

## FreeBSD – a szomszéd vár (2. rész)

Egy FreeBSD rendszer adminisztrációja és finomhangolása.

**A**zt hiszem egyet tudunk érteni, hogy két UNIX rendszer használata között nincs lényeges különbség. Azonos programokat tudunk azonos módon használni: lényeges különbségek leginkább a rendszergazda szemszögéből jelentkeznek.

### Rendszerindítás

A *FreeBSD* kicsit másképp indul mint egy *Linux* rendszer, ugyanis csak egy egyszerűbb rendszerbetöltő modullal rendelkezik, amelynek alapvetően nem feladata más operációs rendszerek elindítása. Ez annyit jelent, hogy csak a *FreeBSD*-t, illetve a *DOS*-t és a hozzá hasonlóan induló a *Windows* operációs rendszereket képes elindítani. Ezeknél a rendszereknél ugyanis egy úgynevezett boot szektort kell beolvasni, a rendszer elindítását már ez a kis program végzi. A jelenkori *Linux* rendszerek viszont ennél már egy kicsit összetettebb indítást igényelnek, lévén pontosan meg kell határozni a gyökér-fájlrendszer elérését, esetleg egy *initrd* elindítása is szükséges lehet. Ez utóbbi feladatokat a *FreeBSD* rendszerindítója nem képes felvállalni. Ha megfordítjuk a helyzetet, akkor egy megfelelően telepített *LILO* vagy *GRUB* képes a *FreeBSD*-t elindítani: érdemes ezt a feladatot egy *Linux* rendszerre bízni.

### A nulladik fázis – MBR program

Ha telepítettük a *BootMgr* programot, akkor a gépünket bekapcsolva a következő kép fogad bennünket:

```
F1 FreeBSD
F2 DOS
F4 ??

Default: F1
```

A *BootMgr* az adott partíció szerkezetének megfelelő szöveget írja ki, ahol rendre a merevlemez partícióit láthatjuk. Amelyik partíció nem aktív, de létezik, ott kettő kérdőjelet láthatunk név helyett.

A logikai partíciók nem látszanak, helyettük csak a kiterjesztett partíció látható (szintén kettő kérdőjellel). Az alapértelmezett mindig az utoljára használt operációs rendszer lesz, bizonyos idő elteltével ez el is fog indulni. A kívánt operációs rendszert a megnevezett funkcióbillentyűvel tudjuk kiválasztani. A *BootMgr* program másolata a */boot/boot0* állományban található.

### Az első fázis – Boot program

A *BootMgr* (vagy egyéb hasonló) program betölti a partíció elején található úgynevezett boot szektort, amely a *FreeBSD* indításának első fázisát tartalmazza. Sok tevékenységet nem lehet belesűriteni, hiszen ennek a szektornak a mérete csak 512 Bájit (*/boot/boot1* fájl másolata); egyedüli dolga a beledrótolt címről a második fázis programját beolvasni.

### Második fázis – Előtöltő program

Az előtöltő program (*BTX loader*) veszi át a stafétabotot, amely közel 8 kilobájit méretű (*/boot/boot2* állomány). Ez már elegendő méret arra, hogy a *FreeBSD* által használt *UFS* (illetve *UFS2*) fájlrendszert alapszinten kezelni tudja. Ki tudjuk választani a betöltendő rendszermagot és a betöltő (*loader*) programot. Ezek általában a */boot/kernel* és a */boot/loader* állományban találhatók. Az előtöltő beolvasa a betöltő programot, majd elindítja azt.

### Harmadik fázis – Betöltő program

A */boot/loader* felel a rendszermag és a hozzá tartozó modulok betöltéséért. Ha saját rendszert fordítunk, ez a betöltő program együtt készül el az új rendszermaggal. Tartalmaz egy egyszerű parancsfelületet, ahol a főbb paramétereket módosítani tudjuk. A betöltő program feldolgoz néhány szöveges állományt is, amelyekkel a betöltést befolyásolni tudjuk. Ilyenek a */boot/loader.rc*, a */boot/defaults/loader.conf* és a */boot/loader.conf*. Ezekből a */boot/loader.conf* állományt érdemes megnézni, ahol a betöltendő rendszermagmodulokat tudjuk megadni:

```
ng_ubt_load="YES"
linux_load="YES"
nvidia_load="YES"
```

### A rendszermag

Az operációs rendszerek lelke az a mag, amely feladata tömören a gépben található hardverek, erőforrások és perifériák kezelése, illetve a programok biztonságos futtatása. A *FreeBSD* rendszermag fényes fehér szöveget ír ki a megtalált eszközökről, illetve azok nevééről; s ez a szöveg akár két-három képernyőnyi is lehet.

### A rendszermag beállításai

A telepítéssel foglalkozó részben kihagytam a rendszermag beállításait, mivel egy általánosságban munkaállomásnak

használt telepítést nem befolyásol érdemlegesen. Ebben a részben viszont témába vág, s szeretném itt megemlíteni ezt a fontos műveletet.

Legegyszerűbb (de nem egyedüli!) módja ennek a beállításnak, hogy a feltelepített rendszert újratelepítjük (zárójelben megjegyzem, hogy a FreeBSD működését jobban elsajátítva három-öt hét múlva egy újratelepítéssel sok előnyhöz juthatunk: másképp nézünk a fájlrendszer szerkezetére; esetleg más programokat használunk, mint először gondoltuk; a feltett komponenseket közül esetleg mást választunk ki). A rendszermag beállításának lényege, hogy feloldjuk az ütközéseket az egymással konfliktusba kerülő eszközmeghajtók között. Ehhez a tevékenységhez *nagyon* ismerni kell a gépünkben található hardvereszközöket és ezek pontos működési paramétereit (*DMA*, *IO* és *IRQ* számok). A rendszermag ugyan jó „hatásfokkal” képes a gépünkben található eszközöket helyesen beállítani, viszont előfordulhatnak olyan hibák, amelyekhez kevés lesz a tudása. Például a rendszermag olyan eszközöket is találni vélhet, amelyek nincsenek; olyan meghajtókat esetleg nem tudhat beállítani, amelyek által kezelt hardverek viszont vannak a gépünkben. Ekkor le kell tiltanunk a nem használandó eszközmeghajtókat, majd engedélyeznünk kell azokat, amelyeket ellenben használnánk. Egyre kevésbé előforduló probléma a *PnP* tudást mellőző (*EISA*) kártyák beállítása, amelyeknél pontosan meg kell adnunk a rendszermagnak a kártya által használt paramétereket.

A rendszermag betöltésekor a betöltő program beolvassa a `/boot/device.hints` állományt, amely a fentieket tartalmazza szöveges formában:

```
hint.fdc.0.at="isa"
hint.fdc.0.port="0x3F0"
hint.fdc.0.irq="6"
hint.fdc.0.drq="2"
```

Ennek az állománynak a szerkezete módosulhat a rendszermag fordításakor, lévén ha az egyes eszközmeghajtókat kihagyjuk vagy beletesszük az új rendszermagba, akkor értelemszerűen ezekhez tartozó `device.hints` sorok megszűnnek vagy újak keletkeznek.

A rendszermag betöltése után a `/sbin/init` program veszi át a vezérlést, amely elvégzi az operációs rendszer alapvető működéséhez szükséges programok betöltését és kezelését. Ekkor a kiírt a szöveg színe már világoszürke lesz.

## Az alaprendszer és a telepített programok beállításai

A legtöbb beállítási igényünket a `/etc/rc.conf` állományban tehetjük a *FreeBSD* számára nyilvánvalóvá. Üres vagy majdnem üres `/etc/rc.conf` állomány esetén sem kell aggódniunk, a legtöbb (alapértelmezett) beállítást a `/etc/defaults/rc.conf` fájlban megtaláljuk, amelyeket a `/etc/rc.conf` beállításai értelemszerűen felülbírálnak. A `sysinstall` program szintén ebbe az állományba írja az általunk választott beállításokat, azonban ezek a beállítások csak egy újraindítás után lépnek életbe. Természetesen nem kell minden egyes változtatás után újraindítani az operációs rendszert, de különösebb tudás nélkül ez a leggyorsabb mód.

Sok egyéb lehetőségünk is van a `/etc/` könyvtárban, ha az alaprendszer különféle komponenseit be szeretnénk állí-

tani (*ISDN*, *Bluetooth*, *PPP*, stb). Fontos azonban tudatában lenni annak, hogy a telepített programok nem az alaprendszerrel közös könyvtárakba kerülnek, hanem a `/usr/local/` könyvtár alá, ahol elegendő program telepítése után határozottan felismerhető lesz egy teljes gyökérkönyvtár. Például egy feltelepített *Apache* webkiszolgáló program beállítása a *Linux* esetén megszokott `/etc/apache/httpd.conf` helyett a `/usr/local/etc/apache/httpd.conf` állományt használhatjuk. Ez eleinte kicsit szétszórtnak tűnik, de hamar megtanulható, hogy az alaprendszeren kívüli összes program a `/usr/local/` alatt található meg. Különösen akkor kettős ez az állapot, ha az alaprendszer szerves tartozékát külső programként is feltelepítjük (*GNU C* fordító, *PERL*, *SSH*, *Sendmail*). Segítségét jelenthet, hogy ez a „filozófia” végigkövethető a legtöbb komponens esetén. Például az alaprendszer programjait a `/etc/rc.d/` könyvtár alá tett kis programcskák indítják, a pluszban feltett programokat pedig a `/usr/local/etc/rc.d/` alatti programcskák. Például az *SSH* futásának ellenőrzése a `/etc/rc.d/sshd status` futtatásával leszünk képesek, mivel az *SSH* az alaprendszer része; ellenben az *Apache2* elindítása a `/usr/local/etc/rc.d/apache2.sh start` paranccsal történhet (mivel ez nem az alaprendszer része).

## A sysctl felület

A `sysctl` egy egyszerű program, amellyel a rendszermag futás közbeni beállítását ejthetjük meg. Használata teljesen azonos a *Linux* alatt használatos `sysctl` programmal. Ha néhány paramétert minden indítás után be szeretnénk állítani, akkor ezt a `/etc/sysctl.conf` állományban tehetjük meg:

```
security.bsd.see_other_uids=0
```

## A fájlrendszer

A *FreeBSD* fájlrendszere az *UFS (UFS2)* nevet viseli, mint *Unix File System (UNIX)* fájlrendszer). Némileg eltér a *Linux* alapvető (*Ext2*) fájlrendszerétől, de alapelveiben azonosak. Egy új fogalmat azonban ismerni kell hozzá: ez a *Soft Updates*. Ennek megértéséhez szükséges ismerni a fájlrendszerek néhány tulajdonságát.

A fájlrendszerek által tárolt adatok alapvetően két részre oszthatók: adatokra és metaadatokra. Az adatokkal mindenki találkozott már, hiszen ezekkel dolgozunk. A metaadatok ezekről a tárolt adatokról szolgáltatnak információkat a felhasználók számára is, de főleg az operációs rendszer felé.

A számítógépes „ókorban” minden adatot szinkron írtak a tároló eszközökre, vagyis a másolás azonnal megtörtént, a programok megvárták a kiírás végét. Ez *PC* esetén egészen a *DOS* operációs rendszer néhány utolsó verziójáig is így történt. Fel sem merült akkoriban, hogy a drága és kevés operatív tárat adatok átmeneti tárolására használják fel. Idővel azonban a memória olcsóbb lett, ezért kifizetődőbb volt az adatok egy részét a sokkal gyorsabb memóriában tartani, mint azonnal kiírni a lemezre. Az adatok akkor kerültek ki a lemezre, amikor arra ideje volt az operációs rendszernek, illetve amikor ez a gyorsítótár megtelt.

A metaadatok általában azonnal kiíródtak a lemezre, mivel ezzel egyszerűbb lehetett a fájlrendszert kezelő modul.

A *Soft Updates* annyit javított ezen a megoldáson, hogy a metaadatok is gyorsítótárba kerültek, így sok apró állomány másolása vagy törlése sokkal gyorsabb lehetett.

Hátrányaként a sérülékenyebb fájlrendszert lehet megemlíteni. Sajnos a *FreeBSD* naplózós fájlrendszert jelenleg még nem támogat rendszerszinten, azonban több éve hallani pletykákat ilyen fájlrendszer alkalmazásáról. Reménykedni tudunk csak, hogy ez a közeljövőben valósággá válik, addig is kerülő megoldásként az *5.x* sorozat már képes váratlan újraindulás (áramszünet vagy rendszerhiba) után háttérben lefuttatni a fájlrendszer ellenőrzését.

## A sysinstall program

Mint már említettem, ez a program a *FreeBSD* mindenese, amely helyileg a */usr/sbin/* könyvtárban található. Használatához alapvető angol tudás szükséges, mivel magyarosítása egyelőre még tervben sem szerepel.

A telepítés során kénytelenek voltunk megismerkedni a használatával, meglepetés lehet azonban, hogy a feltelepített rendszer esetén is azonos menürendszert láthatunk, amelyből az első utáni három menüpont a telepítést indítaná el. Legyünk mindig körültekintőek a használata során, bár egy újabb telepítést nehéz akaratlanul véghezvinni, de jobb a békesség.

A telepített rendszer beállításához igazából egyetlen menüpontot használhatunk a főmenüből, s ez a „*Configure*” nevet viseli. Ebbe beelépve a leggyakoribb alaprendszert illető beállításokat meg tudjuk ejteni.

## Új alaprendszer letöltése

A *FreeBSD* egyik jó tulajdonsága, hogy az alaprendszer folyamatosan változik, s ezt a változást könnyedén követhetjük a telepített rendszerünk esetén. Ehhez egy kicsit bele kell mélyedni a *CVS* működésébe, amely egy egyszerű elvet valószínűleg sok fejlesztő közös munkáját hangolja össze. Maga a *CVS* használata nem egyszerű tevékenység, viszont remek programok készültek e tevékenység egyszerűsítésére. Mivel a *FreeBSD* használói között elenyésző az alaprendszert fejlesztők száma, ezért a készítőknél egyszerű és könnyen használható eszközt kellett adni a rendszergazdák kezébe. Ez az eszköz *cvsup*, amely egyetlen feladata, hogy a *FreeBSD* *CVS* kiszolgálójáról (vagy annak tükörszervereiről) letöltse a legújabb alaprendszer és rendszermag forrását. Erre egy egyszerű *FTP* kapcsolat is elég lenne, viszont a teljes forrás 400-450 megabájt méretet jelent. Ezért aztán a *cvsup* képes arra, hogy kiderítse a különbségeket és csak azzal terhelje a hálózatot. Ezáltal alig néhány száz kilobájt forgalmat okoz heti rendszerességgel történő ellenőrzéseket alapul véve. A legelső probléma az szokott lenni, hogy nincs feltelepítve a *cvsup* program (noha szerintem lehetne az alaprendszer része is :), ezért fel kell telepíteni. Ennek egyik módja a *sysinstall* programot használva a „*Configure*”, majd a „*Packages*” menüpontba lépve a telepítési médium kiválasztása után megkeressük és feltelepítjük a *cvsup* programot. Másik módja, hogy parancssorban kiadjuk a *pkg\_add -r -v cvsup* parancsot, amely megpróbálja letölteni és feltelepíteni. Igazából a *sysinstall* is a *pkg\_add* programot fogja meghívni, de mégis egyszerűbbnek látszik a használata. A programok telepítéséről azonban a következő részben sokkal részletesebben foglalkozok. A következő probléma a *cvsup* helyes és jól működő beállítása. Ehhez egy úgynevezett *sup* állományt használunk, amely például így nézhet ki (az állomány neve lehet például *src-sup*):

```
*default host=cvsup.hu.freebsd.org
*default base=/usr
*default prefix=/usr
*default release=cvsup tag=RELENG_5
*default delete use-rel-suffix
*default compress
```

```
src-all
```

Itt megadjuk a használandó kiszolgáló nevét, néhány fontos és kevésbé fontos paramétert, illetve a használandó ágat. Ez utóbbi több szót érdemel, ennek hibás beállításával hamar megszabadulhatunk a teljes */usr/src/* tartalomtól is, ugyanis nem létező tag-et használva a *cvsup* egyszerűen letörli az eddig használt forrásokat. Természetesen használhatjuk például a *RELENG\_5\_2* tag-et is, melynek hatására megmaradunk az *5.2* verziónál, s csak a biztonsági frissítések fognak újdonságot jelenteni. Ez utóbbit akkor érdemes használni, ha olyan programokkal dolgozunk, amelyek egy jól meghatározott *FreeBSD* verzióhoz kötődnek. Javasolom a *RELENG\_5* használatát, amely esetén királyi utunk lesz az *5.x* verziók közötti váltás során, illetve ezzel biztonságosan végig tudjuk követni a stabil *5.x* ágat, minden stabilnak ítélt újdonságról értesülünk a */usr/src/UPDATING* állományt böngészve. A források letöltését a *cvsup src-sup* parancs kiadásával indíthatjuk el, a program folyamatosan információkkal szolgál az éppen módosított állományokról.

Az alaprendszer nem igényel különösebb beállítást, mivel minden ilyen irányú tevékenységet a */etc/* könyvtárban hajthatunk végre, a rendszermag könnyedén a gépünkre szabható. Ehhez meg kell keresnünk a */usr/src/sys/i386/conf/* könyvtárt, benne pedig a *GENERIC* állományt: ez tartalmazza a kernel beállításait. Szerkezete egyszerű, opciókat és eszközöket sorol fel, amelyek közül kivethetjük a szükségtelen eszközöket (például *RAID* és *SCSI* kártyák), illetve hozzávehetünk szükséges opciókat (például *QUOTA*). Célszerű a *GENERIC* állományról egy másolatot készíteni, hogy bármikor visszatérhessünk egy működő beállításhoz. Az új alaprendszer és rendszermag fordításáról illetve telepítéséről a következő részben írok.



**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* lázat, 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/>.