



A PPPD beállítása Linux alatt (2. rész)

Az első részben Tony a modem beállítási lehetőségeit mutatta be – most továbblép, és az Internet beállításával ismerteti meg bennünket.

Írásunk előző részében (lásd a Linuxvilág 2002. májusi számát) felvázoltuk, hogyan is kell beállítani a modemet. A cikk végére érve egy `/dev/modem` nevű közvetett hivatkozást kaptunk, amely a modem eszközfájljára mutatott. Megbizonyosodtunk arról is, hogy minden helyesen működik, vagyis összekapcsolódtunk a szolgáltatóval (igaz, PPP-kapcsolatot nem hoztunk létre). Ebben a cikkben továbblépünk, és megtudhatjuk, miképpen kapcsolódhatunk az Internethez. Javasolom, minden érdeklődő először az első cikket olvassa végig. Amennyiben ez lehetetlennek bizonyulna, legalább bizonyosodjunk meg afelől, hogy modem helyesen van-e beállítva, illetve hogy létezik-e a `/dev` könyvtár megfelelő eszközére mutató `/dev/modem`.

Hogy írásunknak hasznát vehessük, a szolgáltatónkhoz tartozó összes bejelentkezési adata szükségünk lesz, ideértve a tárcsázandó telefonszámot, a belépési nevet és jelszót, illetve egy érvényes DNS-kiszolgáló címét (bár ez ugyan elhagyható, mivel a szolgáltató segítségével önműködően lekérdezhető). Cikkünk feltételezi, hogy szolgáltatónk elfogadja a PAP azonosítási protokollt. A PAP egy olyan módszer, amellyel azonosítónk és jelszavunk a PPP-protokollon keresztül a szolgáltatóhoz juttatható; ezáltal a felhasználó megtakaríthatja azt a bonyolultabb (gyakran kézi vezérlést igénylő) bejelentkezési folyamatot, amelyet a szolgáltatók rendszerei régebben igényeltek. Az ISP-k (internetszolgáltatók) döntő többsége manapság már megköveteli a PAP használatát. Írásunk azt is feltételezi, hogy hagyományos modemmel és nem úgynevezett Winmodemmel rendelkezünk. A Winmodem beállítása is megoldható, de fásasztó és tárgyalása meghaladja e cikk kereteit.

A kapcsolat kiépítése: alapeszközök

Mindenekelőtt az internetkapcsolat beállításához rendszergazdaként kell bejelentkeznünk. A modemem keresztüli internetkapcsolat a PPP (point-to-point, azaz két pont közötti protokoll) segítségével valósítható meg. A normál TCP/IP-csomagokat PPP-csomagokba zárjuk (encapsulate), így azok a soros vonalon keresztül küldhetők (ugyanis arról van szó, hogy a TCP/IP-csomagok hálózati átvitelhez készültek, és átalakítás nélkül nem férnek el a soros kapcsolaton). Írásunkban feltételezzük továbbá, hogy gépünkön a következő programok megfelelően telepítve vannak:

- a PPP-kezelő rendszermagmodulok (minden általam ismert terjesztés rendszermagváltozata tartalmazza a PPP-modult, így emiatt valószínűleg nem kell aggódnunk);
- a PPPD program, amely a kapcsolat elindítása után alaphelyzetbe állítja a rendszermagmodulokat;
- a chatprogram, amely a kapcsolat létrehozásáért felelős;
- a minicom program, amely egy igen egyszerű terminálprogram, amit a modemmel való kapcsolattartáshoz fogunk használni.

A programok meglétének ellenőrzésére a `which` parancsot

használhatjuk. Ez a parancs ugyanis értesít bennünket, ha a keresett programfájl `$PATH` környezeti változóban felsorolt könyvtárak valamelyikében megtalálható:

```
which pppd
/usr/bin/pppd
which chat
/usr/bin/chat
which minicom
/usr/bin/minicom
```

Amennyiben nem rendelkezünk valamelyik programmal, meg kell szereznünk a megfelelő csomagot, és fel kell telepítenünk.

Kapcsolatáttekintés

Miután beállítottuk a modemet, hogyan csatlakozhatunk az Internethez? A PPPD (point-to-point protokoll démon) nevű programot kell használnunk (és beállítanunk). A PPPD futtatásakor a következők történnek (feltételezve, hogy tárcsázóprogramként a chat-et használjuk, és a PPPD helyesen van beállítva). A démon elindul, beállítja a soros kapu jellemzőit (sebesség stb.). Ezt követően a kapcsolat létrehozásához egy külső programot futtat le (a chat-et), amely a modemhez elküldi a kapcsolatteremtő utasítást (ATDT parancs, amelyet a szolgáltató telefonszáma követ). Ezután a CONNECT karaktersorozat megérkezéére vár a soros kapun. Ha idáig eljutottunk, a kapcsolat már felépült, és úgy is vehetjük, mintha számítógépünket a szolgáltató gépével egy soros kábel kötné össze. Mikor a chat befejezte futását, a vezérlést ismét a PPPD veszi át. Ha a kapcsolatot nem lehet kiépíteni, a PPPD kilép és hibát ad vissza, egyébként elkezd „beszélgetni” a vonal másik végén üldögélő PPP démonnal (ez az a PPP-kézfogás, amit korábban egy halom érthetetlen karakter formájában láthattunk), és végül kap egy IP-címet (abban az esetben, ha dinamikus IP-címmel rendelkezünk – a ford.). Általában ehhez a lépéshez szükség van azonosítónkra és a jelszavunkra (a bejelentkezési adat a PPP-kézfogás alatt küldődik el). A PPPD program ellenőrzi a rendszermag hálózati csatolófelületének létrejöttét és hogy a hálózati forgalom ide irányítódjon.

Néhány szó a bejelentkezésről

A két szükséges program – a PPPD és a chat – nem interaktív. Egyszerűen csak lefutnak, és minden üzenetet a Syslog rendszernapló démonnak küldenek el (`system log daemon`). A `syslogd` azután a megkapott üzeneteket a merevlemezre írja. Az üzeneteknek több típusa is van, és a különböző típusok más-más fájlokban tárolódnak. A tárolás pontos helyét a `syslogd` beállításai határozzák meg. Ezért nem árt, ha beállítjuk a `syslogd`-t, így száz százalékgig biztosak lehetünk benne, hogy a PPPD és chat démonoktól származó üzenetek valóban a merevlemezre kerülnek, illetve megtudjuk, hogy egyáltalán hova is kerülnek. A `syslogd` beállításfájlja a `/etc/syslog.conf`. Mindössze egyetlen kiegészítő

sort kell beírunk. Ehhez a következő parancsot gépeljük be:

```
vi /etc/syslog.conf
```

Természetesen tetszés szerinti szerkesztőt használhatunk (vi, Emacs, joe, pico stb.). Szúrjuk be a következő sort:

```
daemon.debug;* .info /var/log/ppp_article
```

Ne feledjük, hogy az info és a /var/log/ppp_article között egy TAB-nak kell állnia.

Ezután el kell érünk, hogy a syslogd démon észrevegye a beállításfájljában történt változtatást. Futtassuk a következő parancsot:

```
killall -HUP syslogd
```

Létre kell jönnie a /var/log/ppp_article fájlnek, a belsejében pedig egyetlen sornak kell állnia, amely a syslogd újraindítását jelzi. Ellenőrzéséhez a következő parancsot írjuk be:

```
cat /var/log/ppp_article
Aug 4 19:28:46 merc_linux syslogd 1.3-3:
↳restart.
```

A cat parancs helyett, amely a fájl csak olvassa, a -f kapcsolóval kiegészített tail-t is használhatjuk. Ez ugyanis folyamatosan olvassa a fájlt, és az újonnan bekerülő adatokat a képernyőre írja. Ez azt jelenti, hogy amint a syslogd, valamit a ppp_article fájlba ír, a tail azonnal megjeleníti a képernyőn:

```
tail -f /var/log/ppp_article
Aug 4 19:28:46 merc_linux syslogd 1.3-3:
↳restart.
```

Ettől kezdve a chat vagy a PPPD által rögzített adat önműködően megjelenik a képernyőn is. Erősen ajánlott, hogy ezt a konzolt folyamatosan nyitva tartsuk, és ellenőrizzük a rajta megjelenő üzeneteket, amikor csak szükséges.

Ismerjük meg a chatet!

Amint a cikk első részében kiderült, a soros kapcsolat kiépítéséhez az ATDT12345678 (természetesen szolgáltatónk telefonszámával) karaktereket kell a modemhez eljuttatni, majd a modemtől a CONNECT karaktersorozatát várjuk (ami a kapcsolat létrejöttékor fog megérkezni). A CONNECT-en kívüli egyéb üzeneteket vissza kell adni: BUSY, NO CARRIER, NO ANSWER stb. Az előző cikkben a minicom segítségével ezt a gyakorlatban is kipróbáltuk.

Igaz, hogy ezeket a minicom segítségével kézzel is végre tudjuk hajtani, de nem árt, ha van egy programunk, ami megteszi helyettünk. A programnak tudnia kell beszélni a modemmel, képesnek kell lennie adatot küldeni és egy adott karaktersorozatra várakozni. Természetesen létezik ilyen program, a neve chat. Próbáljuk meg például a következő parancsot lefuttatni:

```
chat ABORT "BUSY" "OK" "TRY" "THIS" "TESTING"
↳ "COMMAND"
```

Legyünk óvatosak, ugyanis mostantól a billentyűzet le van zárva, és nem lehet csak úgy kilépni a programból, még a CTRL-C leütésével sem. Gépeljük be az OK szót, amire a TRY

üzenet jelenik meg. Most gépeljük be, hogy THIS, ezután a TESTING szó jelenik meg a képernyőn. Végül üssük be a COMMAND szót, amire a program sikeresen kilép. Próbáljuk meg újra lefuttatni a parancsot: írjunk OK-t, amire ismét megjelenik a TRY szó. Ezután üssünk BUSY-t, és a program azonnal kilép. Ahogy már kitalálhattuk, a chat programot arra tervezték, hogy karaktersorozatokat várjon, és valami mást írjon ki válaszul. A két első szó – ABORT BUSY – különleges jelentésű, és arra utasítja a chat-et, hogy lépjen ki, ha a végrehajtás során bármikor a BUSY szóval találkozik. Ha valami nem működik, a chat parancsot a -v kapcsolóval is lefuttathatjuk:

```
chat -v ABORT "BUSY" "OK" "TRY" "THIS"
↳ "TESTING" "COMMAND"
```

A -v kapcsoló a chat-et beszédre állítja, így mindig elmondja nekünk, hogy éppen pontosan mit is csinál vagy mit vár és így tovább. Természetesen minden hibakereső üzenet a /var/log/ppp_article-re fog kerülni, ha követtük a syslogd-nál korábban leírt utasításokat. Vizsgáljunk meg egy másik chat parancsot:

```
chat ABORT "BUSY" "" "AT" "OK" "ATDT93355100"
↳ "CONNECT"
```

Ahogy valószínűleg mindenki kitalálta, úgy kell utasításokat kiadnunk, mintha mi lennénk a modem – amennyiben azt szeretnénk, hogy a chat sikeresen kilépjen. Egy AT karaktersorozat fog küldeni nekünk, amire OK begépelésével kell válaszolnunk. Erre válaszul az ATDT93355100 karaktersorozatot küldi nekünk, majd várakozik, amíg be nem gépeljük a CONNECT szót. Ezután kilép. Ez feltehetően a legtöbb olvasó számára ismerősen hangzik, pontosan ez kell nekünk az ISP-csatlakozáshoz, csak rá kell vennünk a chat-et, hogy a billentyűzet helyett a modemhez beszéljen. Én szolgáltatómhoz a következő parancsot használom:

```
chat ABORT BUSY ABORT "NO CARRIER"
↳TIMEOUT 120 "" AT OK ATDT94310999 CONNECT
```

Ez elég egyszerű, és igazság szerint sokkal jobban is lehetne csinálni, de nekem megfelel és elégedett vagyok vele. Megtekinthetjük a chat súgó oldalát is (egyszerűen gépeljük be: man chat), és elolvashatjuk a felajánlott lehetőségeket; később esetleg megváltoztathatjuk a kapcsolatteremtő parancsfájlt, hogy kihasználja a chat által felajánlott csinos kis megoldásokat. Következő lépésként egy parancsfájlt kell készítenünk, amely tartalmazza az imént megírt chat parancsot. A fájlt a /etc/ppp könyvtárba helyezzük, és a chat-connect nevet adjuk neki. Létrehozásához gépeljük be a következő parancsot:

```
vi /etc/ppp/chat-connect
```

(természetesen bármilyen más szerkesztőt is használhatunk, ha a vi-t nem kedveljük). A parancsfájlnak a következőképpen kell kinéznie:

```
#!/bin/sh
chat ABORT BUSY ABORT "NO CARRIER"
↳TIMEOUT 120 "" AT OK ATDT94310999 CONNECT
```

A 94310999 helyére a saját ISP-szolgáltatónk számát kell írunk. Ezután mentjük, és lépünk ki a szerkesztőből.

A parancsfájlt futtathatóvá is kell tennünk, ezt érhetjük el a `chmod` parancssal:

```
chmod +x /etc/ppp/chat-connect
```

Győződjünk meg parancsfájlunk működőképességéről a következő parancs futtatásával:

```
/etc/ppp/chat-connect
```

Ha működik, ismét egy lépéssel közelebb kerültünk az internetkapcsolathoz. Tulajdonképpen kifejezetten közel vagyunk a célhoz. Mindössze arra van szükség, hogy a PPPD-t a megfelelő kapcsolókkal futtassuk.

Ismerjük meg a PPPD-t!

Most már elkezdhetünk magával a PPPD beállításával foglalkozni. Az ide vonatkozó fájlok: `/etc/ppp/options`, `/etc/ppp/chap-secrets`, `/etc/ppp/pap-secrets` és `/etc/ppp/peers`. Az `options` fájl a PPPD alapértelmezett beállításait sorolja fel. Egyelőre induljunk el úgy, hogy a `/etc/ppp` könyvtárban található `options` fájl teljesen üres legyen; kedvenc szerkesztőnkkel mindent töröljünk ki belőle. Ha ezt nem szeretnénk megtenni, a tartalmat megjegyzéssé is változtathatjuk, amennyiben a sorok elé beszurjuk a `#` jelet. Nagyon fontos, hogy üres `options` fájljal indítsunk, mivel csak így lehetünk benne biztosak, hogy tiszta lappal indulunk. Első lépésként nézzük meg, hogy az imént elkészített chat parancsfájlunk valós helyzetben is jól működik-e. Ehhez a PPPD-t kell elindítanunk néhány kapcsolóval:

```
pppd /dev/modem 38400 modem lock connect
↳ /etc/ppp/chat-connect
```

A kapcsolókat a PPPD-nek tetszőleges sorrendben megadhatjuk. A `/dev/modem` kapcsoló azt a soros kaput jelképezi, ahová a modemet kötöttük (mint tudjuk, ez közvetett hivatkozás a valódi `ttyS`-eszközre). A `modem` kapcsoló arra figyelmezteti a PPPD démont, hogy modemes kapcsolaton keresztül fog működni, nem pedig egy egyszerű, minket és a szolgáltatót összekötő soros kábelben. A `word lock` azt jelenti, hogy a modemet – amíg használjuk – le kell zárni (ha valaki nem tudja, hogy ez mit jelent, ne aggódjon; csak arról van szó, hogy így garantálható, hogy amíg a kapcsolat él, más program ne férhessen a modemhez). Az utolsó lehetőség a `connect`, amely `/etc/ppp/chat-connect` kapcsolóval együtt értelmezendő, és azt mondja meg a PPPD-nek, hogy melyik programot kell elindítania a szám tárcsázásához és az internetszolgáltatóhoz való kapcsolódáshoz – a mi esetünkben ez most a cikk előző fejezetében megírt chat parancsfájl lesz.

Ha a dolgok nem működnének, chat parancsfájlunkat kiegészíthetjük `-v` kapcsolóval, majd újból próbálkozva figyeljük a naplófájlokat – ezen a ponton még viszonylag könnyen megoldhatjuk a nehézségeket. Ha minden jól megy, láthatjuk, amint modemünk csatlakozik, és hallhatjuk, amint a szokásos sipolási szertartás lezajlik. Most már csak egyetlen lépés választ el bennünket az internetkapcsolattól. Szerkesszük át a `/etc/pap-secrets` fájlt, és jelszavunkat az alábbihoz hasonló sorral adjuk hozzá:

```
felhasznal i_neved * jelszavad
```

Ne feledjük, minden szó között egy TAB-nak kell lennie. Máris készen állunk a nagy próbára, a valódi kapcsolatfelvételre. Próbáljuk ki a következő parancsot:

```
pppd /dev/modem 38400 modem lock connect
↳ /etc/ppp/chat-connect user
↳ felhasznal i_neved defaultroute
```

Az egyetlen új kapcsoló a `user` (amit a `/etc/ppp/pap-secrets` fájlban is szereplő felhasználói nevünk követ), illetve a `defaultroute` lehetőség. Ez utóbbi teszi lehetővé, hogy az Internetre igyekvő csomagok alapértelmezés szerint ezt a kapcsolatunkat használják. Ezzel a lehetőséggel a kapcsolat létrehozása után a PPPD be fogja állítani a megfelelő routing-tábla bejegyzéseket. A naplófájlban ilyesféle üzenetet kell látnunk:

```
Aug 4 16:12:23 merc_linux pppd[4430]:
↳ local IP address 94.232.195.174
Aug 4 16:12:23 merc_linux pppd[4430]:
↳ remote IP address 194.232.195.4*
```

Amennyiben nem így történt, a debug kapcsolóval lefuttathatjuk a PPPD-t, és megnézhetjük a naplófájlt (azaz a `/var/log/ppp_article-t`), hogy megvizsgáljuk, mi is történt:

```
pppd /dev/modem 38400 modem lock connect
↳ /etc/ppp/chat-connect user
↳ felhasznal i_neved defaultroute debug
```

Ha minden működött, gratulálok, immár kapcsolatban állsz az Internettel. Ne feledjük, a lecsatlakozáshoz csak ennyit kell beírunk:

```
killall pppd
```

A kapcsolat kipróbálása

A következő lépés, hogy megnézzük, kapcsolatunk valóban működik-e. Ha meg akarjuk tudni, hogy él-e a kapcsolat vagy sem, a legegyszerűbb, ha az `ifconfig` parancsot futtatjuk le (lásd az 1. listát a 34. CD Magazin/PPPD könyvtárban). Ez a parancs a rendszermag jelenleg működő hálózati csatlóit mutatja. Nálam például létezik egy `lo`, azaz normál hurokeszköz (loopback) csatlófelület, amit akkor használok, ha magamhoz akarok csatlakozni, és egy `ppp0`, ami a modem PPP-csatlófelülete. Ha látni kívánjuk, hogy tényleg kitalálunk-e az Internetre (működik-e a `routing`), futtassuk le a `traceroute` parancsot, amit valamilyen IP-cím követ. Egyelőre használjuk a `-n` kapcsolót, hogy a DNS-névfeloldást (name resolution) kikapcsoljuk (ezt ugyanis még nem állítottuk be).

Például:

```
traceroute -n 198.182.196.56
traceroute to 198.182.196.56 (198.182.196.56),
30 hops max, 38 byte packets

 1  194.232.195.4 (194.232.195.4) 181.518 ms
   139.473 ms 149.822 ms
 2  194.232.195.1 (194.232.195.1) 129.540 ms
   139.739 ms 139.821 ms
...
19 207.245.34.122 (207.245.34.122) 479.696 ms
   479.653 ms *
20 198.182.196.56 (198.182.196.56) 489.711 ms
   479.644 ms 479.874 ms
```

A 198.182.196.56 IP a <http://www.linux.org> névhez tartozik. A `traceroute` mutatja meg nekünk azt az utat, amin

az általunk küldött csomagok az Interneten utaznak. Itt az ideje, hogy a rendszernek a `/etc/resolv.conf` fájl segítségével megadjuk DNS-ünk IP-címét. Az én `resolv.conf` állományom a következőképpen néz ki:

```
nameserver 203.14.168.3
nameserver 202.0.185.226
```

Néhány ISP nem ad DNS-kiszolgálócímet, mivel a számítógép a PPP-kézfogás befejezésekor önműködően kap egyet. Ha ez a helyzet, egyszerűen lépünk ki, majd a PPPD futtatásával a usepeerdns kapcsolóval kiegészítve indítsuk újra a kapcsolatot:

```
pppd /dev/modem 38400 modem lock connect
↳ /etc/ppp/chat-connect user
↳ felhasznál_i_neved defaultroute usepeerdns
```

Rögtön meg is nézhetjük, hogy működik-e a DNS, például a Telnet programmal. A Telnetet most csak arra használjuk, hogy lássuk, képes-e a rendszer a `www.linux.org` nevet IP-címmé fordítani.

```
telnet www.linux.org 80
Trying 198.182.196.56...
Connected to www.linux.org.
Escape character is ^].
```

Működik! Már indíthatjuk is a böngészőnket (Netscape, Mozilla, Opera, Galeon, Lynx stb.), és kedvünkre nézegethetünk a Hálózaton.

Egy kis utómunka

Végre minden jól működik: az internetkapcsolat él, és amikor csak akarunk, csatlakozni tudunk az Internetre. Akad azonban még lehetőség egy kis további fejlesztésre. Az első dolog, amit érdemes megcsinálni, a soros kapcsolat sebességének növelése, majd ellenőrzése. Ehhez a PPPD parancsokban a 38400-as számot 11 5200-ra kell cserélnünk.

Néhány hét után valószínűleg fel fog tűnni, hogy rengeteg kapcsolót kell megadnunk a PPPD parancshoz. Tulajdonképpen minden egyes kapcsolathoz az egész sort be kell gépelnünk:

```
pppd /dev/modem 115200 modem lock connect
↳ /etc/ppp/chat-connect user
↳ your_username_here defaultroute
```

Jó hír, hogy ezeket a kapcsolókat természetesen be tudjuk rakni egy beállításfájlba, mégpedig a `/etc/ppp/options` állományba. Így esetünkben az `options` fájl a következőképpen fog kinézni:

```
/dev/modem
115200
modem
lock
connect /etc/ppp/chat-connect
user your_username_here
defaultroute
```

Ebben a fájlban a kapcsolók sorrendje nem igazán számít. Ettől kezdve egyszerűen a `pppd` parancs begépelésével csatlakozhatunk az Internethez. Mi történik, ha több szolgáltatóhoz is szeretnénk csatlakozni? Ebben az esetben több `options` fájl kell

készítenünk, amelyeket aztán a `/etc/ppp/peers` könyvtárba helyezünk. Az alábbi kimeneten látható, hogy néz ki az én `peers` könyvtáram:

```
ls -l /etc/ppp/peers
total 4
-rw-r--r-- 1 root root 197 Aug  4 15:41
↳ main_net
-rw-r--r-- 1 root root 189 Mar 11 2000
↳ primus
```

A `/etc/ppp/options` állományom üres; amikor a PPPD-t futtatom, mindig ezt írom:

```
pppd call main_net
```

Ilyenkor a `/etc/ppp/peers/main_net` és a `/etc/ppp/options` fájljaim is értelmezve lesznek (melyek közül a második történetesen üres). Ha valamilyen okból kifolyólag a fő szolgáltatóm (Main Net) leállna, még mindig használhatom Primus időkorlátos fiókomat.

A legjobb, amit ezen a ponton tehetünk, hogy elolvassuk a PPPD kézikönyvoldalait (egyszerűen üssük be: `man pppd`) és megnézzük, hogy a sok kapcsoló közt akad-e olyan, ami esetleg a kapcsolatunknak hasznára lehetne. A 2. listában (34. CD Magazin/PPPD könyvtár) egy igen gazdag `options` fájl találok, amit *Pancrazio De Mauro* barátom, egy igazi Linux-guru írt. Vajon, te jobban meg tudod csinálni?

Összegzés

A folyamat talán kicsit félelmetesnek tűnhet; a Linux alatti internetkapcsolat kiépítéséhez szükséges tudásszint elképesztő, különösen a Windows *Remote access interface* egyszerűségével összehasonlítva; ennek fényében kérdéses, hogy megéri-e egyáltalán mindent kézzel beállítani. (Azért tegyük hozzá, hogy a legembertelenebbnek nevezett Debian alatt egy kezdőnek is mindössze fél percébe kerül belőni a PPP-t is – a `pppconf` programmal. Szóval, aki korbáccsal veri magát, az ne sírjon, ha fáj – a fordító).

Véleményem szerint két nagy előnye van annak, ha mindent kézzel állítunk be: az egyik, hogy végig tudunk (sőt végig kell) menni sok PPPD-kapcsolón, ami kapcsolatunkat hatékonyabbá teheti. A második, hogy mostantól, ha az internetelérést grafikus felületen állítjuk be, tudni fogjuk, mi is történik, és sikerrel próbálkozhatunk a helyrehozással, ha az önműködő megoldással nem jutunk dűlőre.

Mielőtt befejeznénk, szeretnék rámutatni, hogy létezik egy parancssoros program (nem GUI), amelyik önműködően végrehajtja a cikkben leírt lépéseket, (megkeresi a modemet, a megfelelő kapcsolók használatával összekapcsolódik a szolgáltatóval stb.). A programot `wvdial` néven találhatjuk meg a <http://www.worldvisions.ca/wvdial/index.html> címen.

Linux Journal március, 95. szám



Tony Mobily

(merc@mobily.com) a Login olasz számítástechnikai magazin szakmai szerkesztője. LCI (Linux Certification Instructor,

<http://www.linuxcertification.com>)

képesítéssel rendelkezik, angol, olasz, C, Perl

és néhány más nyelvet is használ.