda3 partíciót a /usr pontra csatlakoztatod fel, akkor a művelet valahogy így néz ki:

```
$ mount -t ext2 /dev/hda4 /mnt/temp
$ cd /usr
$ tar lcf - .|(cd /mnt/temp; tar xpvf - )
$ cd /mnt/temp
$ umount /usr
$ mkfs -t xfs /dev/hda3
$ mount -t xfs /dev/hda3 /usr
$ tar lcf - .|(cd /usr; tar xpvf - )
```

Ne felejtsd el kicserélni a /usr könyvtár fájlrendszerének típusát az /etc/fstab állományban xfs-re.

Ez minden amit tenned kellett. A többi átmásolandó fájlrendszer esetében is ugyanígy kell eljárni. Ezután ha teljes többfelhasználós módban újraindítod a rendszert, megkapod a saját XFS fájlrendszeren futó szteroiddal fűtött Linuxodat!