Bluetooth – Vezetékek nélkül, szabadon

Manapság sokféle vezetékmentes szabványt használnak (IrDA, Bluetooth, Wifi) és jó pár vár (WiMax) az elterjedésre. Mostanság a Bluetooth-t egyre többen kezdik felfedezni, így én is. Hogy másoknak ne kelljen az ösvényt újra és újra kitaposni, megosztom a Linux és Bluetooth összeházasításával kapcsolatos tapasztalataimat.

A múlt

Mielőtt vadul belevágnék a közepébe, szeretnék rávilágítani, miért pont ***Bluetooth*** a szabvány neve? A tizedik században élt egy dán király: ***Harald Blatand*** (angolul: ***Harold Bluetooth***). Ő volt az aki egyesítette a hadviselő feleket a mai ***Norvégia***, ***Svédország*** és ***Dánia*** területén. Különböző – kultúrájú és gazdaságú – népek békés kommunikációja. A ***Bluetooth*** szabvány is valami ilyet csinál. Összefog több különböző eszközt (mobiltelefon, számítógép, headset, GPS vevő, stb.)

A szabványról

Hogy megértsük a ***Bluetooth*** működését, nem árt, ha ismerünk pár adatot. A szabvány a 2,45 GHz körüli sávokat használja. Ez azt jelenti a gyakorlatban, hogy a falakat nem nagyon szereti, illetve ***wifi***-vel „szennyezettebb" területen gyengülhet a vétel, de mobiltelefon és headset közé ideálisabb, mint a vezeték. Létezik belőle ***Class 1***-es (1 milliwatt adóteljesítmény), ***Class 2***-es (10 milliwatt adóteljesítmény), valamint ***Class 3***-as (100 milliwatt adóteljesítmény). A ***Class 1*** és ***2*** körülbelül 10 méteres hatósugarat biztosít nyílt terepen, míg a Class 3-as akár 100 métert, de ez utóbbi elég ritka.
A ***Bluetooth***-t nem a ***LAN (Local Area Network)*** felváltására tervezték, hiszen csak 1 megabites sávszélességet tud nyújtani (kb. 700 kbit-es letöltés aszimmetrikus módban és körülbelül 500 kbit-es szimmetrikus módban), illetve egy master (mester) eszközhöz maximum 7 slave (szolga) eszköz tud csatlakozni. A ***Bluetooth*** sokkal inkább ***PAN (Personal Area Network)***, vagyis személyi hálózat.

Párosan szép az élet

A ***Bluetooth*** eszközök csak akkor tudnak egymással kommunikálni, ha úgymond párosítják őket. Ez olyan, mint amikor a való életben két üzleti partner névjegykártyát cserél.
A ***Bluetooth***-nál kölcsönösen azonosítót cserélnek, így biztosítva, hogy két kijelöl eszköz csak akkor ismerje egymást, ha mindkettő tulajdonosa jóváhagyta azt.
Jó tudni, hogy sok eszköz – például a ***Bluetooth headset***-ek nagy része – képes több mester eszközt kezelni, így nem kell folyton törölni a már meglévő párosítást, valahányszor változik az eszköz (számítógép, mobiltelefon, ***PDA***). Viszont ha egy ***headset***-et – vagy valami egy másik eszközt – már használ valami, úgy az a többi eszköz számára láthatatlan, még akkor is, ha egyébként párosítva vannak.

A kékfogú pingvin

Linux alatt már elég régóta támogatottak a ***Bluetooth*** eszközök, de én személy szerint igazán csak a 2.6.13.4-es kerneltől kezdve próbálgattam.
A linuxos ***Bluetooth*** tapasztalataim szerint az alábbiakra képes, később be fogom mutatni, hogyan:

- Internetmegosztás (laptophoz, ***PDA***-hoz ideális)
- Fájlátvitel (például mobiltelefonról vagy mobiltelefonra)
- Hangkártya (ha nincs hangkártya a gépben)
- Betárcsázós internet (végszükség esetére)

Persze többre is képes lehet, de ha átlátjuk az alapvető dolgokat, akkor a fenti négy esetből könnyen kitalálhatjuk egy kis utánajárással a dolgokat.

Forogjon a kernel

A ***Linux*** általában kezeli a mai ***USB***-s ***Bluetooth*** kulcsokat. Jelen tesztben egy Class 1-es ***X-Micro*** típusút használtam. Ahhoz, hogy működjön, nyilván az alaplapnak megfelelő ***USB*** vezérlőt le kell fordítani, legyen az ***OHCI***, ***UHCI*** vagy ***EHCI***. Ha van működő ***USB*** alrendszerünk, akkor egy `lsusb` paranccsal megtudhatjuk, milyen eszközünk van konkrétan. Nálam például egy ilyen:

```
...
Bus 002 Device 002: ID
➥0a12:0001 Cambridge Silicon
➥Radio, Ltd Bluetooth Dongle
➥(HCI mode)
...
```

Általánosságban elmondható azonban, hogy az alábbi részek szükségesek a működő ***Bluetooth***-hoz:

```
CONFIG_BT=m
CONFIG_BT_L2CAP=m
```


```
CONFIG_BT_SCO=m
CONFIG_BT_RFCOMM=m
CONFIG_BT_RFCOMM_TTY=y
CONFIG_BT_BNEP=m
CONFIG_BT_HIDP=m
```

A fentiekhez rövid magyarázat: az ***RFCOMM*** a betárcsázós internethez és a fájlok áttöltéséhez szükséges, míg a ***BNEP*** a helyi hálózatért felelős. Az eszközmeghajtóknál pedig érdemes csak azt lefordítani, amivel rendelkezünk.
A 2.6-os kernel esetén, amennyiben a `make menuconfig` segítségével konfiguráljuk a kernelt, úgy a ***Bluetooth*** a ***Networking*** menüben található. Innen érdemes mindent modulba fordítani, hátha kell később. Persze vannak dolgok, amiket bele kell még fordítani, például a betárcsázós internethez a ppp támogatás, de az nem létfontosságú.
Debian alatt a `bluez-utils` csomagot szükséges feltelepíteni annak minden függőségével együtt. Amennyiben a használt disztribúció nem rendelkezik ezzel a csomaggal, úgy azt a ***Bluez*** projekt honlapjáról tölthetjük le.

Internet továbbosztása

A ***Bluetooth*** például kiválóan alkalmas arra, hogy az asztali számítógépünk segítségével például a ***PDA***-ra (vagy akár laptopra) továbbítsuk az internetet. Ehhez nem kell mást tennünk, mint telepíteni a `bluez-pin`, `bluez-utils`, `libbluetooth1` csomagokat.
A csomagok feltelepítése után a `/etc/bluetooth` könyvtárban találunk pár fájlt. Ilyen például a `hcid.conf` és a `pin` fájl. A minimális `hcid.conf` valahogy így néz ki:

```
# HCI daemon configuration
# file.
# HCId options
options {
        autoinit yes;
        security auto;
        pairing multi;
        pin_helper
        ➥/usr/bin/bluez-pin;
}
# Default settings for HCI
# devices
device {
        name "%h-%d";
        class 0x3e0100;
        iscan enable; pscan
        ➥enable;
        lm accept;
        lp
rswitch,hold,sniff,park;
}
```

A `pin` fájl pedig az adott ***Bluetooth*** eszköz (például ***USB kulcs***) általunk választott ***PIN kódját*** tartalmazza. Ezt fogja a másik eszköz kérni párosításkor.
Nos a ***PAN*** (személyes hálózat) felépítéséhez szükségünk van a `pand` kiszolgálóra, amely `bluez-utils` csomagban található.

```
pand --listen --role NAP --
➥master -autozap
pand --connect
➥xx:xx:xx:xx:xx:xx --service
➥NAP -autozap
```

Az `xx`-ek helyére a ***Bluetooth*** adapterünk ***MAC*** címét várja a program, ami vagy szerepel az eszköz csomagolásán, vagy pedig a `hciconfig` paranccsal kérdezhetjük le. Most a másik eszközzel például a ***PDA***-val csatlakozzunk a ***Bluetooth*** adapterhez.
Ha ez sikerült, akkor utána már a `bnep0` hálózati csatoló is megjelenik a hálózati kártyák listájában (`ifconfig -a`), így címet is tudunk hozzárendelni, legyen mondjuk:

```
ifconfig bnep0 192.168.2.1
➥netmask 255.255.255.0 up
```

Fontos, hogy ne abból az alhálózatból válasszunk címet, amiben a számítógépünk hálózati csatolója van.
Ha ez megvan, a ***PDA***-n is vegyünk fel egy hasonló ***IP*** címet (ugyanabból az alhálózatból), legyen ez most a 192.168.2.2. Ezután már pingelhető a két eszköz oda-vissza. Persze még nem lehet teljes az öröm, hiszen a ***PDA*** még nem lát a nagyvilágba. Ehhez szükséges, hogy az ***iptables*** megfelelően legyen benne a kernelben – tudjon ***NAT***-olni (***Network Address Translation***) illetve támogassa a ***masquerade***-ot és a ***forward***-ot. A következő parancsokkal tudunk kapcsolatot kialakítani:

```
iptables -t nat -A POSTROUTING
➥-s 192.168.2.0/24 -o eth0
➥-j MASQUERADE
echo 1 > /proc/sys/net/
➥ipv4/ip_forward
```

A két sorhoz egy kis magyarázat: az első sorban megmondjuk, hogy a 192.168.2.0-ás alhálózatból érkező kéréseket továbbítsa az ***eth0*** hálózati csatolóra – ezen jön be nálam az internet. A második sorban pedig az alapvető csomagtovábbítást engedélyezem a két interfész között.

Állományok átvitele

Az ***OBEX (OBject EXchange)*** szabvány lehetővé teszi, hogy akár ***Bluetooth***, akár infra kapcsolat segítségével állományokat töltsünk fel a telefonunkra, vagy töltsünk le onnan. A művelethez csupán pár apró program szükséges, amely ***Debian*** alatt az `obexftp` csomagban található, de letölthető a projekt honlapjáról is.
A fájl átvitelhez az első lépésként ki kell deríteni a cél eszköz ***MAC*** címét. Ehhez a `hcitool` parancs `scan` opcióját használjuk.

```
debian:~# hcitool scan
Scanning ...
        xx:xx:xx:xx:xx:xx
        ➥Nokia 6021
debian:~#
```

Most, hogy már ismerjük a *MAC* címet, a következő lépés csatlakozni az eszközhöz. Ezt az `rfcomm` parancs `bind` opciójával tehetjük meg az alábbi módon:

```
rfcomm bind /dev/rfcomm0
➥xx:xx:xx:xx:xx:xx
```

Amennyiben nem rendelkezünk a `/dev/rfcomm0` eszközökkel, hozzuk létre az alábbi módon:

```
mknod /dev/rfcomm0 c 216 0
```

Amennyiben valamiért több eszközre lenne szükség, úgy a nullák helyére beírhatunk egytől kilencig számokat. Amennyiben sikerült a kapcsolódás, úgy kiadhatjuk az alábbi parancsot. Ha esetleg kifelejtjük a *MAC* címet, akkor mindig az első és leggyorsabban reagáló *Bluetooth* eszközre próbálja meg végrehajtani a parancsot. Az alábbi parancs például kilistázza az eszköz főkönyvtárát. (Ezt általában *XML* formátumban kapjuk meg.)

```
obexftp -b xx:xx:xx:xx:xx:xx -l
```

A *SajatKepek* könyvtár tartalmát például a `-l` opció paraméterezésével érhetjük el.

```
obexftp -b xx:xx:xx:xx:xx:xx -l
➥/SajatKepek
```

Állományt feltölteni a `-p`, letölteni a `-g` opcióval lehet. Fontos megjegyezni a *műveleti sorrend*et. Az alábbi példában a képet a *SajatKepek* mappába mentem a telefonon, de az opciók fordított – tehát először a `-p` és utána a `-c` – sorrendje nem ezt eredményezné.

```
obexftp -b xx:xx:xx:xx:xx:xx -c
➥/SajatKepek -p goodman02.gif
```

Hasznos opciók még: `-k` a törléshez, `-G` letöltés majd törlés, `-C` könyvtár váltás és létrehozás, amennyiben nem létezik.
Ha le szeretnénk zárni a *Bluetooth* kapcsolatot, adjuk ki a `rfcomm release /dev/rfcomm0` parancsot. Fontos megjegyzés, hogy nem minden telefon támogatja akármilyen állomány átvitelét, illetve az általam kipróbált telefonok egyikéről sem tudtam *JAVA (J2ME)* alkalmazást letölteni, vagy a telefonra feltölteni. Erre ezután is a *GPRS* vagy *CSD* (esetleg *HSCSD*) áll rendelkezésünkre.

Kék a zene...

Megfelelő *Bluetooth headset*-tel és *Bluetooth* adapterrel megvalósítható az is, hogy az *XMMS* a *Bluetooth headset*-ben szólaljon meg. Nagy minőséget azonban ne várjunk, mert elsősorban nem *Mozart* vagy *Beethoven* átvitelére tervezték ezeket az eszközöket, hanem normál beszédre.
Nem is akarom tovább fokozni a kedélyeket. Ahhoz, hogy tudjuk ilyen célra használni a számítógépünket, szükség lesz egy működő *ALSA* rendszerre, valamint az alábbi programokra, csomagokra: `automake` (minimum 1.7-es verzió), `libbluetooth-dev`, `libasound2-dev` (vagy másik nevén: `alsa-devel`). Kernelből sem mindegy, melyiket használjuk, ugyanis a sikerhez minimum ***2.6.11.7***-es-re lesz szükségünk.
Régebbi kernellel, vagy különálló *ALSA*-val nem működik a dolog.
A fordítás általában az alábbi parancsokkal megy végbe:

```
./bootstrap
./configure
make
make install
```

Amennyiben ***duplex*** (kétirányú) módon szeretnénk használni a ***headset***-et, úgy a kernelfordításakor az `emu10k1`-et (***Sound Blaster Live***) is bele kell fordítani.
Van még pár parancs, amit be kell gépelnünk. Később ezek esetleg ***bash scriptbe*** írhatóak, hogy egyszerűbb legyen az élet. Elsőként illesszük be a modult:

```
modprobe snd_bt_sco
```

Mielőtt a következő lépést végrehajtanánk, a modul fejlesztői azt javasolják, állítsuk le az `esound` szolgáltatást, amennyiben fut:

```
esdctl stop
```

Innen már szabad az út, készítsük elő hangátvitelre a ***Bluetooth adaptert*** a következő módon:

```
hciconfig hci0 voice 0x0060
```

Indítsuk el a szükséges szolgáltatást (az `xx`-ek helyére természetesen írjuk be a ***headset MAC*** címét, amit a már korábban ismertetett `hcitool` segítségével kérdezhetünk le)

```
btsco xx:xx:xx:xx:xx:xx
```

Innen kezdve például az *XMMS*-ben már látszik a ***Bluetooth headset***, mint hangkártya. (***Opciók -> Beállítások -> Kimenet: ALSA -> Beállítás***)
Amíg szeretnénk használni a ***headset***-et, addig ne lépjünk ki a `btsco`-ból. A `modprobe` és a `hciconfig` futtatásához természetesen rendszergazdai jogosultság kell, míg a `btsco` normál felhasználóként is futtatható. Ezt elkerülendő a `snd_bt_sco` modult érdemes beírni a `/etc/modules` fájlba, illetve

Egy tipikus kapcsolódás

```
debian:~#  wvdial gprs
--> WvDial: Internet dialer
➥version 1.54.0
--> Cannot get information
➥for serial port.
--> Initializing modem.
--> Sending: ATZ
ATZ
OK
--> Modem initialized.
--> Sending: ATDT*99***1#
--> Waiting for carrier.
ATDT*99***1#
CONNECT
~[7f]}#@!}!} }
➥}2}#}$@#}!}$}%\}"}&} }*} }
➥g}%~
--> Carrier detected.
➥Waiting for prompt.
~[7f]}#@!}!} }
➥}2}#}$@#}!}$}%\}"}&} }*} }
➥g}%~
--> PPP negotiation
➥detected.
--> Starting pppd at Tue Mar
➥14 18:56:09 2006
--> pid of pppd: 14656
--> Using interface
➥ppp0
--> local  IP address
➥84.224.17.98
--> remote IP address
➥10.6.6.6
--> primary   DNS address
➥193.225.153.17
Caught signal #2!  Attempting
➥to exit gracefully...
--> Terminating on signal 15
--> Connect time 3.4
➥minutes.
--> Disconnecting at Tue Mar
➥14 18:59:32 2006
debian:~#
```

a `hciconfig`-ot ***setuid*** jogosultsággal ellátni – bár ez utóbbi ***nem javallott*** biztonsági okok miatt.

Elnézést, kapcsolná az Internetet?

Manapság már nem nagy igény, hogy mobiltelefonról tudjunk internetezni, sőt az is megoldható, hogy a telefon csak a ***modem*** szerepét játssza és a laptopon is internetezhetünk.

A jó hír: a mobilszolgáltatók már jó ideje lehetővé teszik azt is, hogy ne az eltöltött idő arányában, hanem a ***le/feltöltött bájtok után fizessen*** az ügyfél (***GPRS – General Packet Radio Service***). Persze a tarifák még messze állnak a Kánaántól, de már nem elérhetetlenek.

Először is szükséges egy jól konfigurált ***kernel***, ami tartalmaz ***Bluetooth*** soros port (***tty***) és ***ppp*** (***point-to-point protocol***) támogatást. Ha ezzel megvagyunk, fel kell telepíteni a ***ppp*** démont (***Debianban*** `ppp` a csomag neve), valamint egy tárcsázót (én a `wvdial`-t használom).

A `wvdial` beállítása egyszerű.

A ***/etc/wvdial.conf*** fájlnak a következőt kell tartalmaznia:

```
[Dialer gprs]
Phone = *99#
// vagy Phone = *99***1#
// ez az opció telefonfüggő
Username = ''
Password = ''
New PPPD = yes
Modem = /dev/rfcomm0
```

A kapcsolat felépítése az ismertetett konfigurációs állomány után már egyszerűnek mondható. Adjuk ki az `rfcomm bind /dev/rfcomm0 xx:xx:xx:xx:xx:xx` parancsot. Amennyiben az `rfcomm0`-át már használjuk, úgy az `rfcomm1`-et és így tovább, ekkor azonban ne felejtsük el átírni a ***/etc/wvdial.conf*** fájlban. Ezután adjuk a `wvdial gprs` parancsot. Ha mindent jól csináltunk, akkor megkezdődik a tárcsázás. A folyamatot bizonyos telefonoknál a telefon kijelzőjén is nyomon követhetjük. Ha végeztünk, a kapcsolat befejezéséhez nyomjunk a ***wvdial*** ablakában egy ***CTRL+C*** kombinációt. Ha valami hiba lépne fel a tárcsázás során, úgy az a ***/var/log/messages*** fájlba kerül.

Amennyiben már bootoláskor szeretnénk csatlakozni a telefonunkhoz, úgy a ***/etc/bluetooth/rfcomm.conf*** fájlt hozzuk létre az alábbi tartalommal – az `xx`-ek helyére a mobiltelefonunk *MAC címét* kell írni:

```
rfcomm0 {
  bind yes;
#       # Bluetooth address of
          ➥the device
          device
➥xx:xx:xx:xx:xx:xx;
#       # RFCOMM channel for
          ➥the connection
          channel 1;
#       # Description of the
          ➥connection
          comment "mobiltelefon";
}
```

Bluetooth 2.0

Az új változat már köztünk van. ***Háromszor gyorsabb***, mint elődje, ***alacsonyabb*** az ***energiafogyasztása*** is – ez utóbbi mobil alkalmazásoknál jöhet jól –, továbbá a mostani rendszerekkel ***kompatibilis***. A nagy kérdés, hogy van-e rá igény, hiszen itt a ***wifi*** és küszöbön a ***wireless USB*** is. Mindezek sokkal nagyobb sávszélességet ígérnek, mint az új ***Bluetooth***. Szerintem azonban még sok kernelverziónak kell kijönnie, mire akár a ***wifi***, akár a ***wireless USB*** leváltja a ***Bluetooth***-t a jelenlegi alkalmazásokban, tehát 2-3 évre mindenképp ajánlott a ***Bluetooth*** beszerzése, amennyiben nem a sebességet hajszoljuk.

Medve Zoltán
(e-medve@e-medve.hu)

2001-ben kezdett „Linuxolni”, de már korábban is ismerkedett a szabad szoftverek világával. Ha éppen nem a gép előtt ül, akkor fotózgat, olvasgat vagy bicajozik.

KAPCSOLÓDÓ CÍMEK

Linux kernel
➲ ftp://ftp.kernel.org/pub/linux/kernel/v2.6

BlueZ projekt honlapja
➲ http://www.bluez.org/download.html

Bluetooth ALSA honlap
➲ http://bluetooth-alsa.sourceforge.net/

ObexFtp honlap
➲ http://triq.net/obexftp.html


Linuxvilág
www.linuxvilag.hu