

## A pingvin igenis vándormadár (2. rész)

Avagy használjunk Linuxot a mobil eszközökön.

**A** cikksorozat bevezető részében telepítettünk a gépünkre egy *Debian Linuxot*, valamint elkezdjük beállítani a rendszert. Első lépésben megvizsgáltuk milyen kernel és felhasználói program szintű megoldások léteznek a mobil eszközünk energiagazdálkodási paramétereinek beállítására.

Ha már megteremtettük a lehetőséget, hogy a gépünkkel eltávolodjunk a hálózati tápellátástól – a konnektortól, akkor jó volna ha a hálózati kábelt is ki tudnánk húzni a gépből és vezeték nélkül tudnánk Internetezni. Nézzünk is gyorsan pár megoldási lehetőséget.

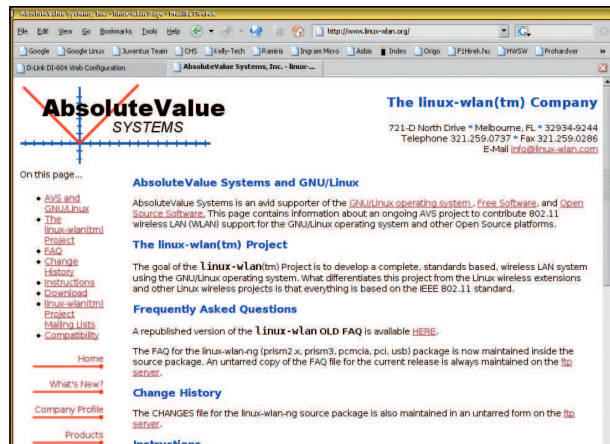
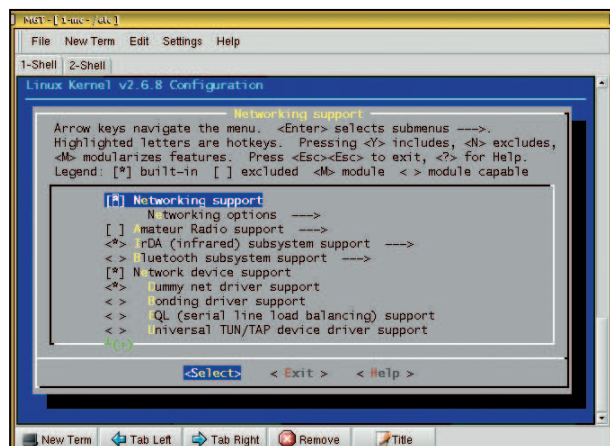
### A WLAN – a vezeték nélküli hálózati csatlakozás

Napjainkban egyre népszerűbb a technológia. Mostanra minőségben és sebességben is elérte azt a szintet, hogy bizonyos felhasználási környezetekben alternatívát nyújtson a vezetékes hálózati kapcsolattal szemben. Arról nem is beszélve, hogy bizonyos környezetekben pedig egyenesen forradalmi változásokat idézhet elő. Ilyen például a szállodákban, konferencia központokban, éttermekben, oktatási intézményekben vagy éppen a reptereken gomba módjára szaporodó úgynevezett a forró pontok (hot-spot), ahol egy vezeték nélküli hálózati csatlakozással ellátott eszközzel szabadon (esetleg díjfizetés ellenében) tudunk az Internetre csatlakozni. Ezzel az említett intézmények olyan értékknövelt szolgáltatást tudnak a vendégeik, látogatóik számára nyújtani, amely minden bizonnyal nagyon gyorsan nagyon népszerű lesz.

Mind ehhez semmi másra nincs szükségünk, mint beszerezni egy megfelelő vezeték nélküli hálózati kártyát, betenni a gépünkbe és beállítani. Előbbi kettőben ugyan nem, az utóbbiban viszont megpróbálok ezen hasábokon segítséget nyújtani.

Mind a 2.4-es, mind a 2.6-os rendszermag támogatja a vezeték nélküli hálózati kártyákat, így minden bizonnyal nem lesz problémánk ezen kártyák telepítésével. A régebbi ilyen kártyák esetleg jelenthetnek kihívást, de a *Linux* egyik legnagyobb áldása, hogy ezekre a problémákra és megoldásukra, vagy az oda vezető útra percek alatt rábukkanhatunk az Interneten. Erre a legjobb kiindulási pont vagy a [www.tuxmobile.org](http://www.tuxmobile.org), vagy egyszerűen a [www.google.com/linux](http://www.google.com/linux) oldal.

A most a kiskereskedelmi forgalomban kapható, otthoni vagy kisvállalati (*SOHO – small office, home office*) környezeti felhasználásra szánt kártyák többsége a 2.6-os rendszermagban a *Prism* meghajtót használja. Természetesen van



olyan kártya is, amely nem, de ezt egy pillanat alatt kideríthetjük. 2.6-os rendszermagot használva futtassuk a *root* felhasználó jogosultságaival az *lspci* parancsot, amely kilistázza nekünk az összes a rendszerben jelen lévő PCI eszközt. Amennyiben a gépünkben beépített, vagy *PCI* kártyás vezeték nélküli hálózati csatlakozás van, akkor ezek között meg kell, hogy találjuk az eszközünket. Amennyiben *USB*-n keresztül csatlakozó hálózati kártyánk van, akkor az *lsusb* parancs által listázott eszközök között kell, hogy szerepeljen a hálózati kártya. Amennyiben pedig *PC* kártya (*PCMCIA* kártya) formátumú *WLAN* kártyánk van, akkor a *pcmcia-cs* csomag *cardctl* programjának futtatásával kaphatjuk meg a jelenleg dokkolt *PC* kártyák adatait.

## Egy példa a waproamd-re

Nézzünk egy példát arra, hogyan is hozhatunk létre egy ismert elérési ponthoz egy olyan szkriptet, amely az elérési pont felismerése után automatikusan csatlakozik a felismert elérési ponthoz.

Hozzuk tehát létre az `/etc/waproamd/scripts` könyvtárban az elérési pont `MAC` címével, vagy `ESSID` nevével ellátott állományt. Fontos, hogy a létrehozott állomány rendelkezzen futtatási joggal, különben nem fog elindulni!

A `waproamd` szkriptjeinek futtatási sorrendjéről jó tudni, hogy először a `MAC` címmel ellátott állományokat próbálja futtatni, azok közül is először a kisbetűségeket, majd a nagybetűségeket. Ha nem talál illeszthető mintát, akkor megpróbálja az `ESSID` alapján megkeresni a megfelelő futtatható állományt. Ha ez sem járt sikerrel, akkor jön a `default` nevű állomány futtatása, amely megpróbálja kitalálni, hogy mi az elérési pont `MAC` címe, vagy `ESSID` azonosítója.

Mennyiben `MAC` címmel ellátott állományt szeretnénk létrehozni, akkor adjuk névnek például a `00-01-02-03-a0-a1` nevet és adjunk rá futtatási jogot. Amennyiben a `HomeWLAN` `ESSID`-hez szeretnénk szkriptet írni, akkor adjuk az állománynak az `ssid:HomeWLAN` nevet. Amennyiben szóközt, vagy egyéb speciális karaktert is tartalmazna az elérési pont neve, akkor használjuk a szabványos `http URL escape` szintaktikát, vagyis tegyük a karakter ASCII kódját egy `%` után.

A szóköz például a `%32` kóddal írható le.

Ha ezzel megvagyunk, akkor még a `WEP` titkosításhoz tartozó kulcs beállítása lehet hátra. Ezt a `/etc/waproamd/keys` könyvtárban az előbbi névhasználattal megegyező módon létrehozott állományban tehetjük meg. Egyszerűen írjuk bele azt a 10, vagy 26 számjegyet, ami kell.

Nézzünk egy példát egy `WLAN` indító programra. Ha létrehoztuk a megfelelő állományt, akkor helyezzük el benne az alábbi tartalmat:

```
#!/bin/bash
case "$1" in
    start)
        iwconfig eth1 essid "HomeWLAN"
        iwconfig eth1 mode Managed
        iwconfig eth1 key 1122334455
        ;;
    stop)
        iwconfig eth1 key off 2> /dev/null
        ;;
esac

[ "$1" = "start" ] && exec
/etc/waproamd/waproamd.action eth1 up
[ "$1" = "stop" ] && exec
/etc/waproamd/waproamd.action eth1 down
```

Ezzel csináltunk egy szkriptet, amely csatlakozik a `HomeWLAN` nevű elérési ponthoz az `eth1` hálózati csatlakozóval, amely az `1122334455` kulcsot használja a `WEP` titkosításhoz.

Amennyiben a hordozható gépünkön van olyan gomb, amely a vezeték nélküli hálózati csatlakozót aktiválja, akkor ezt a gombot hozzátársítva a `waproamd` programhoz megtehetjük azt, hogy nem állandóan fut a démon, hanem csak akkor amikor mi ezt szeretnénk. Ez nagyon hasznos lehet abból a szempontból, hogy így feleslegesen nem meríti az állandó keresés az akkut.

Amennyiben már telepítettünk grafikus felületet a gépünkre, akkor a `cardinfo` parancs futtatására egy jópofa grafikus PC kártya felügyeleti programot tudunk indítani.

Ha megtaláltuk, hogy a kártyánkat milyen lapkakészlettel (chipset) látták el, akkor a mag beállításai között a `Device Drivers/Networking Support/Wireless LAN` menüpont alatt megtaláljuk a rendszermag által támogatott modulokat. Mint említettem a beépített, a `PCI` és a `PC` kártyák jelentős része a `Prism` meghajtót használja. Amennyiben `802.11b` szabványú kártyánk van, akkor a `Hermes` lapkakészlet és a `Prism 2.5` modulokat fordítsuk a rendszermagba, amennyiben viszont `802.11g` szabvány (`22/54/108Mbit/s` sebességű kártyák) is támogató kártyánk van, akkor válasszuk az `Intersil Prism` modult és fordítsuk bele a magba.

Ha újrarendeltük a rendszermagot és újraindítottuk a gépet ezzel a rendszermaggal, akkor a betöltés folyamán már meg kellett, hogy jelenjen a kártya. Ezt egyébként ellenőrizhetjük a `dmesg | more` parancs kiadásával. Ha a kártya fizikailag rendelkezésre áll, akkor telepítsük fel a `WLAN` kártyák kezeléséhez szükséges programokat tartalmazó `linux-wlan-ng` csomagot. Ebben a csomagban található többek között az `iwconfig` állomány is, amellyel a kártya kapcsolódását tudjuk beállítani a hozzáférési pontokhoz (access point). Ha ez is megtörtént, akkor nincs más dolgunk hátra, mint beállítani a hálózati beállításokat. Kezdjük azzal, hogy meg-

keressük a megfelelő elérési pontot és megtudjuk a paramétereit. A legfontosabb paraméterek, amelyekre szükségünk lesz az elérési pont `ESSID`-je, ami gyakorlatilag egy szöveges azonosító, az elérési pont `MAC` címe, a használt titkosítás és a kulcs kiosztásának módja. Ezeket az információkat vagy elkérjük a rendszergazdától, vagy megpróbáljuk felderíteni. Utóbbiban segítségünkre lehet például a `waproamd (Wireless Access Point Roaming Daemon)` nevű csomag. Ez a program alapértelmezésben a rendszer indításakor elindul és folyamatosan keresi az elérhető csatlakozási pontokat. Amint talál egy elérési pontot, akkor megnézni, hogy az `/etc/waproamd/scripts` könyvtár tartalmaz-e az elérési pontra jellemző információt – vagy a `MAC` címet, vagy az `ESSID` azonosító nevet – és amennyiben igen, úgy lefuttatja az adott szkriptet. Ha nem talál, akkor az alapértelmezett (default) szkript fut le és próbálja meg létrehozni a csatlakozást. Ez a nyilvános hotspot-ok esetén egy jó választás lehet, azonban ha az elérési pont tartalmaz valamilyen speciális beállítást, akkor már kérdéses, hogy a felderítés mennyire jár sikerrel. Amennyiben olyan elérési ponthoz szeretnénk csatlakozni, amelyek `WEP` titkosítást használ, akkor az `/etc/waproamd/keys` könyvtárba helyezzünk el egy `<access_point_MAC_címe>.wep` nevű állományt, amely a `WEP` kulcsot tartalmazza.

Az előbb bemutatott `waproamd` egy nagyon jópofa szolgáltatás, de az élet azért nem fenéki tejfel. Sajnos vannak olyan

kártyák – főleg a régebbi típusok -, amelyek nem, vagy csak részben támogatják a felderítési funkciót. Arról, hogy kártyánk rendelkezik-e ilyen képességekkel meggyőződhetünk az `iwlist <wlan_csatoló_neve> scan` parancs beírásával, ahol a **WLAN** csatoló neve az az interfész, amelyik az `iwconfig` parancs által listázott eszközök között **WLAN** támogatást mutat. Amennyiben az `iwlist` azt adná vissza, hogy az eszközünk nem támogatja a **WLAN** felderítő opció használatát, még ne keseredjünk el. Van ugyanis lehetőség arra, hogy kézzel beállítsa a kapcsolati paramétereket tudjuk a vezeték nélküli hálózatot. Sőt, annak ellenére, hogy az `iwlist` nem ad vissza értékelhető eredményt, a `waproamd` még talán működni is fog. Az én notebookom például egy olyan beépített **WLAN** kártyával van megáldva, amelyik az `iwlist` parancsra meg sem mukkan, de remekül be tudtam állítani a `waproamd`-t és a bekapcsoló gombra tökéletesen fut. Amennyiben valamilyen oknál fogva mégis a kézi beállítás mellett döntünk, úgy ehhez az `iwconfig` parancsot kell használni. Használata teljesen megegyezik az `ifconfig` parancs használatával, gyakorlatilag hasonló írásmódot kell használni. Nagyon jó sűgőoldal tartozik az `iwconfig` és tulajdonképpen az összes vezeték nélküli hálózati csomagban szereplő eszközhöz, úgyhogy nézessük szorgalmasan a sűgőt! További hasznos információkhoz juthatunk a <http://www.linux-wlan.org/> oldalon, valamint a már említett oldalak bármelyikén.

Most tehát beállítottuk a vezeték nélküli hálózati kártyánkat, ezzel már boldogan tudunk netezni. Amíg van a közelben egy elérési pont. De mi történik, ha nincs? Akkor bizony marad a régi jól bevált megoldás, a mobiltelefonos Internet. A mobiltelefonos Internet elérésnek két fajtája van, a kapcsolt vonali elérés (ma már általában minden szolgáltató támogatja a **HSCSD**-t, amely egy nagy sebességű kapcsolt vonali elérést biztosít), illetve a csomagkapcsolt megoldás, a jól ismert **GPRS**. Mi most inkább a **GPRS** kapcsolattal foglalkozunk, mivel ennek használata a jövőben talán lényegesen elterjedtebbé válik, mint a kapcsolt vonali netezés. Ahhoz azonban, hogy **GPRS**-en át tudjuk a hálózatot elérni a notebookunkról, ahhoz bizony először össze kell kapcsolni a gépet és a mobiltelefont. Ezt megtehetjük vezetékkel, vagy infrán, vagy **bluetooth**-on keresztül. Én most először az infrás kapcsolatot mutatom be, mert az infra mára gyakorlatilag minden notebookban és telefonban megtalálható, így ezt minden bizonnyal minden Kedves Olvasó ki fogja tudni próbálni.

## Az IrDA

Ahhoz, hogy a notebookunk infra portját a hálózat elérésére tudjuk használni, először is elérhetővé kell tenni. Nem megfelelő módon a dolgunk az, hogy először a megfelelő rendszermag modulokat belefördítsük a kernelbe. Indítsuk el ismét a kernel beállítási felületét és menjünk a **Device Drivers/Networking Support/IrDA (infrared) subsystem support** menüponthoz. Ott találunk egy pár modult, ezeket én az alábbi összeállításban fordítottam a rendszermagba:

```
--- IrDA protocols
<*> IrLAN protocol
<M> IrNET protocol
<*> IrCOMM protocol
```



```
[*] Ultra (connectionless) protocol
--- IrDA options
[*] Cache last LSAP
[*] Fast RRS (low latency)
```

Ha ez megvan, akkor lépünk be az **Infrared-port device drivers** menübe és állítsuk be az alábbi modulokat is:

```
--- SIR device drivers
<M> IrTTY (uses Linux serial driver)
--- Dongle support
[*] Serial dongle support
--- Old SIR device drivers
<M> IrPORT (IrDA serial driver)
```

Ennek a három modulnak a beállításával nekem eddig majdnem minden notebookon sikerült beállítani az infra kapcsolatot, de ettől függetlenül lehetnek olyan eszközök is, amelyekhez más modulokat is be kell fordítani. Erről bővebbet olvashatunk a rendszermag forrásának a dokumentációjában.

Ha idáig eljutottunk, akkor természetesen fordítsuk újra a kernelt és indítsuk újra a gépet. Miután betöltődött a friss rendszermag, a `dmesg`-gel megnézhetjük, hogy a gép az induláskor felismerte-e az infra portot a gépben. Ha nem, akkor próbálkozhatunk más modulokkal a rendszermagban, ha igen, akkor folytassuk a beállításokat.

Következő dolog, amit el kell érniünk az az, hogy megszólaltassuk a gépünk infra portját. Ehhez fel kell tenni az **irda-utils** nevű csomagot. A telepítés folyamán a telepítő több kérdést is feltesz. Az első, hogy rendszerindításkor az infra engedélyezve legyen-e. Válasszuk az igent. Második kérdés, hogy engedélyezzük-e a felderítési módot, amellyel a gépünk a közelben lévő eszközökhöz akar majd csatlakozni. Én itt is az igent választottam.

Megjegyzés: a notebook akkujának kímélése szempontjából esetleg indíthatjuk kézzel is az infrát, ha úgy érezzük erre szükségünk van.

Következő kérdés, az IrDA eszköz típusa. Itt notebookok esetén 95%-ban a `serial` választ kell megjelölni, hacsak nem valamilyen natív – általában külső – eszközt csatlakoztattunk. Következő lépésben meg kell adnunk a soros port helyét. Nálam ez a `/dev/ttyS1`. Az utolsó kérdés az infra vezérlőnkhez tartozó dongle típusa. Notebookok esetén itt

99%-ban a none választ kell kijelölni.

Ha ezzel megvagyunk, akkor jelentem, bekonfiguráltuk az infra portot. Most tegyük vele szembe egy megfelelő telefont és indítsuk el az `irdadump` nevű programot. Ez a folyamat gyakorlatilag egy pingelésnek fog megfelelni. Az infra port összegyűjti a látható eszközöket, amelyeknek meg kell jelenniük a kidobott listában. Nálam egy Nokia 7210-es mobiltelefon a tesztalany és a következő üzenetet kapom:

```
19:52:14.297525 xid:rsp db94b91e < 0000edd6 S=6
s=0 Nokia 7210 hint=b125 [ PnP Modem Fax Telephony
↳ IrCOMM IrOBEX ] (27)
19:52:14.303534 xid:cmd db94b91e > ffffffff S=6
↳ s=1 (14)
19:52:14.393525 xid:cmd db94b91e > ffffffff S=6
↳ s=2 (14)
19:52:14.483510 xid:cmd db94b91e > ffffffff S=6
↳ s=3 (14)
19:52:14.573497 xid:cmd db94b91e > ffffffff S=6
↳ s=4 (14)
19:52:14.663481 xid:cmd db94b91e > ffffffff S=6
↳ s=5 (14)
19:52:14.753466 xid:cmd db94b91e > ffffffff S=6
↳ s=* trikko hint=4400 [ Computer LAN Access ] (22)
```

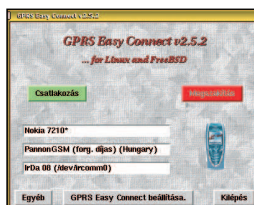
Ez teljesen jó, az első sorban ott látszik a telefon, meg az, hogy ez egy modem és egy faxmodem is egyben. Ezzel gyakorlatilag működik is az infra és a telefon közötti kommunikáció. Nincs más hátra, mint az Internetes kapcsolat beállítása.

### GPRS Internet mobiltelefonos beállítása

Ahhoz, hogy beállítsuk a telefonos kapcsolatot, szükségünk lesz a `ppp` csomag telepítésére. Ennek a *Debian Sid* gyűjteményben a cikk írásakor a 2.4.2-es verziója volt feltöltve. Célszerű ebből a csomagból mindig a legfrissebbet használni, mert olykor ez sok fejtevéstől kímélhet meg minket. Ha feltettük a `ppp` csomagot, akkor két út áll előttünk. Az első, hogy szépen kézzel elkészítjük a `ppp` kapcsolódási szkripteket, a másik, hogy egy segédprogramhoz fordulunk, amely ezt megteszi helyettünk. Bár az előbbi sem egy különösen bonyolult feladat, ráadásul az Interneten tucat-szám található tippet, tanácsokat és példa beállításokat is, akár szolgáltató specifikusan is, mi most mégis nézzünk egy programot, amellyel nem csak könnyen beállíthatjuk és vezérelhetjük a `GPRS` kapcsolatot, de mindennek fejében mindezt azzal a tudattal tehetjük, hogy egy olyan segédprogramot használunk, amelyet honfitársaink készítettek. A program neve: *GPRS Easy Connect*.

### Amit a GPRS Easy Connectről tudni kell

Személy szerint a nyáron, amikor a fejembe vettem, hogy végre beállítom a `GPRS` kapcsolatot a gépemem, hogy ne csak kábel mellől tudjak vezetni, rengeteget kutakodtam a hálózaton, hogy találjak valami használható, felhasználóbarát megoldást a kapcsolat beállítására. Olyat, amelyet nem csak én, de adott esetben egy mindennapi felhasználó is könnyen és gyorsan be tud állítani. Nem kis örömmre



rábukkantam a *GPRS Easy Connect*-re, amelyre a mai napig is úgy tekintek, mint egy áldásra. A program egy teljesen jól használható grafikus felülettel rendelkezik és a beállítása tényleg nem igényel különösebb tudást. Amire pedig szükség lehet, az le van dokumentálva.

Aki kedvet kapott a program kipróbálásához,

az letöltheti a legfrissebb verziót a <http://easyconnect.linuxuser.hu/modules.php?name=index> webcímről.

Egy másfél megabájtos `tar.gz` csomagot kell letöltenünk, amelyet kicsomagolunk, majd a telepítő könyvtárban található `INSTALL` állomány futtatásával egy egyszerű grafikus telepítőt indíthatunk. A telepítés menete nagyjából követi a „next-next-finish” módszert, tehát senkinek nem okozhat különösebb problémát.

Ha befejeződött a telepítés akkor az `/usr/bin/gprsec` parancsot futtatva indíthatjuk a programot. A program a telepítőhöz hasonlóan egy nagyon baráti grafikus felületet ad mind a beállítások elvégzésére, mind pedig a kapcsolat nyomkövetésére.

Kezdjük a beállításokat a „*GPRS Easy Connect beállítása*” feliratú gombra kattintva. Itt kiválaszthatjuk a készülékünk típusát, a szolgáltatót, a szolgáltató `APN`-jének nevét megadhatjuk, akár csak a felhasználó nevünket, jelszavunkat. Ki kell választanunk továbbá a csatlakozási kaput, ez valószínűleg a `/dev/ircomm0` lesz. Engedélyezhetjük a dinamikus `DNS`-t, vagy akár megadhatjuk saját magunk is két `DNS` kiszolgáló címét, ha nekünk éppen úgy tetszik.

A *Választható paraméterek* gombra kattintva beállíthatjuk a kapcsolat sebességét, az újrcsatlakozások számát és időtartamát.

A *Ritkán használatos paraméterek* között találunk egy jelölő négyzetet, amelynek a beállítása sok fejtevéstől kímélhet meg minket. Nálam a telepített alapbeállításokon nem volt beállítva az *Önhitelesítés tiltása*. Ez olyan kellemetlen problémát okozott, hogy bár a gépem felcsatlakozott a hálózatra, de mivel a vissza irányú hitelesítés nem működött, ezért azonnal le is bomlott a kapcsolat. Erre figyeljünk oda, ha ilyet tapasztalnánk, ez lehet a megoldás.

A *Limitfigyelés* menüpont alatt beállíthatunk Mb-ban mért adatforgalmi korlátot. Hasznos lehet azoknak a felhasználóknak, akik egy bizonyos mennyiségű ingyenes, vagy előre kifizetett forgalommal rendelkeznek. A *Forgalmazott adatmennyiség* menüben statisztikai adatokat találunk.

Amennyiben befejeztük a beállításokat, akkor a *Rendben/Adatok mentése* gombra klikkelve elmenthetjük a beállításainkat. Ezután nincs más dolgunk, mint a nagy zöld *Csatlakozás* gombra kattintani és internetezni.

Miután elindult a kapcsolat, az `ifconfig` paranccsal listázott csatlakozók között meg kell jelennie egy `pppX` csatlakozónak, ez lesz a `GPRS` kapcsolat által használt eszköz.

Ezzel a cikk végére is értünk. A következő részben további hasznos tippeket és beállítási tanácsokat osztok meg a Kedves Olvasóval. Aki azonban türelmetlen lenne, azt csak arra tudom biztatni, hogy használja a gépét, próbálkozzon önállóan a beállításokkal. Sok sikert!

Illés Viktor