

FreeBSD gyorstalpaló Linux® felhasználók számára

Kivonat

Ez a cikk azért íródott, hogy röviden megismertesse a FreeBSD alapjait a középhaladó-haladó Linux® felhasználókkal.

Fordította: Páli Gábor, utolsó ellenőrzés: 2010.11.28.

Tartalomjegyzék

| | |
|--|---|
| 1. Bevezetés | 1 |
| 2. Parancsértelmezők: hova tűnt a Bash? | 1 |
| 3. Csomagok és portok: szoftverek telepítése FreeBSD alatt | 2 |
| 4. A rendszer indítása: hova lettek a futási szintek? | 4 |
| 5. A hálózat beállítása | 5 |
| 6. Tűzfalak | 6 |
| 7. A FreeBSD frissítése | 7 |
| 8. procfs: eltűnt, de nem nyomtalanul | 8 |
| 9. Gyakori parancsok | 9 |
| 10. Lezárás | 9 |

1. Bevezetés

Ebben a leírásban a FreeBSD és a Linux® közti alapvető eltéréseket igyekszünk szemléltetni, aminek révén a középhaladó és haladó Linux® felhasználók pillanatok alatt bepillantást nyerhetnek a FreeBSD alapjaiba. Ez egyszerűen csak egy szakmai jellegű bevezetés, és nem foglalkozik a két rendszer felépítése közti "filozófiai" különbségekkel.

A leírás feltételezi, hogy korábban már telepítettük a FreeBSD rendszert. Amennyiben ezt még nem tettük volna meg, vagy segítségre lenne szükségünk a telepítésben, akkor olvassuk el a FreeBSD kézikönyv [A FreeBSD telepítése](#) című fejezetét.

2. Parancsértelmezők: hova tűnt a Bash?

A Linuxról áttérő felhasználók gyakran meglepődnek azon, hogy a FreeBSD-ben nem a Bash az alapértelmezett parancsértelmező. Sőt, a Bash még az alaprendszerben sem található meg. Helyette a `tcsh(1)` az alapértelmezett parancsértelmező a FreeBSD-ben. Természetesen a Bash, a többi szintén közkedvelt parancsértelmezőhöz hasonlóan megtalálható a FreeBSD [Csomag- és](#)

Portgyűjteményében.

Ha más parancsértelmezőket is telepítettünk, akkor a `chsh(1)` parancs segítségével tudjuk megváltoztatni az alapértelmezett parancsértelmezőt. A `root` felhasználó alapértelmezett parancsértelmezőjének megváltoztatását azonban nem javasoljuk. Ennek oka, hogy azok a parancsértelmezők, amelyek nem részei az alaprendszernek, általában a `/usr/local/bin` vagy a `/usr/bin` könyvtárakban találhatóak, és bizonyos vészhelyzetekben előfordulhat, hogy ezeket az állományrendszereket nem tudjuk csatlakoztatni. Ilyen esetekben a `root` sem lesz képes elérni a saját alapértelmezett parancsértelmezőjét, amivel lényegében megakadályozzuk, hogy be tudjon jelentkezni. Erre a célra a `root` felhasználó egy alternatíváját, a `toor` felhasználót hozták létre, amelyet az alaprendszeren kívül található parancsértelmezőkkel is használhatunk. A [toor hozzáféréséről](#) a GYIK biztonsági kérdésekkel foglalkozó részében tudhatunk meg többet (angolul).

3. Csomagok és portok: szoftverek telepítése FreeBSD alatt

A szoftverek telepítésének hagyományos UNIX®-os megoldásain (a forrás letöltésén, kitömörítésén, a forráskód módosításán és lefordításán) túl az alkalmazások telepítésének további két módját is felkínálja a FreeBSD: ezek a csomagok és a portok. A rendszerhez elérhető összes port és csomag teljes listáját [ezen](#) a címen érhetjük el.

3.1. Csomagok

A csomagok lényegében előre lefordított alkalmazások, amelyek megfelelnek a Debian/Ubuntu rendszerekben megtalálható `.deb`, vagy a Red Hat/Fedora rendszerekben megtalálható `.rpm` állományoknak. A csomagok a `pkg_add(1)` segítségével telepíthetők. Például az alábbi parancs az Apache 2.2 alkalmazást rakja fel:

```
# pkg_add /tmp/apache-2.2.6_2.tbz
```

Az `-r` kapcsolóval arra utasítjuk a `pkg(add)` programot, hogy magától töltsse le és telepítse a csomagot, valamint annak függőségeit:

```
# pkg_add -r apache22
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6.2-
release/Latest/apache22.tbz... Done.
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6.2-release/All/expat-
2.0.0_1.tbz... Done.
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6.2-release/All/perl-
5.8.8_1.tbz... Done.
[nyissz]
```

```
To run apache www server from startup, add apache22_enable="YES"
in your /etc/rc.conf. Extra options can be found in startup script.
```



Ha a FreeBSD valamelyik kiadását használjuk (6.2, 6.3, 7.0 stb., tehát CD-ről telepítettük), akkor a `pkg_add -r` az adott kiadáshoz tartozó csomagokat fogja letölteni. Ezek a csomagok azonban *nem feltétlenül* az alkalmazás legújabb verziójához tartoznak. Ezt az alapértelmezett viselkedést felül tudjuk bírálni, ha a `PACKAGESITE` környezeti változót az `ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6-stable/Latest/` értékre állítjuk, és így például a 6.X sorozathoz készült legfrissebb csomagokat tölthetjük le.

A FreeBSD különböző változatairól a [Válasszuk ki a nekünk igazán megfelelő FreeBSD verziót!](#) című cikkben olvashatunk bővebben.

A csomagok használatával kapcsolatban a FreeBSD kézikönyvében kaphatunk részletesebb felvilágosítást, lásd [A csomagrendszer használata](#).

3.2. Portok

A FreeBSD-ben az alkalmazások telepítésének másik módja a Portgyűjtemény használata. A Portgyűjtemény lényegében Makefile állományok és javítások gyűjteménye, amelyek a különféle alkalmazások forráskódját készítik fel arra, hogy a FreeBSD-n is használhatóak legyenek. Amikor telepítünk egy portot, akkor a rendszer először letölti az alkalmazás forráskódját, elvégzi a szükséges módosításokat, lefordítja a forrást és végül telepíti az alkalmazást (valamint mindezt megteszi az összes függősége esetében).

A Portgyűjtemény, vagy gyakran egyszerűen csak a "portfa", a `/usr/ports` könyvtárban található. Itt nyilván feltételezzük, hogy a Portgyűjteményt is kiválasztottuk a FreeBSD telepítése során. Amennyiben a Portgyűjteményt még nem telepítettük volna, a `sysinstall(8)` segítségével feltehetjük a telepítőlemezről, vagy esetleg a `csup(1)`, illetve `portsnap(8)` használatával letölthetjük a FreeBSD Projekt valamelyik szerveréről. A Portgyűjtemény telepítésének részletes bemutatása megtalálható a kézikönyv [4.5.1. szakaszában](#).

A telepítéshez (általában) csak be kell lépni az adott port könyvtárába és el kell indítanunk a fordítást. A következő példában az Apache 2.2 alkalmazást telepítjük a Portgyűjteményből:

```
# cd /usr/ports/www/apache22
# make install clean
```

A portok alkalmazásának egyik legnagyobb előnye, hogy a szoftverek telepítése során testre tudjuk szabni azok beállításait. Például amikor az Apache 2.2 alkalmazást portként telepítjük, a `WITH_LDAP` `make(1)` változó megadásával engedélyezhetjük a `mod_ldap` használatát:

```
# cd /usr/ports/www/apache22
# make WITH_LDAP="YES" install clean
```

A Portgyűjteménnyel kapcsolatos további információk tekintetében olvassuk el a FreeBSD kézikönyv [A Portgyűjtemény használata](#) című szakaszát.

3.3. Portok vagy csomagok, mégis melyiket használjam?

A csomagok tulajdonképpen előre lefordított portok, ezért igazából csak abban van köztük különbség, hogy forrásból (portok) vagy binárisan telepítjük-e az alkalmazásokat. Mindegyik módszernek megvannak a maga előnyei:

Csomagok (bináris)

- Gyorsabb telepítés (a nagyobb alkalmazások lefordítása viszont nagyon sokáig is eltarthat).
- Nem szükséges megértenünk a szoftverek lefordításának mikéntjét.
- Nem kell fordítóprogramokat telepítenünk a rendszerünkre.

Portok (forrás)

- A telepítés beállításait tetszőlegesen szabályozhatjuk. (A csomagok általában szabványos beállításokkal készülnek. A portok esetében azonban lehetőségünk van ezeket kedvünk szerint megváltoztatni, mint például további modulok fordítását kérni, vagy átállítani a telepítés alapértelmezett helyét.)
- Ha készítenést érzünk, akkor akár a saját javításainkat is beletehetjük a forráskódba.

Ha nincsenek különös igényeink, akkor a csomagok minden bizonnyal tökéletesen megfelelnek számunkra. Amikor viszont valamit külön be szeretnénk állítani, akkor ahhoz a portokat érdemes választanunk. (Ne felejtjük el azonban, hogy ha elsősorban a csomagokhoz ragaszkodunk, de mégis módosítanunk kell valamit bennük, akkor a `make package` parancs kiadásával a portokból is tudunk csomagot készíteni, majd átmásolni azokat más szerverekre.)

4. A rendszer indítása: hova lettek a futási szintek?

A Linux® a SysV rendszerindítási sémáját alkalmazza, miközben a FreeBSD a hagyományos BSD típusú `init(8)` megoldást. A BSD típusú `init(8)` esetén nincsenek futási szintek és nem létezik `/etc/inittab` állomány. Helyette az `rc(8)` vezérli a rendszer indítását. Az `/etc/rc` szkript beolvassa az `/etc/defaults/rc.conf` és `/etc/rc.conf` állományokat, amelyekből megállapítja, hogy milyen szolgáltatásokat indítson el. A megadott szolgáltatásokat ezután az `/etc/rc.d` és a `/usr/local/etc/rc.d` könyvtárakban található megfelelő indítószkriptek segítségével indítja el. Ezek a szkriptek hasonlóak a Linux® rendszereken az `/etc/init.d` könyvtárban található szkriptekhez.

A szolgáltatások indításáért felelős szkriptek miért két különböző helyen találhatóak? Az `/etc/rc.d` könyvtárban található szkriptek az "alaprendszer" részei (mint például a `cron(8)`, `sshd(8)`, `syslog(3)` és a többi). A `/usr/local/etc/rc.d` könyvtárban pedig a felhasználó által telepíthető alkalmazások, például az Apache, Squid stb. szkriptjei találhatóak.

Mi a különbség az "alaprendszerben" található és a felhasználó által telepített alkalmazások között? A FreeBSD-t egy összefüggő operációs rendszerként fejlesztik. Ezt másképpen úgy lehetne fogalmazni, hogy a rendszermagot, a rendszerszintű függvénykönyvtárakat és a

hozzájuk tartozó programokat (mint például a `ls(1)`, `cat(1)`, `cp(1)` stb.) együtt fejlesztik és adják ki. Ezt nevezzük az "alaprendszernek". A felhasználó által telepíthető alkalmazások lényegében azok, amelyek nem részei ennek az "alaprendszernek", például az Apache, X11, Mozilla Firefox stb. Ezek általában a FreeBSD [Csomag- és Portgyűjteményéből](#) telepíthetőek. Mivel a felhasználók által telepített alkalmazásokat igyekszünk elkülöníteni az "alaprendszerrel", ezért ezek a `/usr/local/` könyvtárba kerülnek. Ennek következtében a felhasználók által telepített binárisok a `/usr/local/bin` könyvtárban, míg a hozzájuk tartozó konfigurációs állományok a `/usr/local/etc` könyvtárban találhatóak, és így tovább.

A szolgáltatásokat az `/etc/rc.conf` állományban (lásd [rc.conf\(5\)](#)) tudjuk engedélyezni a `SzolgáltatásNév_enable="YES"` sor megadásával. A rendszer alapértelmezett beállításait az `/etc/defaults/rc.conf` állományban találhatjuk meg, ezeket az `/etc/rc.conf` állományban tudjuk felülbírálni. Az alkalmazásokhoz tartozó szolgáltatások engedélyezésének lépéseire pedig a telepítésük után ne felejtsük el átolvasni a hozzájuk tartozó dokumentációt.

Az `/etc/rc.conf` állományból származó most következő rövid kódrészlet az [sshd\(8\)](#) és Apache 2.2 szolgáltatásokat engedélyezi, valamint az Apache számára beállítja az SSL használatát.

```
# az SSHD engedélyezése
sshd_enable="YES"
# az Apache és benne az SSL támogatásának engedélyezése
apache22_enable="YES"
apache22_flags="-DSSL"
```

Miután az `/etc/rc.conf` állományban engedélyeztük a szolgáltatásokat, a parancssorból el is tudjuk indítani ezeket (a rendszer újraindítása nélkül):

```
# /etc/rc.d/sshd start
```

Ha egy szolgáltatást nem engedélyeztünk, akkor a parancssorból a `forrestart` paraméter megadásával tudjuk elindítani:

```
# /etc/rc.d/sshd forrestart
```

5. A hálózat beállítása

5.1. Hálózati interfészek

A hálózati csatolófelületekre a Linux esetén alkalmazott általános `ethX` alakú azonosítók helyett a FreeBSD az adott hálózati kártya meghajtójának nevével és utána egy sorszámmal hivatkozik. Az [ifconfig\(8\)](#) itt látható kimenetében két Intel® Pro 1000 hálózati kártya jelenik meg (`em0` és `em1`):

```
% ifconfig
```

```
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=b<RXCSUM,TXCSUM,VLAN_MTU>
    inet 10.10.10.100 netmask 0xffffffff broadcast 10.10.10.255
    ether 00:50:56:a7:70:b2
    media: Ethernet autoselect (1000baseTX <full-duplex>)
    status: active
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=b<RXCSUM,TXCSUM,VLAN_MTU>
    inet 192.168.10.222 netmask 0xffffffff broadcast 192.168.10.255
    ether 00:50:56:a7:03:2b
    media: Ethernet autoselect (1000baseTX <full-duplex>)
    status: active
```

5.2. Az IP-cím beállítása

Az interfészekhez az `ifconfig(8)` paranccsal tudunk IP-címet rendelni. Az IP-címek beállítása azonban csak akkor marad meg az újraindítást követően is, ha felvesszük az `/etc/rc.conf` állományba. A most következő példában megadunk egy hálózati nevet, IP-címet és egy alapértelmezett átjárót:

```
hostname="szerver1.minta.com"
ifconfig_em0="inet 10.10.10.100 netmask 255.255.255.0"
defaultrouter="10.10.10.1"
```

DHCP esetén használjuk a következőt:

```
hostname="szerver1.minta.com"
ifconfig_em0="DHCP"
```

6. Tűzfalak

Hasonlóan a Linuxban található IPTABLES megoldáshoz, a FreeBSD is kínál fel rendszermagszintű tűzfalazást. A FreeBSD jelen pillanatban három tűzfalat támogat:

- [IPFIREWALL](#)
- [IPFILTER](#)
- [PF](#)

Az IPFIREWALL, avagy IPFW (az IPFW szabályrendszerit az `ipfw(8)` paranccsal tudjuk kezelni) a FreeBSD fejlesztői által készített és karbantartott tűzfal. A forgalomszabályozás megvalósításához és különböző típusú hálózati kapcsolatok szimulációjához az IPFW kiegészíthető a [dummynet\(4\)](#) használatával.

Ez az IPFW szabály engedélyezi a beérkező SSH-kapcsolatokat:

```
ipfw add allow tcp from any to me 22 in via $ext_if
```

Az IPFILTER tűzfalat Darren Reed dolgozta ki. Nem csak FreeBSD alatt találkozhatunk vele, több operációs rendszerre is portolták, többek közt NetBSD-re, OpenBSD-re, SunOS-re, HP/UX-ra és Solarisra.

Ez az IPFILTER parancs engedélyezi a beérkező SSH-kapcsolatokat:

```
pass in on $ext_if proto tcp from any to any port = 22
```

Az utolsó tűzfal, a PF, az OpenBSD Projekt fejlesztése. A PF eredetileg az IPFILTER leváltására készült. Emiatt a PF szabályainak megadási módja nagyon hasonlít az IPFILTER esetében megismertekhez. A minőség alapú (QoS) forgalom szabályozás létrehozásához a PF az [altq\(4\)](#) megoldásával egészíthető ki.

Ez a PF parancs engedélyezi a beérkező SSH-kapcsolatokat:

```
pass in on $ext_if inet proto tcp from any to ($ext_if) port 22
```

7. A FreeBSD frissítése

A FreeBSD rendszer háromféleképpen frissíthető: forráskódból, binárisan és telepítőlemezek használatával.

A forráskódon keresztüli frissítés ugyan a legbonyolultabb ezek közül, azonban ez kínálja fel egyben a legnagyobb rugalmasságot is. Ennek során szinkronizálnunk kell a FreeBSD forráskódjának nálunk levő (helyi) másolatát a FreeBSD CVS (Concurrent Versioning System) szervereivel. Miután ez megtörtént, le tudjuk fordítani a rendszermagot és a hozzá tartozó programokat. A források frissítésével kapcsolatban olvassuk el a FreeBSD kézikönyv [frissítésről szóló fejezetét](#).

A bináris frissítés a Linux® típusú rendszereken elérhető `yum` vagy `apt-get` parancsok esetén megszokottakhoz hasonló. A `freebsd-update(8)` parancs letölti a frissítéseket és telepíti ezeket. Ez a frissítési folyamat a `cron(8)` használatával ütemezhető.



Amikor a `cron(8)` segítségével ütemezzük a frissítéseket, a `crontab(1)` állományban lehetőség szerint a `freebsd-update cron` parancsot használjuk, ezáltal igyekezzünk csökkenteni annak valószínűségét, hogy egyszerre több számítógép is ugyanakkor terhelje a szerveret.

```
0 3 * * * root /usr/sbin/freebsd-update cron
```

Az utolsó frissítési módszer, a telepítőlemezek használata lényegében egy egyértelmű folyamat. Indítsuk el számítógépünket a telepítőlemezzel, és a telepítőben válasszuk a frissítés (upgrade)

opciót.

8. procfs: eltűnt, de nem nyomtalanul

A Linux® alatt a `/proc/sys/net/ipv4/ip_forward` használatával tudjuk megmondani, hogy az IP-csomagok továbbítása engedélyezett-e rendszerünkben. Mivel a [procfs\(5\)](#) a FreeBSD jelenlegi verzióiban már elavultnak számít, ezért ezt a [sysctl\(8\)](#) paranccsal nézhetjük meg a rendszer egyéb beállításai mellett. (A `sysctl` viszont Linux® alatt is egyaránt megtalálható.)

Ha az IP-csomagok továbbításáról szóló példánál maradunk, akkor az alábbi módon kérdezhetjük le, hogy engedélyezett-e a FreeBSD rendszerünkön:

```
% sysctl net.inet.ip.forwarding
net.inet.ip.forwarding: 0
```

Az `-a` paraméter megadásával a rendszer összes jelenlegi beállítását le tudjuk kérdezni:

```
% sysctl -a
kern.ostype: FreeBSD
kern.osrelease: 6.2-RELEASE-p9
kern.osrevision: 199506
kern.version: FreeBSD 6.2-RELEASE-p9 0: Thu Nov 29 04:07:33 UTC 2007
    root@i386-builder.daemonology.net:/usr/obj/usr/src/sys/GENERIC

kern.maxvnodes: 17517
kern.maxproc: 1988
kern.maxfiles: 3976
kern.argmax: 262144
kern.securelevel: -1
kern.hostname: server1
kern.hostid: 0
kern.clockrate: { hz = 1000, tick = 1000, profhz = 666, stathz = 133 }
kern.posix1version: 200112
...
```



Bizonyos `sysctl`-értékek írásvédettek.

Adódhatnak olyan alkalmak, amikor mégis szükségünk lehet a procfs használatára, mint például régi szoftverek futtatása, a rendszerhívások nyomkövetése a [truss\(1\)](#) segítségével, vagy a [bináris Linux kompatibilitás](#) használata. (Noha a bináris Linux kompatibilitás egy saját procfs állományrendszert, egy [linprocfs\(5\)](#) rendszert használ.) A procfs típusú állományrendszerek csatlakoztatásához a következőt kell megadnunk az `/etc/fstab` állományban:

```
proc                /proc              procfs rw,noauto      0      0
```




A `noauto` beállítás megadásával megakadályozzuk, hogy a `/proc` a rendszerindítás során magától csatlakoztatódjon.

A `procfs` típusú állományrendszereket így lehet csatlakoztatni:

```
# mount /proc
```

9. Gyakori parancsok

9.1. A csomagok kezelése

| Linuxos parancs (Red Hat/Debian) | A FreeBSD-s megfelelője | Leírás |
|---|--------------------------------|---|
| <code>yum install csomag</code> / <code>apt-get install csomag</code> | <code>pkg_add -r csomag</code> | A <i>csomag</i> telepítése egy távoli számítógépről |
| <code>rpm -ivh csomag</code> / <code>dpkg -i csomag</code> | <code>pkg_add -v csomag</code> | Csomag telepítése |
| <code>rpm -qa</code> / <code>dpkg -l</code> | <code>pkg_info</code> | A telepített csomagok megjelenítése |

9.2. A rendszer kezelése

| Linuxos parancs | A FreeBSD-s megfelelője | Leírás |
|-----------------------|---|--|
| <code>lspci</code> | <code>pciconf</code> | A PCI-os eszközök megjelenítése |
| <code>lsmod</code> | <code>kldstat</code> | A betöltött rendszermagmodulok felsorolása |
| <code>modprobe</code> | <code>kldload</code> / <code>kldunload</code> | Modulok betöltése és eltávolítása |
| <code>strace</code> | <code>truss</code> | A rendszerhívások nyomkövetése |

10. Lezárás

Bízunk benne, hogy ez a leírás eleget mutatott be ahhoz, hogy elkezdjünk ismerkedni a FreeBSD-vel. Ha az érintett témák még jobban érdekelnek minket, vagy olyanról szeretnénk többet megtudni, ami itt nem szerepelt, akkor mindenképpen olvassunk bele a [FreeBSD kézikönyvbe](#).