

FreeBSD gyorstalpal– Linux[®] felhasználók számára

Kivonat

Ez a cikk azért íródott, hogy röviden megismertesse a FreeBSD alapjait a széleskörűen használt Linux[®] felhasználóknak.

Fordította: Páli Gábor, utolsó ellenőrzés: 2010.11.28.

Tartalomjegyzék

1. Bevezetés	1
2. Parancsértelmező k: 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 hálózati beállítás	5
6. Tűzfal	6
7. A FreeBSD frissítése	7
8. procfs: elmentés, de nem nyomtalanul	8
9. Gyakori parancsok	9
10. Lezárás	9

1. Bevezetés

Ebben a leírásban a FreeBSD és a Linux[®] közötti alapvető eltéréseket igyekezzünk szemléltetni, aminek révén a széleskörűen használt és haladékos Linux[®] felhasználók pillanatok alatt bepillantást nyerhetnek a FreeBSD alapjaiba. Ez egyszerűen csak egy szakmai jellegű bevezetés, és nem foglalkozik a konkrét rendszer felépítésével és a "filozófiai" különbségekkel.

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

2. Parancsértelmező k: hova tűnt a Bash?

A Linux[®]-l ellentétben a 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 széleskörűen használt parancsértelmezőhöz hasonlóan megtalálható a FreeBSD [Csomag- és](#)

Portgyűjteményben.

Ha más parancsértelmezőket is telepítettünk, akkor a `chsh(1)` parancs segítségével tudjuk megváltoztatni az alapértelmezett parancsértelmezőket. A `root` felhasználó alapértelmezett parancsértelmezőinek megváltoztatásán 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 vizsgálathelyzetekben előfordulhat, hogy ezeket az állományrendszereket nem tudjuk csatlakoztatni. Ilyen esetekben a `root` sem lesz képes elérni a saját alapértelmezett parancsértelmezőt, amivel könnyebben megakadályozzuk, hogy be tudjon jelentkezni. Erre a célra a `root` felhasználó egy alternatívját, a `toor` felhasználót hoztuk létre, amelyet az alaprendszeren kívül található parancsértelmezőkkel is használhatunk. A [toor hozzáférési útmutató](#) a GYIK biztonsági kérdésekkel foglalkozó részben tudhatunk meg többet (angolul).

3. Csomagok és portok: szoftverek telepítése FreeBSD alatt

A szoftverek telepítésének hagyományos UNIX-os megoldásain (a forrás letöltésén, kitöltésén, a forráskódot más-dosításán 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 könnyebben 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 magát a leírt 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 feltöltényl* az alkalmazás legújabb verziójához tartoznak. Ezt az alapértelmezett viselkedést felül tudjuk bírni, 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 kérésre változatairól a [Váltszuk ki a nekünk igazán megfelel FreeBSD verziót!](#) című cikkben olvashatunk bővebben.

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

3.2. Portok

A FreeBSD-ben az alkalmazások telepítésnek mindig megdja a Portgyűjtemény használata. A Portgyűjtemény lényegében Makefile állományok és javítások gyűjteménye, amelyek a kérésre alkalmazások forráskódját készítik fel arra, hogy a FreeBSD-n is használhatók legyenek. Amikor telepítünk egy portot, akkor a rendszer elszőr letölti az alkalmazás forráskódját, elvégzi a szükséges módosításokat, lefordítja a forrást és végül telepíti az alkalmazást (valamint mindezt megteszi az összes függő esetben).

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

A telepítéshez (általában) csak be kell lépni az adott port könyvtárba 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 elnye, hogy a szoftverek telepítésre soron 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 tekintetben olvassuk el a FreeBSD kézikönyv [A Portgyűjtemény használata](#) című szakaszát.

3.3. Portok vagy csomagok, 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 megírtenünk a szoftverek lefordításának mikéntjét.
- ☞ Nem kell fordítóprogramokat telepítenünk a rendszerünkre.

Portok (forrás)

- ☞ A telepítés beállításeit tetszőlegesen szabályozhatjuk. (A csomagok általában szabványos beállítésekkel készülnek. A portok esetében azonban lehetőségek vannak ezeket kedvünk szerint megváltoztatni, mint például további modulok fordítását kérni, vagy állítani a telepítés alapértelmezett helyét.)
- ☞ Ha készített szűrőnk, akkor akár a saját javításainkat is beletehetjük a forrásokba.

Ha nincsenek különleges igényeink, akkor a csomagok minden bizonnyal tökéletesen megfelelnek számunkra. Amikor viszont valamit külön be szeretnénk állítani, akkor ahhoz a portokat írddemes választanunk. (Ne felejtjük el azonban, hogy ha első sorban a csomagokhoz ragaszkodunk, de mégis másodtanunk kell valamit bennük, akkor a `make package` parancs kiadásával a portokból is tudunk csomagot készíteni, majd átírni 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)` esetben nincsenek futási szintek és nem létezik `/etc/inittab` állomány. Helyette az `rc(8)` vezérlő 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 indító szkriptek segítségével indítja el. Ezek a szkriptek hasonlóak a Linux rendszerében az `/etc/init.d` könyvtárban található szkriptekhez.

A szolgáltatások indítására felelős szkriptek miért kettő különbsz 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 telepített 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 szűfő operációs rendszerként fejlesztik. Ezt megköveteli egy lehetne fogalmazni, hogy a rendszermagot, a rendszerszintű függvénykönyvtárakat és a

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

A szolgáltatásokat az `/etc/rc.conf` fájlban (lásd `rc.conf(5)`) tudjuk engedélyezni a `SzolgáltatásNév_enable="YES"` sor megadásával. A rendszer alapértelmezett beállításait az `/etc/defaults/rc.conf` fájlban találhatjuk meg, ezeket az `/etc/rc.conf` fájlban tudjuk felülírni. Az alkalmazásokhoz tartozó szolgáltatások engedélyezésének lépéséhez pedig a telepítéskor utána ne felejtse el átvizálni a hozzájuk tartozó dokumentációt.

Az `/etc/rc.conf` fájlban szűrmazó módra kiegészítve 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 támogatásnak engedélyezése
apache22_enable="YES"
apache22_flags="-DSSL"
```

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

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

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

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

5. A hálózati beállítás

5.1. Hálózati interfészek

A hálózati csatlakozásokra a Linux esetén alkalmazott általában azonos névvel és utána egy sorszímmal 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-c'm beállítás

Az interfészekhez az `ifconfig(8)` paranccsal tudunk IP-c'met rendelni. Az IP-c'mek beállítás azonban csak akkor marad meg az újraindításkor is, ha felvesszük az `/etc/rc.conf` állományba. A most következő példában megadunk egy hál-zati nevet, IP-c'met és egy alapértelmezett titk-jr-t:

```

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 szabályrendszerét az `ipfw(8)` paranccsal tudjuk kezelni) a FreeBSD fejlesztői által készített és karbantartott tűzfal. A forgalomszabályozás megvalósításához és különböző típusú hál-zati 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 t zzfalat Darren Reed dolgozta ki. Nem csak FreeBSD alatt tal lkozhatunk vele, t bb oper ci s rendszerre is portolt k, t bbek k sz NetBSD-re, OpenBSD-re, SunOS-re, HP/UX-ra  s Solarisra.

Ez az IPFILTER parancs enged lyezi a be rkez  SSH-kapcsolatokat:

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

Az utols  t zzfal, a PF, az OpenBSD Projekt fejleszt se. A PF eredetileg az IPFILTER lev lt s ra k sz lt. Emiatt a PF szab lyainak megad si m dja nagyon hasonl  az IPFILTER eset ben megismertekhez. A min s galap  (QoS) forgalomszab lyoz s l trehoz s hoz a PF az [altq\(4\)](#) megold s val eg sz thet  ki.

Ez a PF parancs enged lyezi a be rkez  SSH-kapcsolatokat:

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

7. A FreeBSD friss' t ze

A FreeBSD rendszer h romf lek ppen friss' thet : forr sk-db-l, bin risan  s telep' t lemezek haszn lat val.

A forr sk-don kereszt li friss' t z s ugyan a legbonyolultabb ezek k sz l, azonban ez k'n lja fel egyben a legnagyobb rugalmass got is. Ennek sor n szinkroniz lnunk kell a FreeBSD forr sk-dj nak n lunk lev  (helyi) m solat t a FreeBSD CVS (Concurrent Versioning System) szervereivel. Miut n ez megt rt nt, le tudjuk ford' tani a rendszermagot  s a hozz  tartoz  programokat. A forr sok friss' t z s vel kapcsolatban olvassuk el a FreeBSD k zik nyv [friss' t zsr-l sz-l-fejezet t](#).

A bin ris friss' t z s a Linux' t pus  rendszereken el rhet  `yum` vagy `apt-get` parancsok eset n megszokottakhoz hasonl . A [freebsd-update\(8\)](#) parancs let lti a friss' t zseket  s telep' ti ezeket. Ez a friss' t zi folyamat a [cron\(8\)](#) haszn lat val  temezhet .

!

Amikor a [cron\(8\)](#) seg' ts g zvel  temezz k a friss' t zseket, a [crontab\(1\)](#)  llom nyban lehet  s g szerint a `freebsd-update cron` parancsot haszn ljuk, ez ltal igyekezz nk cs kkenteni annak val-sz'n s g t, hogy egyszerre t bb sz m' t-g p is ugyanakkor terhelje a szerveret.

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

Az utols  friss' t zi m dszer, a telep' t lemezek haszn lata l nyeg ben egy egy rtelm z folyamat. Ind' tsuk el sz m' t-g p nket a telep' t lemezr l,  s a telep' t ben v lasszuk a friss' t z s (upgrade)

opci-t.

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 verzióban 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 szótlanul 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
É   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 rísvédettek.

Adhatnak olyan alkalmak, amikor mégis szükséges lehet a `procfs` használatára, mint például régi szoftverek futtatása, a rendszerhívások nyomonkövetése a [truss\(1\)](#) segítségével, vagy a [binaris Linux kompatibilitás](#) használata. (Noha a binaris Linux kompatibilitás egy saját `procfs` íllományrendszert, egy [linprocfs\(5\)](#) rendszert használ.) A `procfs` típusú íllományrendszerek csatlakoztatásához a következőt kell megadnunk az `/etc/fstab` íllományban:

```
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 / apt-get install csomag</code>	<code>pkg_add -r csomag</code>	A csomag telepítése egy tívoli számítógépre
<code>rpm -ivh csomag / dpkg -i csomag</code>	<code>pkg_add -v csomag</code>	Csomag telepítése
<code>rpm -qa / 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>pci conf</code>	A PCI-os eszközök megjelenítése
<code>lsmode</code>	<code>kl dstat</code>	A betöltött rendszermagmodulok felsorolása
<code>modprobe</code>	<code>kl dload / kl unload</code>	Modulok betöltése és eltávolítása
<code>strace</code>	<code>truss</code>	A rendszerhívások nyomkövetése

10. Leírás

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