

## FreeBSD – a szomszéd vár (5. rész)

Az előző részből már csak néhány „apróság” maradt ki, amelyek még szükségesek egy átlagos munkaállomáson: a nyomtatás, a hálózat beállítása, illetve CD és DVD írás.

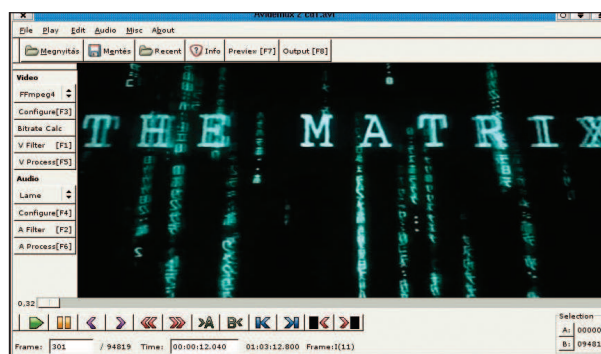
A legnagyobb kihívást a nyomtatók beállítása jelenti, ezek között van néhány amelyet a *FreeBSD* egyik meghajtója vagy segédprogramja sem támogat. A hálózat már egyszerűbb – bár sokrétűbb feladat – mivel a *FreeBSD* lételeme a hálózat, így nagyon kevés eszközt nem támogat, amelyek köze van bármilyen hálózati kapcsolathoz. Végül a *CD* és *DVD* készítéséről írok, amely szintén nem okoz nehézséget, mivel része az alaprendszernek egy *CD*-író program.

### Nyomtatás

A nyomtatás lehetősége már hozzátartozik a legtöbb számítógéppel végzett tevékenységhez, szinte minden felhasználói program támogatja a nyomtatás lehetőségét, legyen a nyomtatás tárgya egy levelezőprogram esetén egy levél, egy böngészőprogram ablakának tartalma, akár egy képszerkesztő programban egy retusált fénykép.

Legegyszerűbb dolgunk akkor van, ha a nyomtatónk képes értelmezni a *PostScript* (*PS*) állományokat, mivel a legtöbb program ebben a formátumban próbálkozik meg a nyomtatással. Azt is mondhatnánk, hogy *UNIX* alatt a programok kivétel nélkül szimpla szöveggént, vagy *PostScript* nyelven próbálnak nyomtatni.

Sajnos a legtöbb manapság kapható nyomtató nem érti a *PostScript* nyelvet, és pár éve már a szimpla szöveges (úgynevezett *DOS kompatibilis*) nyomtatást sem. A nyomtatók egyszerűen csak így lehettek annyira olcsók, hogy szinte bárki képes legyen megvenni egyet (értem ez alatt a *PostScript* és a szöveges implementáció teljes hiányát). Nem kell azonban elkeserednünk, ha csak egy olcsó lézernyomtatónk van, hiszen ezek leginkább cégeknél vannak használatban, és ezért a legtöbbet támogatja a *GhostScript* programcsomag, amely a ports adatbázisból feltelepíthető, és a print kategóriában a *ghostscript-gnu* alatt található meg. Sajnos nem triviális dolog beállítani a megfelelő szűrőket, amelyeket az `lpd` majd kezelni fog, ezért rövidítsünk a problémán: telepítsük fel az *apsfilter* csomagot (`/usr/ports/print/apsfilter`). Ennek segítségével könnyedén be tudjuk állítani a rendszert a nyomtatáshoz. Egyetlen megkötése van a program használatának: úgynevezett *Cardware* kategóriába tartozik, vagyis nem ingyenes a használata. Nem kell megjedni, csak küldenünk kell egy képes-



1. ábra

lapot a szerző címére, ez maximum párszáz forintunkba kerül, ennyit megér a használata. Az elején egy szándéknyilatkozatot kell elfogadni, amelyben megígérjük, hogy küldünk majd egy lapot. A beállítás a

```
$ /usr/local/share/apsfilter/SETUP
```

parancs kiadása után indul el. Néhány ismertető képernyőszöveg után eljutunk a program főmenüjéig:

- (D) Available Device Drivers in your gs binary
- (R) Read Ghostscript driver documentation
  - ↳ (devices.txt)
- (1) Printer Driver Selection [pcl3/unspec]
- (2) Interface Setup [parallel]
- (3) Paper Format [a4]
- (4) Printing Quality [high]
- (5) Color Mode [full]
- (6) Print Resolution in "dots per inch"
  - ↳ [1200x1200]
- (7) Default Printing Method [auto]
- (T) Print Test Page
- (V) View performance log (times of print attempts)
- (A) Abort installation (don't do anything)
- (I) ==> Install printer with values shown above -
  - ↳ repeat this

step for installing multiple printers  
(Q) ==> Finish installation

A beállítás nem bonyolult, a számozott lépéseken kell áthaladnunk, kiválasztjuk a nyomtatónk kategóriáját (milyen meghajtóprogram-csoport kezelje), majd a nyomtató pontos típusát. Ezek után kiválasztjuk a csatoló felületet, vagyis azt, hogy milyen kábellel kötöttük össze a gépünkkel (párhuzamos, **USB**, soros), illetve milyen eszköznévvel éri majd el a program (*/dev/lpt0*, */dev/ulpt0*). A papírméret, a minőség, a színek jellege, illetve a felbontás triviális válaszokat igényel. A beállítások végeztével tudunk próbanyomatot kérni a T megnyomásával, ha ez sikeres volt, akkor az I lenyomásával hozzáadhatjuk a rendszernyomtatókhoz az általunk megadott néven. Mindenképpen készítsünk egy általános lp nevű nyomtatósort, mivel ezt keresi a legtöbb program. Ellenben érdemes több kombinációt elkészíteni (például 1p-mono-300-draft, 1p-color-1200-photo, ...), így a megfelelő nyomtatósor kiválasztásával kiválasztjuk a nyomtatás jellemzőit is. A Q megnyomásával elkészül a */etc/printcap* állomány, ezzel végeztünk is a nyomtató beállításával, bármilyen programból képesek leszünk nyomtatni.

## Hálózat

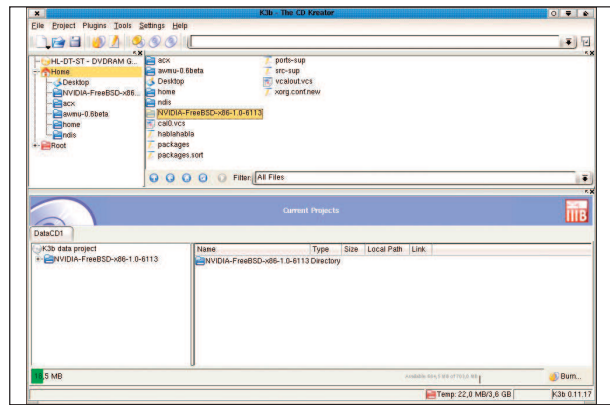
Manapság hálózati hozzáférés nélkül nehéz boldogulni, gondoljunk itt vállalati vagy otthoni munkaállomásra. Legkönnyebb dolgunk egy vállalati hozzáférés kialakításánál van, mivel ott gyakorlatilag csak **UTP** kábellel kialakított **LAN** található, egyszerűen csak be kell állítanunk **DHCP**-t vagy fix **IP** címet, és a megfelelő átjárót (esetleg a **proxy** kiszolgáló címét). Ezen paraméterek megszerzéséhez a vállalati rendszergazdáját kell megkérnyékeznünk.

Otthoni hálózati hozzáférés nagyon sokrétű, sokkal nehezebb dolgunk van ezek beállításával: a modem, az **ADSL**, a **GPRS** vagy a **WiFi** mind egy-egy megoldás a sok közül. Mindegyik módszer más-más alrendszer igényel az internetre való kapcsolódás folyamán.

## Modem és ISDN

10 éve szinte csak modemmel lehetett hozzáférni a világhálózathoz, így ez a legősibb és egyben legstabilabban implementált kapcsolódási mód. Legalábbis ez sokáig igaz volt, amíg a költségcsökkentés okán a gyártók el nem kezdtek a modemeken is spórolni. (Tizenegy éve egy 28800bps külső modem számított a csúcsnak, amelynek ára a mai minimálbér, vagy ha úgy tetszik az akkori közalkalmazotti átlagkereset majd kétszerese volt. Máig őrzöm, mert még anyag van benne...). Ennek megfelelően érthető a spórolás oka, viszont a módszer nagyon sok ablakmentes felhasználónak okozott szenvedést. A modemet vezérlő szoftvert átmozgatták a számítógép oldalára, így az operációs rendszerre bízta annak működtetését meghajtóprogramot pedig csak **Windows** alá írtak. Sajnos az utóbbi öt évben kapható modemek – kevés kivételtől eltekintve – mind ilyen szoftver-modemek, amelyek **Linux** általi támogatása is nagyon lassan fejlődött ki, **FreeBSD** alatt viszont szinte csak a hardver-modemek a támogatottak. Ennek megfelelően a rendszermag által írt üzenetek között találnunk kell a

```
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type 16550A
```



2. ábra

hasonló sorokat, és ekkor a */etc/ppp/ppp.conf* állományt megfelelő módon szerkesztve képesek leszünk a szolgáltatónk által biztosított internet elérés használatára. Ha ilyet nem találunk, akkor nagy eséllyel – sajnos – nem fogunk hálózathoz csatlakozni modemmel.

**ISDN** modemek nagy része támogatva van, meg kell keresnünk a soros kommunikációt lehetővé tévő eszköz nevét, s azt kell használnunk.

A *ppp.conf* szerkesztése nem olyan nehéz, mint először látszik. A legrövidebb beállítás alább látható, a vastagon szedett szöveget kell csak szerkeszteni, illetve több szolgáltató esetén az elkülönülő blokkot kell még egy példányban (más elnevezéssel) lemásolnunk:

default:

```
set log Phase Chat LCP IPCP CCP tun command
ident user-ppp VERSION (built COMPILATIONDATE)
set device /dev/cuaa0
set speed 115200
set dial "ABORT BUSY ABORT NO\\SCARRIER
TIMEOUT 5 \
    \\\" AT OK-AT-OK ATE1Q0 OK \\DATDT\\T
    TIMEOUT 40 CONNECT"
set timeout 180
enable dns
```

szolgáltató:

```
set phone "+3651-xxx-xxx"
set authname username
set authkey password
set login "TIMEOUT 10 \\\" \\\" gin:-gin: \\U
    word: \\P col: ppp"
set timeout 300
set ifaddr 0.0.0.0 255.255.255.255
    255.255.255.255 0.0.0.0
add default HISADDR
```

A tárcsázás nem okoz nehézséget, be kell írunk (root joggal) a parancssorba, hogy **ppp**, majd kiadni a **dial** szolgáltató parancsot.

## ADSL

Nagyfokú különbség nem fogad minket, ha **ADSL**-t szeretnénk használni, egyszerűen a megfelelő hálókártyába

dugjuk az *ADSL* „modemet” a kapott kábellel, majd létrehozunk egy *ppp.conf* állományt a következők szerint (a vastagon szedett szöveget szerkesztjük):

```
default:
  set log Phase tun command # you can add more
  ↳ detailed logging if you wish
  set ifaddr 10.0.0.1/0 10.0.0.2/0
```

```
szolgaltato:
  set device PPPoE:x11
  set authname username
  set authkey password
  set dial
  set login
  add default HISADDR
```

Az *x11* helyett a megfelelő hálózati csatoló megnevezését kell írunk. A tárcsázás azonos módon történik, mint modem esetén.

## GPRS

Dinamikus és gyorsan fejlődő ága a hálózatoknak a mobil végpontok használata. Ekkor egy lappal és egy mobiltelefonnal a világ végéről is tudunk kapcsolódni a világhálóra. Több elterjedt módja is van a mobiltelefon számítógéphez való kapcsolásának: soros kábel, *USB* kábel, *IrDA* vagy *BlueTooth*.

A legkényelmesebb megoldás *FreeBSD* szempontjából a kábeles kapcsolat, amelyet soros portra kell kötnünk. A *GPRS* képes telefonok igen nagy része képes modemként működni, így a *ppp* programot tudjuk kapcsolódásra használni. A *ppp.conf* sem lesz összetettebb, mint modem-es kapcsolat esetén:

```
default:
  set log Phase Chat LCP IPCP CCP tun command
  ident user-ppp VERSION (built COMPILATIONDATE)
  set redial 2.2 300
  set reconnect 2.2 300
  set device /dev/cuaa0
  set speed 115200
  set timeout 180
  enable dns
```

```
szolgaltato:
  enable force-scripts
  disable ipv6cp
  set lcp-echo-failure 0
  set authname username
  set authkey password
  set phone “*99***1#”

  set dial “ABORT BUSY ABORT NO\\SCARRIER TIMEOUT 5 \
    \\” AT OK-AT-OK ATE1Q0 OK \\DATD\\T
    ↳ TIMEOUT 40 CONNECT”
  set logout “ABORT BUSY ABORT ERROR TIMEOUT 30 \
    \\” +++ATH OK-ATH-OK AT+CGATT=0 OK”
  set login
  set timeout 300
  enable dns
```

```
resolv rewrite
```

```
set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.0
↳ 0.0.0.0
add default HISADDR
```

*USB* kábelek közül a legtöbbször egy kis lapka *USB*-soros átalakítást csinál, amelyet az operációs rendszer visszaalakít soros kommunikációra a megfelelő meghajtóprogrammal. A gyakori lapkákat támogatja a *FreeBSD*, de azért ne bizzuk a véletlenre – próbáljuk ki megvásárlás előtt a kábelt. Infravörös átvitel sokáig divatos volt a mobil kommunikációban, nagyon sok gyártó készített asztali gépekhez *IrDA* csatlakozót, ezek egy része működni is fog *FreeBSD* alatt: egy soros port fog létrejönni, ezt tudjuk megadni a *ppp.conf* állományban és egyszerűen csak tárcsáznunk kell. Leggyakoribb és legkényelmesebb megoldás a *BlueTooth* kapcsolódás, szinte az összes 1-2 éves telefon ismeri a kékfogat, mint adatátviteli módszert. A számítógép 10-15 méteres sugarában bárhol lehet a telefon, nincs egy másfél méteres kábelhez vagy infravörös látótávolsághoz kötve. Praktikus is, mivel egy *USB BlueTooth* végpont annyiba kerül, mint egy adatkábel, viszont szabadon válthatunk telefont, a *BlueTooth* azzal is működni fog. *FreeBSD* alatti beállítása és használata önmagában egy egész cikket megtöltene, ezért most csak a lényegi részére szorítkozok (a többi olvasható – egyelőre – angol nyelven a *FreeBSD* kézikönyvben). Egy *rfcomm* nevű alrendszerre lesz szükségünk, amely a széleskörűen használható, szélessávú *BT* adatcsatornából kiharásít egy részt virtuális soros kommunikáció céljára. Ehhez egyedül a telefon *BT MAC* címére lesz szükségünk: `$ rfcomm_pppd -a 00:0E:07:62:57:28 -d -c -C dun -l` ↳ szolgaltato

A program futása során létrehoz egy ilyen virtuális csatornát, és elindítja rajta a *ppp.conf* állományban leírtak alapján (szolgaltato).

## WLAN és WiFi

Ha rendelkezünk egy megfelelően konfigurált *802.11g* vagy *802.11b* *WLAN* kártyával, akkor előttünk a lehetőség a vezeték nélküli szélessávú hálózatok használatára (akár 54Mbps elérésére is). A legtöbb *WiFi* kártya (legyen az *PCI* vagy *PCMCIA*) támogatott *FreeBSD* alatt, ha nem natív meghajtóprogrammal, akkor az *Ndiswrapper* segít ebben, amelyet az előző részben már említettem. Sikeresen beállított *WiFi* kártya esetén az *ifconfig wi0* parancs hatására az alábbi láthatjuk:

```
wi0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,
↳ MULTICAST> mtu 1500
  inet6 fe80::202:2dff:fe2d:c938%wi0
  ↳ prefixlen 64 scopeid 0x7
  inet 0.0.0.0 netmask 0xff000000 broadcast
  ↳ 255.255.255.255
  ether 00:09:2d:2d:c9:50
  media: IEEE 802.11 Wireless Ethernet
  ↳ autoselect (DS/2Mbps)
  status: no carrier
  ssid ""
```

```
stationname "FreeBSD wireless node"
channel 10 authmode OPEN powersavemode OFF
↳ powersavesleep 100
wepmode OFF weptxkey 1
```

A beüzemeléshez mindenképpen meg kell adnunk egy SSID azonosítót, például

```
$ ifconfig wi0 ssid 'azonosito'
```

s ezek után már a

```
status: associated
ssid "azonosito"
```

szöveget kell látnunk majd, ha az adott nevű hálózat vételi körzeten belül van. Ezzel hozzá tudunk kapcsolódni egy nyilvános vezeték nélküli hálózathoz, a legtöbb esetben azonban WEP titkosítással vannak a lefedett körzetek védve. A WEP megadásához szintén az ifconfig parancsot tudjuk használni:

```
$ ifconfig wi0 wepmode on wepkey 0x0123456789ABCDEF
```

Az IP cím és a többi információ megadása – beállítása azonos módon történik egy átlagos hálózati kártya esetén megszokott módszerrel.

## Levelezés

Alapvetően minden szükséges program megvan az alaprendszerben, ha levelezni támad kedvünk, hiszen egy *Sendmail* programot használhatjuk levelek fogadására és küldésére. Alapbeállításképp a *sendmail* nem fogad leveleket a gépünk IP címén, csak a 127.0.0.1/localhost címen, de ezen változtathatunk a */etc/rc.conf* szerkesztésével `sendmail_enable="YES"`

és ezáltal már (megfelelő beállítások után) fogad kívülről is leveleket a program. Ha nem használjuk levélküldésre a gépünket – például a szolgáltató SMTP szerverét használjuk, akkor a */etc/rc.conf* állományba a `sendmail_enable="NONE"`

sort írjuk be. Ezáltal a *sendmail* nem fog elindulni, így ennyivel több erőforrás marad a többi programunk számára. Levelező programok közül szinte az összeset megtaláljuk a ports adatbázisban, legyen az a pine vagy a mutt; illetve grafikus felületen az *Evolution* vagy a *Kmail*.

## CD írás

A *FreeBSD* olyannyira támogatja az optikai tárolóeszközök (CD/DVD) használatát, hogy az alaprendszere már (az 5.0, illetve a 4.8 verziók óta) tartalmaz CD-író programot: a *burncd-t*, amelyet a `/usr/sbin/burncd` helyen találunk meg. Sajnos – a többi íróprogramhoz hasonlóan – nem alkalmas arra, hogy egyszerű felhasználóként CD-t írhasson bárki is, mivel erre nincs kitaposott megoldás (esetleg a `sudo` tűnhet még jó ötletnek).

A *burncd* használata önmagában kissé nehézkes, mivel csak *ATAPI* írókat támogat, illetve *nincs* az alaprendszerben olyan program, amely elkészíti az *ISO9660* fájlrendszert, s erre szükségünk lesz egy összetettebb egyedi CD vagy DVD írásához, ezért érdemes lesz feltelepítenünk az

*mkisofs* programot (a *cdrtools* ports csomag része). Ezáltal tudunk zenét CD-re égetni, akár CD-t másolni, de például mentésre és adatok felírására is használható ez a program...

Természetesen a ports adatbázisban találunk egypár segédprogramot, amelyek jobban és kényelmesebben használhatók a CD/DVD készítésekor.

A legfontosabbat már említettem: az *mkisofs* nélkülözhetetlen eszköz a CD/DVD fájlrendszerének elkészítéséhez. Használata teljesen azonosan történik, mint *Linux* esetén, a paraméterezése is ugyanaz.

Sokkal jobban járunk, ha a *burncd* helyett a *cdrecord* programot használjuk, amelyet jó esetben az *mkisofs* programmal együtt már feltelepítettünk, ugyanis a *cdrtools* csomag része. A *cdrecord* támogatja a SCSI és az USB íróeszközöket is.

Érdekességképpen az *ATAPI CD* meghajtók esetén a zenét tartalmazó CD-k egyszerűen másolhatók, mivel létrejön a */dev/acd0* eszközön kívül minden sávhoz tartozni fog egy */dev/acd0t $n$*  eszköz, amely az adott sávot tartalmazza. Ezt a másolást legegyszerűbben a

```
$ dd if=/dev/acd0t01 of=track01.cdr bs=2352
$ dd if=/dev/acd0t02 of=track02.cdr bs=2352
```

paranccsal tudjuk megtenni, természetesen a megfelelő számú sávval és megfelelő nevű eszközzel. A kapott *.cdr* állományokat fel tudjuk írni egy üres CD-re a következő paranccsal:

```
$ burncd -f /dev/acd0 audio track1.cdr track2.cdr
↳ ... fixate
```

Természetesen a *burncd* segítségével képesek vagyunk CD lemezeire írni az aktuális mentést, ehhez a

```
$ burncd -f /dev/acd0 -s 12 data archive.tar.gz
↳ fixate
```

parancsot kell kiadni, amely úgy írja a CD lemezeire az adatot, hogy az különösebb fájlrendszerként funkcionálna. A probléma mindössze annyi, hogy nem tudjuk felcsatolni a fájlrendszerbe, hanem a

```
$ tar xzvf /dev/acd1
```

parancs magáról a CD eszközről binárisan olvasva tömöríti ki a ráírt állomány tartalmát. Nos, ezt a kis nehézséget tudjuk az *mkisofs* használatával áthidalni...

Mivel a téma igazából a munkaállomások beállítása, azért feltelepíthetünk akár grafikus felületű CD író programot, mint például az *X-CD-Roast* (`/usr/ports/sysutils/xcdroast`) vagy a *K3b* (`/usr/ports/sysutils/k3b`). Ezekkel kényelmesebbé válik az írás, viszont nehézséget okozhat, hogy az eszközhöz root jogok kellenek, így vagy root jogokkal használjuk a grafikus felületet, vagy – például – a *kdesu* programmal oldjuk meg a rendszergazdai jogokkal való futtatást...

## DVD írás

DVD írásához már más eszközök szükségesek, mint például a *growifsofs*, amely a *dvd+rw-tools* (`/usr/ports/sysutils/dvd+rw-tools`) csomag része. A DVD írásához nem szükséges használnunk az *mkisofs* programot,



de telepítve rendelkezünk kell vele, mert a program meghívja azt. Egy egyszerű adat DVD írásához a

```
$ growisofs -dvd-compat -Z /dev/cd0 -J -R
↳ /home/usernev/könyvtar/
```

parancsot kell kiadni, és ezzel a megadott könyvtár tartalma a DVD-re íródik. Az újraírható DVD típusok (DVD±RW) írása hasonlóképpen történik, ezekhez hozzá tudunk írni (-M opció), illetve törölni is tudjuk a tartalmát (-Z opció):

```
$ growisofs -Z /dev/cd0 -J -R
↳ /home/usernev/könyvtar/
$ growisofs -M /dev/cd0 -J -R
↳ /home/usernev/könyvtar/
```

A DVD+RW lemezeket azonban először meg kell formázni, amelyhez a

```
$ dvd+rw-format /dev/cd0
```

parancsot kell használnunk. Többször is megformázhatunk egy-egy DVD+RW lemezt, bár ennek nincs sok értelme, akkor próbáljuk ki, ha nagyon nem tudjuk már kezelni a lemezt. Képesek vagyunk DVD-RAM lemezeket is használni, személy szerint mentéshez ajánlom ezeket, mert a lemezeket szokásos módon megformázzuk

```
$ newfs -U -L "cimke1" -m 1 /dev/cd0
```

majd - szintén szokásos módon – felcsatoljuk a megfelelő helyre:

```
$ mount /dev/cd0 /mnt
```

Ezek után úgy tudjuk használni, mint egy 4.5G-s nagyon gyors hajlékonylemezt (vagy lassú merevlemezt). Bármilyen programmal tudunk rá fájlokat másolni és törölni; állományokat szerkeszteni, közvetlenül mentést készíteni rá (és cserélni a lemezeket, ha szükséges). Egy probléma van csak, a FreeBSD nem tudja olvasni sem a Linux által készített UDF2.0 fájlrendszert... de ez talán idővel megoldódik.

Ha kedvünk támad, akkor videó DVD-t is készíthetünk, ehhez egy tetszőleges videóállomány, és két program kell csak: az *Avidemux2* és a *DVDAuthor*.

Az *Avidemux2* egy videószerkesztő program, amely nevéből adódóan az AVI formátumra van felkészítve, bár olvas és ír nagyon sok más videóformátumot is.

A program részletes ismertetése meghaladja a cikk keretét, ezért csak a témára szorítkozok. Egy DVD lemezre MPEG2 formátumban kell a videót elkészíteni, a hozzá tartozó hangokat pedig AC3 formátumban. Több hangszávet is tudunk egyszerre kezelni (esküvői videó esetén az eredeti és a rákevert hangot például).

A legördülő menüknél kiválasztjuk a DVD és az AC3 formátumot, a szűrőknél a videó méretét átállítjuk DVD méretre (720x576), majd bekapcsoljuk mind a két *Process* gombot. Ekkor a videót elmentve megindul a konvertálás MPEG2/AC3 formátumra (amely egy két órás videó esetén egy 2GHz-es gépen is 10-11 óra lehet!). A kész videót már csak rá kell varázsolni a DVD lemezre, erre a *DVDAuthor* programot tudjuk használni.

Ennek egy XML állomány kell, amelyben leírjuk, hogy mit szeretnénk készíteni:

```
<dvdauthor dest="/home/user/video/" jumppad="1">
  <vmgm />
  <titleset>
    <titles>
      <pgc>
        <vob file="/home/user/input.mpeg" />
      </pgc>
    </titles>
  </titleset>
</dvdauthor>
```

Ezek után egyszerűen csak el kell indítani:

```
$ dvdauthor -x dvdauth.xml
```

A megadott könyvtárban elkészül egy DVD adatstruktúra (AUDIO\_TS/VIDEO\_TS), amelyet tetszőleges íróprogrammal fel tudunk írni egy üres DVD lemezre. Ezzel el is készültünk, a kapott lemez bármilyen DVD lejátszóban működni fog.

## Összegzés

Remélem senkit nem riasztottam el ezzel a pár oldalas rövid ismertetővel, amelyből kiviláglik, hogy a FreeBSD egyelőre nem (és előreláthatólag a későbbiekben sem) való munkaállomásnak. Ugyan szinte minden feladat megoldható, amelyeket egy átlagos felhasználó igényel, ennek ellenére sok munkánk adódhat azokkal a beállításokkal, amelyeket egy átlagosnál barátságosabb Linux terjesztés automatikusan elvégez (nyomtatósorok beállítása, hálózati hozzáférések beállítása, stb). Nem véletlen, hogy a BSD rendszerek inkább a kiszolgálók területén nagyobb arányú jelenlétet mutatnak, sok nagyobb cég FreeBSD operációs rendszert használ webkiszolgálásra és levelezésre; illetve tűzfalnak, útválasztónak és forgalomszabályzóknak – a következő részekben megnézzük ennek okát.



**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/>