

FreeBSD – a szomszéd vár (3. rész)

A FreeBSD alaprendszer és a külső programok telepítése.

Hajdanán bőven megelégedtünk volna programok oly bő választékával, amit egy *FreeBSD* alaprendszer nyújt: programozható parancssor (burok), szövegszerkesztő programok, fordítóprogramok (GNU C), hálózati munkát segítő programcsomag, s még sok száz kis programocská, amelyek megkönnyítik a munkánkat. Manapság ezen programok megléte teljesen természetes, s egyben az „igazi” munkához kevés.

Az alaprendszer telepítése

Ha minden igaz, akkor rendelkezünk egy friss és naprakész alaprendszer forrással; sőt, talán még a helyes beállítások is már régen megtörténtek. Nincs más teendő, mint lefordítani, majd a helyére másolni az új rendszermagot és az új alaprendszert. Ez egyszerűnek hangzik, de valójában is egy egyszerű művelet. Eleinte azonban bonyolult és összetett tevékenységnek gondoljuk, de hónapok alatt rutinná, majd „szertartássá” válhat.

Ha valaha fordítottunk már mások által írt alkalmazásokat, akkor nem lesz ismeretlen a `make` program használata. Az alaprendszer fordítását és telepítését egy – a fejlesztők által írt – *Makefile* vezényli le, feladatunk ennek a folyamatnak az elindítása, illetve (nem túl megterhelő) megfigyelése. Felmerülhet a kérdés, hogy ha az alaprendszer és a rendszermag együtt biztosít megfelelő működést, akkor mi módon tudjuk a lehető legkevesebb ideig tartó kiesést elérni. Megnyugtatók mindenkit, erre megvannak a megfelelő módszerek, nézzük:

```
$ cd /usr/src/
$ make buildworld
$ make buildkernel KERNCONF=SAJAT
$ make installkernel KERNCONF=SAJAT
```

Az első sor önmagáért beszél, a többről már írnom kell. A `buildworld` hatására készül el az új alaprendszer bináris tükörképe, amelyet egyelőre a `/usr/obj/` könyvtárban találhatunk meg. Ennek a fordításnak az ideje a gép tudásától függően 10-15 perctől (többprocesszoros nagyszámítógép) akár több napig is eltarthat (egy olcsó 486-os PC). A fordítás sikeres végét arról ismerhetjük meg, hogy visszakaptuk a parancssort, és az utolsó egy-két képernyőnyi szövegben nem látjuk az „*error*” szót. Hasonló feladatot lát el a `buildkernel` is, amelynek van egy paramétere, mégpedig annak a konfigurációs állománynak a neve, amely az új rendszermag beállításait tartalmazza. Ezt el is hagyhatjuk,

ekkor *GENERIC* rendszermagot fogunk fordítani, amely az esetek nagy részében jó kompromisszum. A rendszermag fordítása nagyjából negyedannyi időt vesz igénybe, mint az alaprendszeré. Az utolsó parancs a lefordított rendszermagot telepíti a helyére, ezáltal a következő rendszerindításkor már az új rendszermag fogja működtetni a gépünket. Ezen kívül semmi lényeges módosítás nem történt a rendszerben, minden úgy működik, ahogy eddig tette: az alaprendszer változatlan, az új rendszermag pedig nem aktív, hiszen ehhez újra kell indítanunk a gépet. A fordítások befejeztével kiválasztunk egy kellemes és *tervezett* időpontot az újraindítás elvégzéséhez, amikor képesek vagyunk a konzolhoz fizikailag hozzáférni, mivel az elkövetkező pár perces munkát mindenképpen az úgynevezett *singleuser* módban *kell* elvégeznünk. Természetesen lehetőségünk van *multiuser* módban is elvégezni, azonban ez nem támogatott a fejlesztők részéről, így előfordulhatnak nem várt komplikációk. A *singleuser* módot (jelenleg) a negyedik menüponttal válasszhatjuk ki, amikor újraindítjuk a gépünket.

```
$ cd /usr/src
$ mergemaster -p
$ make installworld
$ mergemaster
```

Új fogalom jelent meg: ez a `mergemaster`, amely összetett feladatot hajt végre, az új alaprendszer által megkívánt változtatásokat fésüli össze az aktuális beállításokkal. Nem árt a program futtatása előtt a `/etc/` könyvtár archiválása, bár még nem találgoztam olyan programhibával, amely bármi kárt tett volna (de mint tudjuk: a *FreeBSD* ördög nem alszik :). Az első `mergemaster` elmenti az aktuális állapotot, amelyet az `installworld` módosítani fog. A második `mergemaster` feladata, hogy felderítse és elvégezze a beállítások módosítását, persze a rendszergazda jóváhagyásával. Ezen utóbbi tevékenység nem vesz el pár percnél több időt, így egy éles rendszer kiesése mindössze ennyi idő lesz. Siker esetén – egy megismételt újraindítás után – már az új, friss és ropegős *FreeBSD* alaprendszert vehetjük használatba.

Programok telepítése

Ha többet szeretnénk kihozni a *FreeBSD* rendszerből (és miért ne szeretnénk), mint az alaprendszer használata, akkor programokat kell telepítenünk. Erre sok lehetőségünk van, amelyek között szabadon választhatunk belátásunk szerint.

A sysinstall használata

Legegyszerűbb módja a programtelepítésnek a sysinstall nevű mindenre használata. Mint már említettem, a „Configure” menüpont „Packages” almenüjében, a megfelelő telepítési médium kiválasztása után, egyszerűen kijelöljük a telepítendő programcsomagokat. Ezek előre le vannak fordítva (a használt architektúra „leggyengébb” láncszemére), ezért azonnal telepíthetők. A módszer előnye az azonnali használatbavétel, hátránya az optimalizált fordítás hiánya, amely 5-50% teljesítményvesztést is okozhat. Kezdőknek bátran ajánlom ezt a módszert.

A pkg_add program

Ha nem szíveljük a menürendszer által vezérelt programok használatát, akkor a pkg_add programot nekünk találták ki. Ezzel ugyanis azt végezhetjük el, amit a sysinstall is megtenne, de lassú hálózati kapcsolaton át sokkal hatékonyabb és gyorsabb lehet a munkánk. A program használata egyszerű, bár ismernünk kell a telepítendő csomag pontos nevét:

```
$ pkg_add -r mozilla
```

„Letöltjük, kicsomagoljuk, lefordítjuk, feltelepítjük” módozat

Ez a módszer a legősibb programtelepítési mód UNIX rendszerek esetén. A programok nagy része C/C++ nyelven megírt forráskód formájában áll rendelkezésre, amelyeket használat előtt le kell fordítani. Ez több okból alakult így, részben a UNIX rendszerek változatos hardverei miatt, részben a nyílt forrás okán. Egy programot több tíz architektúrára előfordítani időigényes és felesleges munkának bizonyult, a programok forrása pedig többnyire rendelkezésre állt. Így célszerűen letöltötték-megszerezték minden szükséges program forrását, ezeket lefordították a használat helyén, majd „feltelepítették”. Sajnos ez a módszer egyre összetettebbé vált a programok méretének növekedésével, illetve egyre nehezkesebbé – programozásban kellően nem gyakorlott – rendszergazdák számára is. Ez utóbbi esetben nem a letöltés vagy a kicsomagolás okozott gondot, hanem a forráskód lefordítása. Ehhez ugyanis magas szinten érteni kellett a programok fordításához, illetve a C/C++ nyelvhez, ha egy nem telepített programkönyvtárra vagy forrásfájlcra hivatkozott az aktuális program. A probléma megoldására olyan programot fejlesztettek ki, amely felderíti és leellenőrzi a telepítendő program igényeit, s ezeket hiányát közérthető módon jelzi, ez a configure. Jó esetben egy ilyen telepítés csak pár utasításunkba kerül:

```
$ cd /usr/local/src/program/
$ ./configure
$ make
$ make test
$ make install
$ make clean
```

Sajnos sok olyan programmal találkozni újabban, amelyek nagyon erőteljesen hozzá vannak láncolva a Linux konvenciókhoz. Ezek sok szenvedés árán fordíthatók le FreeBSD alá, mivel olyan dolgokat keresnek olyan helyen, amelyek FreeBSD esetén egyáltalán nincsenek, vagy azon a helyen nem léteznek. Remélhetőleg idővel a fejlesztőknek sikerül közös nevezőre jutniuk.

A ports „adatbázis” használata

A ports adatbázis lényege, hogy a programcsomagok karbantartói a telepítendő programokról kiegészítő információkat gyűjtöttek egybe, amelyek jelentős segítséget adhatnak a fordítás során. Az előkészületek azonosak az alaprendszer karbantartásához: az adatbázist letöltjük a CVS kiszolgálóról, ehhez el kell készítenünk egy sup állományt, amely a ports frissítéséért felelős (ennek neve lehet például ports-sup):

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

```
ports-all
```

Általában az éles szeműeknek is csak egyetlen különbség tűnik fel az src-sup állományhoz képest: src-all helyett ports-all van az állomány végén. Azonban érdemes megnézni a tag opciót is, ahol most egy darab pontot láthatunk csak, ugyanis a ports független a FreeBSD alaprendszer verziójától! A többinek már mennie kell, mint a karikacsapásnak: ki kell adnunk a cvsup ports-sup parancsot, amely a /usr/ports/ könyvtár alá szinkronizálja a legfrissebb adatbázist. Jogos kérdéseket vélek hallani: „De mire jó ez a 300 megabájt letöltött adat?” A kérdés jó!

A ports adatbázis (jelenleg) közel 11500 program adatait tartalmazza, amelyeket felkészítettek a FreeBSD alatti komplikációktól mentes fordításra. Több tíz kategória közül választhatunk, amelyek sok száz – a kategóriába tartozó – programot tartalmazhatnak. Például a BitchX IRC kliens-program feltelepítéséhez bele kell lépünk a /usr/ports/irc/bitchx könyvtárba, ahol körülnézve alig látunk néhány állományt. Ez így van rendjén, ugyanis csak az összefoglaló információkat tudhatjuk meg a kiválasztott programról (pkg-descr), a feltelepítendő állományok nevét és helyét (pkg-plist), valamint egy Makefile is ott árválkodik. A program forrásának nyoma sincs, és ez így van rendjén. Gondoljunk csak bele, hogy 11500 program forrása mekkora helyet igényelne! A make parancsot kiadva tudjuk a program telepítését megkezdeni, s a gépezet működni kezd: letölti a megadott helyek egyikéről a program forrását:

```
>> ircii-pana-1.1-final.tar.gz doesn't seem to
  exist in /usr/ports/distfiles/.
>> Attempting to fetch from ftp://ftp.bitchx.org/
  pub/BitchX/source/
```

Amint láthatjuk, először a /usr/ports/distfiles/ könyvtárban keresi a letöltendő állományt, amely könyvtár egyfajta gyorsítótár funkciót valósít meg. Ha elfeledkeznénk erről a könyvtárról, könnyen előfordulhat, hogy valami rejtélyes oknál fogva elfogy a hely a /usr/ fájlrendszeren. Ilyen esetben elsősorban erre a könyvtárra gyanakodjunk (második gyanúsítottunk a /usr/obj/ lehet, amely az alaprendszer lefordított állományainak ad helyet).

A letöltött forrás kicsomagolásra kerül, s innentől azonos módon járunk el, mint az előző módszer: lefordítjuk és fel-

telepítjük a kívánt programot. Látható, hogy a felmerült problémák nagy részét sikerült megoldani, mivel nem kell keresgélni és letölteni a programok forrását, illetve teljesen biztosan működni fognak *FreeBSD* alatt. Továbbá már nem kell kézzel vadászni a szükséges komponenseket, mivel a *ports* adatbázis már tartalmaz információkat a függőségekről, így rekurzív úton automatikusan feltelepítésre kerül minden szükséges csomag.

A portinstall használata

Szinte hallom a morgolódást, hogy „Ennyi? Mi ebben a rendszergazdabarát?” Igen, én is ezt kérdeztem magamtól, amíg rá nem találtam az „egyetlen igaz útra”: a `portinstall` használatára. Aki egyszer ráérez az ízére, soha nem feleli. A `portinstall` a `/usr/ports/sysutils/portupgrade/` könyvtárból telepíthető fel (mivel azonos a `portupgrade` programmal), természetesen az előző részben megismert módon. A programok telepítésének (eddigi) legkényelmesebb módja a `portinstall` program használata, amely jelentősen megkönnyíti ezt a tevékenységet. Első használatkor a program elkészíti a saját adatbázisát a feltelepített és a feltelepíthető csomagokról, ez sok ideig is eltarthat. Alapvetően könnyű a használat, egyszerűen meg kell neveznünk a kívánt program nevét, `portinstall` `apache`, ezzel elindul a megnevezett program telepítése. Feltéve, ha van ilyen, illetve csak egyetlen ilyen nevű programot talál a `portinstall`. Telepíthető program hiányában egy hibaüzenetet kapunk, amely arról tájékoztat, hogy nincs ilyen nevű program a *ports* adatbázisban. Több találat esetén sorban megkérdezi a feltelepítendő program pontos nevét, és verziószámát:

```
--> Found 2 ports matching 'apache':
      www/apache13
      www/apache2
Install 'www/apache13'? [yes] n
Install 'www/apache2'? [yes] y
```

A telepítés innentől teljesen azonosan történik a *ports* adatbázis használatával, eltekintve attól, hogy nem kell más parancsot kiadnunk, a `portinstall` mindent megold és elvégez. Érdeemes a telepítéskor mindig rákérdeztetni a telepítendő programokra, ezzel sok felesleges munkától szabadulunk meg (nem kívánt program eltávolítása); ezügyben a `-i` opciót kell használnunk. Fontos lehet a `-P` opció ismerete is, amely kombinálja az előfordított csomag telepítését a `portinstall` előnyeivel: a program megpróbálja letölteni a kívánt csomag előfordított állományát, s csak akkor használja a *ports* adatbázist, ha nincs ilyen előfordított állomány. Ezzel időt tudunk megspórolni, ha gyorsan kell egy programot telepíteni.

A portupgrade használata

A telepítések mellett nagyon fontos a telepített programok naprakészen tartása. Ehhez alapvetően a *ports* adatbázis frissítése szükséges, amelyet a `cvsup ports-sup` parancs kiadásával tehetünk meg. Ekkor a frissítendő programot el kell távolítani, majd a helyére feltenni az újabb verziót. Feltéve, ha van újabb, viszont ennek eldöntése kicsit „favágó” munka, mert egyenként végig kell lépkednünk a telepített programokon, és megnézni, hogy van-e újabb.

Ezt az idegőrlő munkát teszi átláthatóvá és egyszerűvé a `portupgrade` program, ugyanis végignyálazza helyettünk a feltelepített programokat; majd utánanézi, hogy van-e újabb verzió. Gyakorlatilag csak a `portupgrade -a` parancsot kell kiadni, óvatosabbak a `-i` opció is használhatják `portupgrade -a -i`.

Általános esetekben némi molyolás után listát kapunk a feltelepített és a frissíthető programokról. Előfordulhat azonban, hogy a programcsomagok függőségei a frissítés során (aljas módon :) megváltozhatnak. Ekkor egy (első látásra riasztó) üzenetet láthatunk, mint például

```
Stale dependency: intlfonts-1.2.1 -> XFree86-
↳ 3.3.6_11 - manually run 'pkgdb -F' to fix, or
↳ specify -o to force.
```

Nem kell megijedni, csak futtatnunk kell a `pkgdb -F` programot, amely a segítségünket igénybe véve kijavítja a hibát.

```
Stale dependency: intlfonts-1.2.1 -> XFree86-
↳ 3.3.6_11 (x11/XFree86):
New dependency? (? to help):
```

A megoldás egyszerű, az *Xfree86* csomag helyett már *Xorg* néven kell a grafikus felületre hivatkozni:

```
New dependency? (? to help): xorg-6.7.0_1
Fixed. (-> xorg-6.7.0_1)
```

Ugye, nem is volt olyan bonyolult? Figyelni kell arra, hogy a függőség megadásánál pontosan nevezzük meg a programot, annak minden verziószámával együtt. Sajnos ez a „hiba” ritka, és ennél fogva zavarba ejtheti a gyanútlan *FreeBSD* rendszergazdát. Ennél is ritkább, ha szinte egy komplett nyomozati apparátusra van szükségünk, ha fel szeretnénk deríteni a hiba okát, mivel igencsak rejtélyes lehet a függőségek szövedéke. Érdeemes gyakran frissíteni a *ports* adatbázist, majd azonnal futtatni a `pkgdb -F` parancsot, mivel hosszabb idő (hónapok) kihagyása után esélytelen lehet a függőségek kibogozása.

A fenti problémák nélkül egy listát kapunk a telepített programokról, amelyeknek nagy részét nem kell frissíteni, mert nincs belőlük újabb verzió. Természetesen felkérhetjük a `-f` opcióval, hogy minden telepített programot fordítson újra: előfordított programok esetén megtehetjük, ha rengeteg felesleges időnk van.

```
** No need to upgrade 'xpdf-3.00_3' (>= xpdf-
↳ 3.00_3). (specify -f to force)
** No need to upgrade 'urwfonts-1.0' (>= urwfonts-
↳ 1.0). (specify -f to force)
```

Néhány csomag esetén viszont megkapjuk a kérdést:

```
--> Upgrade of java/jdk14 started at: Wed, 03 Nov
↳ 2004 19:39:06 +0100
--> Upgrading 'jdk-1.4.2p6_5' to 'jdk-1.4.2p6_6'
↳ (java/jdk14)
OK? [yes]
```

Elég egy `ENTER`t ütni, s elindul a kiválasztott csomag frissítése, amely során a portupgrade elkészíti az újabb verzió telepítésre kész csomagját, kivesszi a csomaglistából a régi programot, elmenti a régi csomag minden telepített állományát, majd megpróbálja az újat telepíteni. Ha sikerül, akkor nincs már semmi gond, ettől kezdve az újabb programot használhatjuk. Probléma esetén visszakerül a helyére a régi program, így a hibák nem okoznak felesleges leállást.

Program keresése a ports adatbázisban

Gyakran előfordul az a helyzet, amikor egy bizonyos feladatra keresünk megfelelő programot. Egyik lehetséges megoldás, hogy a számunkra érdekes kategóriát nézzük át, és bízunk a szerencsénkben: hátha belebotlunk a keresett programba. Sokkal célravezetőbb megoldás, ha a `/usr/ports` könyvtárba lépve utasítást adunk erre a keresési feladatra. Ez egyszerűen egy megfelelő `make` parancs kiadása lesz:

```
$ make search key=open | less
```

Ezzel az utasítással a program végignyázza a teljes ports adatbázist, és minden olyan csomagot kiír, amely neve, rövid vagy bővebb leírása tartalmazza a megadott kulcsot, amely jelen esetben az „open” szó. A „less” programot azért érdemes futtatni, mert sok száz találat is lehet, amelyek hamar felfutnak a képernyőn. Előfordulhat, hogy olyan találatot is olvashatunk, amely látszólag nem tartalmazza a kulcsszót:

```
Port: lzo-1.08_1
Path: /usr/ports/archivers/lzo
Info: Portable speedy, lossless data compression
     ↳ library
Maint: ports@FreeBSD.org
```

Nem kell azonban aggódnia, a keresés kiválóan működik, a program bővebb leírása tartalmazza az „open” kulcsszót, egyszerűen csak ez a bővebb leírás nem jelenik meg. A kapott listát végigkövetve hamar megtalálhatjuk azt a programot, amelyet fel szeretnénk telepíteni.

A telepített programok adatbázisa

A telepített programok leírásai egy speciális adatbázisban foglalnak helyet, amelyet a `/var/db/pkg/` könyvtárban találunk. Nem kell hozzá speciális program, ugyanis egy program – egy könyvtár elven találjuk meg az összes regisztrált (telepített) program jellemzőit.

Természetesen létezik külön program, amely kényelmesebbé teszi ezen adatbázis használatát, ez a `pkg_info`, amelynek annyi a dolga, hogy a paraméterében megkapott csomagról információkat szolgáltatson:

```
$ pkg_info hu-openoffice-1.1.3
Information for hu-openoffice-1.1.3:
```

```
Comment: Integrated wordprocessor/
↳ dbase/spreadsheet/drawing/chart/
↳ browser
```

Programok törlése

Eljőhet az idő, amikor egy telepített program feleslegessé válik, bár én ritkán találkozom olyan esettel, amikor programok törlése valóban szükséges lépésnek bizonyul (gondoljunk csak a merevlemezek szinte határtalan kapacitására). Az egyik törlési megoldás igazából nem törlés, hanem a program frissítése során alkalmazott technikai megoldás, amely azonban önmagában is használható: `pkg_delete`. Ekkor a telepített állományokat nem töröljük a helyükről, csak a telepített csomagok adatbázisából kerülnek ki, így azonos program újabb verziójának telepítésénél nem lesz akadálya. Nem szép megoldás, ezért ne nagyon használjuk függőségi problémák megoldására.

Véglegesen a `pkg_delete` parancs segítségével tudjuk törölni a kívánt programot, ekkor a telepített programok adatbázisa alapján a törölődnek a nem módosított állományok. Ez utóbbi azért fontos, mert különben a programok beállításai is törölődnek:

```
$ pkg_delete lsof-4.73.1
```

A programnak hibátlan lefutás esetén nem lesz kimenete, ha szeretnénk, hogy részletesen számoljon be a tevékenységéről, akkor használunk kell a `-v` paramétert is. Ha nem tudjuk pontosan a törölendő program nevét *verziószámával* együtt, akkor használhatjuk a megszokott helyettesítő karaktereket is:

```
$ pkg_delete lsof*
```

Mind a kettő program azonos paramétereket fogad el, amelyek közül elsődlegesen a `-f` lehet érdekes számunkra, amely megadásakor a törlés akkor is végrehajtódik, ha függőségi problémák merülnének fel. Érdemes a `-r` és a `-R` paraméterre is gondolnunk, amelyek hatására a törlés rekurzíván hajtodik végre. A `-R` esetén mindazon csomagok törlésre kerülnek, amelyekre a megadott csomagnak szüksége van, ezzel biztosak lehetünk benne, hogy nem maradnak olyan telepített csomagok, amelyekre igazából már nincs más csomag felől hivatkozás (így igazából feleslegesek). A `-r` ennek ellentéte: az összes olyan csomag törölődni fog, amelyeknek a megadott csomagra szüksége van; ezzel óvatosan érdemes bánni, mert egy alapvetően szükséges csomagot megadva akár az összes telepített csomag törölődni fog.



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/>.