



FreeBSD – a szomszéd vár (12. rész)

A szomszéd szomszédai

© Kiskapu Kft. Minden jog fenntartva

Néha szükséges, hogy a gépünkre más operációs rendszer is telepítsünk, s itt elsősorban Linux rendszerre gondolok. Valami oknál fogva általában egy gépen a Linux mellé kerül egy FreeBSD, esetleg a FreeBSD mellé egy Linux. Ennek oka vagy az új rendszer megismerése, vagy a hiányzó funkciók kihasználása.

■ Ha már két operációs rendszer van egy számítógépen, akkor ésszerű a lehető legkevesebb eltérést beállítani, amely *FreeBSD* és *Linux* viszonylatában – bizonyos korlátokat betartva – megoldható. Érdemes úgy kialakítani a rendszert, hogy legyen egy közös *home* könyvtár, esetleg a *swap* partíció is legyen közös használatban, s egy ablakkezelő helyes működése sem árt. Ha a legtöbb program képes mind a két rendszer alatt megfelelően működni, akkor már semmi egyéb kívánságunk nem lehet.

A közös fájlrendszer

Fájlrendszer tekintetében ez a két rendszer néhol oly mértékben nem tud együttműködni egymással, hogy ilyet még a *Linux* és *Windows* viszonylatban is ritkán látni. Ez főleg a teljesen szabadon hozzáférhető fájlrendszer specifikációk esetén tűnik furcsának, ugyanis a *FreeBSD* egyedül az *Ext2* fájlrendszert írja-olvassa, az *Ext3* esetén sokszor olvasni sem képes, s gyakorlatilag csak a naplózás egyszerűsítésével képes írni. Ha mégis kezeli, akkor az alábbi beszédes üzenettel képes örületbe kergetni a gyanútlan felhasználót:

```
$ mount_ext2fs /dev/ad0s4 /mnt/
mount_ext2fs: /dev/ad0s4: Invalid argument
```

```
$ dmesg
WARNING: mount of ad0s4 denied due to
↳ unsupported optional features
```

Ez például csak annyit jelent, hogy az adott fájlrendszer nincs rendbe téve, érdemes lenne egy *fsck* futtatása:

```
$ fsck.ext3 /dev/ad0s4
e2fsck 1.38 (30-Jun-2005)
/: recovering journal
/: clean, 120706/555968 files, 765563/1110493
↳ blocks
$ mount_ext2fs /dev/ad0s4 /mnt/
```

Ne bízzuk el magunkat túlságosan, ugyanis hosszabb *FreeBSD* alatti munka után – amikor a *Linux*ot választjuk a következő rendszerindításkor – szinte biztos, hogy elindul és valami hibát talál az *fsck*. Persze ez fordított esetben is így történik, ugyanis a *Linux* sem kezeli rendesen a *FreeBSD* „új” (vagyis lassan három éves) *UFS2* fájlrendszerét, így hosszas munka után itt is azt veszem észre a *FreeBSD* indulásakor, hogy a háttérben elindít egy *fsck* folyamatot is a hibák javítása ügyében. Ráadásul a *SuSE 9.3* „gyári” rendszermagja csak olvasni tudja az *UFS2* fájlrendszert, külön rendszermagot kell fordítani, ha írni is szeretnénk azt. Ezen túlmenően én az *UFS2* fájlrendszer javasolom közös nevezőnek, mert a *Linux* rendszermag *UFS2* kezelő része jobban van megírva (bár tudtommal nem ismeri/kezeli a *SoftUpdate* lehetőséget), mint a *FreeBSD* *Ext3* kezelő modulja.

Ezt még tetézi, hogy ha nem kézzel csatoljuk le a *FreeBSD* leállítás előtt a felcsatolt *Ext2* vagy *Ext3* fájlrendszereket, akkor ez a leálláskor problémát okoz. A *FreeBSD* rendszermagja ugyanis hibásan kezeli az említett fájlrendszerek gyorstárazását, s úgy érzékeli, hogy nem tudja kiírni

a lemezre az összes módosított blokkot. Ennek következménye az, hogy egyik fájlrendszert sem zárja le helyesen, és lecsatolás nélkül eldobja őket, s ez okból a következő indításkor egy *fsck* vár minket néhány teljesen hibátlan fájlrendszeren. Bosszantó és zavaró hiba, bár problémát vagy adatvesztést nem okoz. Kissé érdekes, hogy a világhálón található egyetlen *Linux+FreeBSD HowTo*-ban mindössze a következő sor található ezzel a – még az 5.4-es *FreeBSD*-ben is megtalálható – hibával kapcsolatban: „*The bug has been fixed in FreeBSD 3.x.*”

További hiányosság, hogy az újabb *Linux* kiadások már *UTF-8* fájlrendszert használnak, míg a *FreeBSD* ezt nem képes használni, nincs felkészítve erre a feladatra. Érdekes a *Linux* rendszerben beállítani például az *ISO8859-2* kódlapot a fájlrendszerre is. Érdekes tehát egy *UFS2* fájlrendszerre létrehozni a */home* könyvtárat, amelyet többnyire mind a két rendszer jól kezel.

Telepítési sorrend

Gyakorlatilag teljesen mindegy, hogy melyik rendszer telepítjük fel először, majd ezek után melyikkel folytatjuk. Javasolom azonban először a *FreeBSD*-t feltelepíteni, mivel ennek az igényei a kevésbé rugalmasak, hiszen csak elsőleges partícióra tud feltelepülni, illetve indulni. A *Linux* rendszer már rugalmasabb lehet, főleg egy jól bejártott *GRUB* használatával még a két rendszer indítása is sokkal szebb lehet.

A *GRUB* beállítása egyszerű, a *FreeBSD* teljesen azonos módon indul, mint egy *Windows*, vagyis a megfelelő állományba (*SuSE* esetén a */boot/grub/menu.lst*) kell a következő sorokat beszúrni (megfelelő adatokkal):

```
title FreeBSD
  root (hd0,2)
  chainloader +1
```

A *LILO* is teljesen hasonlóképpen állítható be:

```
other=/dev/hda3
  table=/dev/hda
  label=FreeBSD
```

Közös swap partíció

A két rendszer képes – helyel-közzel közös – *swap* partíciót használni, ha *Linux* alatt minden induláskor lefuttatjuk az *mkswap* parancsot a megfelelő fájlrendszerre. A *FreeBSD* ezt megteszi a *swap* aktiválásakor (nincs is *mkswap* parancsa). Ha *Linux* alatt használjuk a „hibernálást”, akkor a közös *swap* használatot nyugodtan elfelejthetjük, mivel a *Linux* rendszer ide menti le a memória tartalmát, mielőtt lekapcsolná a gépet: a *FreeBSD* pedig minden indulásakor lazán – különösebb üzenet és figyelmeztetés nélkül – felülírja ezeket az adatokat.

Felhasználók

A *passwd* és *shadow* (*FreeBSD* esetén a *master.passwd*) állományok különbözősége okán a felhasználókat külön-külön fel kell vennünk, lehetőleg azonos jelszóval és azonos *UID/GID* számokkal. Érdekes a megnevezéseket is közös nevezőre hozni, mivel a *FreeBSD sysinstall* mindenese

a kitöltetlen csoport esetén létrehoz egy felhasználói névvel azonos nevű csoportot. Ha a */home* közös (és miért ne lenne az), akkor az egyik rendszer problémázni fog a már létező felhasználói könyvtár miatt. Érdekes először *Linux* alatt létrehozni a felhasználókat, mivel itt több állomány kerül be a felhasználó könyvtárába kezdésképp, amelynek nagy részére szükség is van. A *FreeBSD* már képes ezeket megfelelően használni, vagy figyelmen kívül hagyni.

Grafikus felület, KDE

A két rendszer grafikus felületét külön-külön beállíthatjuk, hiszen az alaprendszer külön helyen található. Igyekezzünk azonos felbontást és tulajdonságokat beállítani, különben azonos programokat használva a két rendszer esetén apró kis bosszantó hibák kerülnek napvilágra (kilógó ablakok, eltűnő ikonok, stb.).

A *KDE* (és *GNOME*) esetén különösen ügyeljünk arra, hogy a két operációs rendszer alá feltelepített programok azonos verziószámmal bírjanak, különben képesek apró hibákat okozni a konfigurációs állományokban, amelyeket kibogozni – ha nem is lehetetlen – de időrabló és hosszadalmas feladat (velem többször is előfordult, hogy inkább letöröltem és újra beállítottam az adott programot).

Sajnos a *Linux* terjesztők saját fejlesztései kismértékű problémát okozhatnak, ha belefutunk egy-egy ilyen programba. Például a modemes tárcsázáshoz a *SuSE* egy saját *Kinternet* nevű programot használ, amelyet nehézkes lenne *FreeBSD* alatt használni, így marad a *Kppp*, amelyet viszont *SuSE* alatt nehéz jól beállítani (egyszerűbb átmásolni a *FreeBSD* alatti beállítófájlokat). Érdekes *FreeBSD* alatt kezdeni a beállításokat, mert nagyobb az esély arra, hogy *Linux* alatt is lesz ilyen program, mint fordítva (bár akadnak kivételek). Ezzel véget ért a *FreeBSD* operációs rendszer néhol részletesre, néhol szűkre sikerült bemutatása. Sok sikert és örömet kívánok mindenkinek a szomszéd vár további felfedezéséhez.



Auth Gábor (auth.gabor@enaplo.hu)

Egy pécsi középiskolában informatikát és programozást oktat. Tíz éve botlott először a UNIX rendszerekbe, 7 év *Linux* használat után kapta el a *FreeBSD* látzat, amiből máig nem tudott kigyógyulni.

KAPCSOLÓDÓ CÍMEK

A *FreeBSD* projekt honlapja: ➔ <http://www.freebsd.org>

A magyar *FreeBSD* honlap: ➔ <http://www.freebsd.hu>

A magyar *BSD* honlap: ➔ <http://www.bsd.hu>

A kézikönyv magyar fordítása

➔ <http://www.enaplo.hu/FreeBSD/handbook/>