

Nokia-mobiltelefonok Linux alatt – 2. rész

Ígéretet tettem rá, hogy folytatom ezt a témát, még-hozzá a kiszolgálófelügyelet témakörével. Nézzük, mi haszna lehet egy Linuxra kötött telefonnak a kiszolgálókönyezetben! Már egy kiszolgáló üzemeltetése esetén is érdemes folyamatosan figyelemmel kísérmünk gépünket, hiszen egy behatolást vagy kiszolgálóleállást általában olyan események szoktak megelőzni, amelyekből a legtöbb esetben ki lehet találni, hogy baj lesz.

```
Welcome to minicom 1.83.1
OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
Compiled on Nov 21 2001, 00:35:58.
Press CTRL-A Z for help on special keys

NO CARRIER
ATZ
OK
atdt .....
CONNECT 57600/V34/LAPM/V42BIS/33600:TX/33600:RX
Debian GNU/Linux ttyS0 57600 (57600)
.
mail!login: guska
Password:

Linux timeout 2.4.17 #19 2002. máj. 25., hétfő, 21.23.36 CET i686 unknown
Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program
are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

1.1 All
```

Ez természetesen csak akkor lenne lehetséges, ha a nap 24 órájában a konzol előtt ülve figyelnék a naplóbejegyzéseket. Amennyiben előre gondolkodunk, és megfelelő eszközt választunk a naplók elemzésére, hamar értesülhetünk arról, hogy a lemez be fog telni, vagy valaki behatolással próbálkozik. Szinte mindegyik elemzőprogram lehetőséget kínál rá, hogy a figyelmeztetést levélben juttassa el hozzánk, vagy az adatokat egy másik programnak adja át. Ebben az esetben mi sem egyszerűbb, mint ha az előző cikkben leírtak alapján a Gnokiit SMS-küldésre állítjuk be, valamint rávesszük a naplózó programot, hogy a Gnokiin keresztül küldjön nekünk értesítést. Ha a figyelőprogram nem ad módot az SMS közvetlen küldésére, legyünk ötletesek! Az adatokat akár egy pár soros héjprogram megalkotásával átadhatjuk, például az alábbi módon:

- A figyelőprogram csak elektronikus levelet tud egy megadott címre küldeni – semmi baj, írjunk egy Perl-programot, amelyet meghívunk a Procmailból, így a levél tartalma máris a kezünkben van, és egyszerűen át tudjuk adni a Gnokiinak:

Procmail:

```
:0 c | /home/guska/sms-kuldo.pl
```

Ebben az esetben a levél teljes tartalmát átadtuk az *sms-kuldo.pl* programnak, amely esetleg 160 karakterre darabolja fel, és a *gnokii --sendsms* segítségével eljuttatja a telefonunkra, továbbá egy fájlba természetesen feljegyzést készít az üzenet küldéséről, hogy tudjuk, mennyi SMS-t küldtünk.

- A figyelőprogram csak egy meghatározott napló-fájlba tud bejegyzést írni – ugyancsak semmi gond: vagy írunk egy olyan programot, amely az időzítőből futtatva akár másodpercenként megnézi, történt-e bejegyzés, és ha igen, meghívja a Gnokiit, vagy egy kicsit bonyolultabb programot készítünk, amely a fájl folyamatosan olvassa.

Gondos figyelemmel válasszuk meg a programot, amely a figyelmeztetéseket létrehozza, hiszen előfordulhat, hogy egy túlbuzgó felhasználó POP3-kéréseit a gép DoS-támadásnak veszi, és fölöslegesen ébreszt fel minket. Arra is nagyon ügyeljünk, hogy egy folyamatos hibánál – ha már értesültünk róla (például az egyik kiszolgáló leállt) – a gép ne küldjön másodpercenként üzenetet, hanem időintervallumok legyenek. Például a *mon* nevű program lehetőséget nyújt, hogy amennyiben a gép elérhetetlen, késleltetést adjunk meg. Ha nem érhető el, küldjön SMS-t, majd ötpercenként próbálkozzon, de utána már csak akkor üzenjen SMS-sel, amikor a gép újra elérhető. Ez azért is fontos, mert ha jól állítjuk be, a gép csak abban az esetben küld értesítést, amikor felhasználói beavatkozásra van szükség, és nem ér rá reggelig. Ebben az esetben bátran úgy állíthatjuk be saját mobilunkat, hogyha a kiszolgálóra kötött telefonszámról az éjszaka kellős közepén SMS érkezik, hangosan csipogjon, egyéb esetekben csak rezegjen, hiszen csak akkor van értelme az egésznek, ha megbízhatunk benne. Pont ezért nem megbízható az elektronikus levéllel történő értesítés, hiszen az több levelezőkiszolgálón is átmegy, ahol meghamisíthatják, esetleg lehallgathatják. Igaz, már SMS-t is lehet hamisítani, de pontosan ismerni kell hozzá a küldő és fogadó számát, valamint az üzenetek felépítését. Természetesen egyik sem nyújt százszázalékos biztonságot a hamisítással szemben. A kiszolgáló soros kapujára kötött telefonnak további nagyon hasznos felhasználása – ha beépített modemmel rendelkezik –, hogy kihasználjuk a modem adta lehetőségeket, és szükség esetén akár távolról is beléphetünk a gépre. Jókor jöhet ez a szolgáltatás, ha például:

- leálltak a távoli elérési lehetőségek: SSH, Telnet;
- egy félresikerült tűzfalbeállítással kitiltottuk magunkat;
- a kiszolgáló *init 0* szintre jutott – például egy lemezhiba miatt;
- a kiszolgáló elérhetetlen, mivel az Internet megállt.

Ebben az esetben az *mgetty* csomag lehet a segítségünkre. Ennek a remek programnak számos felhasználási módja van:

- konzolként,
- adatbehívópontként és
- faxbehívópontként használhatjuk.

Ha az első lehetőséget vesszük figyelembe, akkor kiszolgálónk egy másik gépről történő felügyeletét egy soros kábel árából megoldhatjuk. Ez csak akkor hasznos, ha



egy szolgáltatónál legalább két gépet egymás közelében helyezünk el. Ebben az esetben, ha mondjuk a tűzfalbeállításban valamit elszúrtunk, bejelentkezhünk a másik gépre, és egy minicom (terminálprogram) segítségével átjelentkezünk a másik konzoljára.

A fenti példa megvalósításához a következőket tegyük: a `/etc/inittab` állományba írjuk be a következő sort:

```
S0:12345:respawn:/sbin/mgetty
-s 57600 ttyS0
```

Ahol az `S0` a jelenlegi soros kapu, amire a kábelt kötjük, tehát `COM1` esetén `S0`, kettő esetén `S1` és a többi, a `-s` kapcsoló a csatlakozási sebesség csúcsértékét adja meg, a `ttyS0` ugyancsak a kábel helyét határozza meg.

A `/etc/mgetty/mgetty.config` állományba a következőket írjuk:

```
port ttyS0 # Itt ugyanazt a soros
kapuszámot adjuk, mint az inittabba.
direct y
speed 57600 # Itt ugyanazt a
sebességet adjuk, mint az inittabba.
toggle-dtr n
```

Egy `init q` parancs után már működik is. Ellenőrizni a `/var/log/mgetty/ttyS0.log` állományban tudjuk, vagy a `ps aux | grep mgetty` paranccsal. Ezek után a másik gépről egy minicom program segítségével rá tudunk jelentkezni az `mgetty`-re, ahol is egy egyszerű naplózóablakot kapunk.

Amennyiben mobilról akarunk a gépre jelentkezni, egy kábelre lesz szükségünk, amelyet ugyancsak az egyik soros kapura „akasztunk”. Az összes többi beállítás az előzőhöz hasonló, valamint az `inittab`-ban a `-n` kapcsolóra van szükségünk, mivel ez határozza meg, hogy a gép hány csörgés után vegye fel a telefont. A soros kapu sebességét `9600`-ra vagy `19200`-ra állítsuk (ez a szám a kapu és a telefon közti sebességet árulja el).

```
S0:12345:respawn:/sbin/mgetty
-s 19200 -n 2 tyS0
```

A telefonos betárcsázást az első megoldással kombinálhatjuk, ilyenkor egy másik gépről és telefonról is lehetőségünk lesz bejelentkezni.

Az egyszerű kábeles bejelentkezés akkor is hasznos lehet, ha se másik gép, se telefon nincs, amire rákössük, mivel több szolgáltatónál előfordul, hogy bár lehetőség van hozzáférni a gép konzoljához, de éppen nincs AT (rég) típusú billentyűzetük vagy átalakítójuk, csak PS/2. Ilyenkor vagy magunkkal viszünk egyet, vagy ha ott derül fény a gondra, szerezni próbálunk egyet. Az esetek többségében az AT-billentyűzet beszerzése 1–2 óránál több időt szokott igénybe venni. Helyette elővesszük soros kábelünket és egy PDA-t vagy notebookot, amelyen egy terminálprogrammal be tudunk jelentkezni a gépre – billentyűzet és monitor nélkül is.

Régen egy kiszolgálóteremben fordult elő, hogy lustaságból nem állítottam be az `mgetty`-bejelentkezést. Ment is szépen sokáig, amíg egy rendszermagfrissítés miatt a gépet újra nem kellett indítani. A gép újraindult, és

ahogyan várni lehetett, egy apró gond miatt nem indult el többször. Kirohantam a helyszínre, hogy megnézzem, mi történt. A régi rendszermagot nem lehetett elindítani, mert se monitor, se régi típusú billentyűzet nem állt



rendelkezésre.

Megközelítőleg kétórás futkosás után egy közelben lakó ismerős-

tól elkoboztam a billentyűzetét, és vakon begépeltem az `mgetty` beállítását. Ezek után már egy konzolról be tudtam jelentkezni a gépre és megoldottam a rendszermag-fordítási hibát, amely abból állt, hogy az `ext2` fájlrendszer-támogatást nem fordítottam bele, így természetesen `ext2`-alapú rendszerem működésképtelen lett. Utolsó lépcsőként megpörgettem a Lammer-számlálót, mivel alapszintű probléma miatt fáradtam fölöslegesen. Azóta nem felejttem el a soros kapun történő kapcsolattartási lehetőséget a rendszerbetöltő programba is belerakni, hiszen ha se monitor, se billentyűzet nincs kéznél, fontos, hogy már az indításkezelő program (`LILO` vagy `Grub`) szintjén kapcsolatot tudjunk tartani (például kiválaszthatjuk, hogy melyik rendszermag töltődjön be).

Ehhez a `/etc/lilo.conf`-ba a következőket írjuk:

```
serial=0,19200n8 # 0 a soros kapu
számát, 19200 a kapcsolat sebessége,
n8 az adatátviteli t pusa
```

Majd futtassuk a `lilo` parancsot. A következő indításkor a terminálprogram bekapcsolása után a PDA-n, illetve a notebookon figyelemmel kísérhetjük a rendszer indítását, és szükség esetén beavatkozhatunk.



Varga S. Csaba

(guska@guska.hu) Az 1.1-es Slackware óta linuxozik. Kedvteléseinek közé tartozik a fotózás és Linux telepítése PDA-kra. Legszívesebben a Gerecsében túrázik a barátaival.

Kapcsolódó címek

Lammer-számláló ➔ <http://dawn.elte.hu/~raas/lc.html>
Minicom ➔ <http://www.clinet.fi/~walker/minicom.html>
Mgetty ➔ <ftp://alpha.greenie.net/pub/mgetty/source/>