

## A rendszermag beállításai és fordítása

A 2.5-ös sorozat esetében a korábbinál sokkal egyszerűbb saját rendszermagot készíteni vagy illesztőprogrammal bővíteni a rendszerünket.

A rendszermagfordítás művelete két részből áll: először meg kell adni a megfelelő beállításokat, majd ezekkel a beállításokkal el kell végezni magát a fordítást. A 2.5-ös előtti magváltozatoknál a beállítások a *Config.in* nevű állományokba kerültek, ilyen minden alkönyvtárban volt; sűgőként pedig egy központi leírás, a *Documentation/Configure.help* fájl szolgált. A fordítási folyamat leírására szolgáló nyelv parancssori jelleget öltött, és a felhasználónak felkínálható beállításokat az éppen megjelenített beállításoktól függően választotta ki.

A dolog viszonylag jól működött, ám idővel túl sok különféle beállítás terhelte le ezt a nyelvet, ami már nem tudta megfelelően kezelni őket. A 2.5.45-ös változat megjelenésekor *Roman Zippel* újírta a beállítónyelvet, és új beállítóprogramok kerültek a fő rendszermagfába is. Az új beállító nyelv rugalmasságát és tudását tekintve egyaránt felülmúlja elődjét. A sűgő szövegét is egyesíti a beállításokat vezérlő részekkel, így az illesztőprogramok foltjainak telepítése is könnyebb, és nem kell a központi *Configuration.help* fájljal való ütközések miatt aggódn.

A 2.5-ös sorozatban *Kai Germaschewski* és további kbuild-fejlesztők lassan átdolgozták a rendszermag *Makefile* rendszerét, így a rendszermagnak a kiválasztott beállításokkal való lefordítása is egyszerűbb lett. Írásomban a 2.5-ös rendszermagban szereplő *Makefile* és beállítófájlok formátumát, illetve az illesztőprogramok hozzáadásának és a rendszermag lefordításának a módját ismertetem.

### A rendszermag beállításai

A rendszermag beállításait szöveges vagy grafikus beállítóprogrammal adhatjuk meg. A szöveges beállítóprogram a *make config* parancsral indítható; a beállítási lehetőségeket sorban egymás után kínálja fel a felhasználónak (1. kép). Az ugyancsak szöveges *ncurses* és változat jóval népszerűbb, indításához a *make menuconfig* parancsot kell kiadni (2. kép). A Qt alapú grafikus beállítóprogram a *make xconfig* parancsral vehető használatba (3. kép).

A rendszermag beállításait kezelő program indításkor kiolvassa a fő beállítófájl – *i386* alapú gépeknél az *arch/i386/Kconfig* – tartalmát. A többi géptípus is saját beállítófájlokkal bír a megfelelő könyvtárakban. A fő beállítófájl további állományokat is magába olvaszt a rendszermag könyvtárjában lévő különféle alkönyvtárakból. Ezek a fájlok hasonló módon további állományokat foglalhatnak magukba. Az *arch/i386/Kconfig* fájlban például az alábbi sor szerepel:

```
source "sound/Kconfig"
```

Vagyis beillesztjük a fenti állomány tartalmát. A *sound/Kconfig* fájl ugyanakkor több más állományt is magában foglal:

```
source "sound/core/Kconfig"
source "sound/drivers/Kconfig"
```

```
[greg@hump linux-2.5.15] make config
make[1]: `scripts/kconfig/conf' is up to date.
./scripts/kconfig/conf arch/i386/Kconfig
#
# using defaults found in .config
#
*
* Linux Kernel Configuration
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL) [Y/n/?]
```

1. kép A rendszermag beállításainak megadása a *make config* segítségével

```
Linux Kernel v2.5.59 Configuration
Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module <> module capable

Code maturity level options --->
general setup --->
loadable module support --->
processor type and features --->
power management options (ACPI, APM) --->
bus options (PCI, PCMCIA, EISA, MCA, ISA) --->
executable file formats --->
Memory Technology Devices (MTD) --->
parallel port support --->
plug and play support --->

<Select> <Exit> <Help>
```

2. kép A *make menuconfig* használatakor könnyebb visszalépni és kijavítani a hibákat

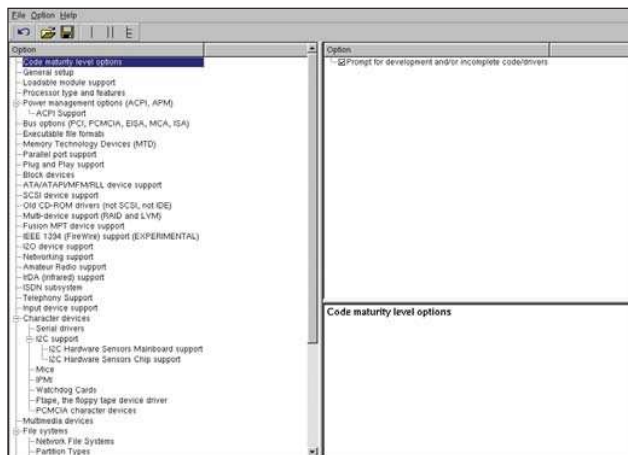
```
source "sound/isa/Kconfig"
source "sound/pci/Kconfig"
source "sound/ppc/Kconfig"
source "sound/arm/Kconfig"
source "sound/usb/Kconfig"
```

A *sound/usb/Kconfig* fájl az összes olyan beállítást tartalmazza, ami az ALSA USB illesztőprogrammal kapcsolatos:

```
# ALSA USB drivers

menu "ALSA USB devices"
depends on SND!=n && USB!=n

config SND_USB_AUDIO
tristate "USB Audio/MIDI driver"
depends on SND && USB
help
Say `Y' or `M' to include support for
USB audio and USB MIDI devices
```



3. kép A Qt alapú make xconfig

### endmenu

A Kconfig-fájlokban megjegyzéseinket a kettős kereszt karakter (#) után írhatjuk be. A mögötte, azonos sorban található szövegrészeket a beállítóprogram nem olvassa be.

A menu és endmenu parancsok a beállítóprogramot arra utasítják, hogy újabb menüsintet vagy képernyőt nyisson meg. A menu sorban a menü nevét idézőjelek közé szűrva kell megadni. A fenti fájlban a menü neve a következő:

"ALSA USB devices".

A menük és a beállítási lehetőségek esetében megadható, hogy megjelenjen-e a képernyőn vagy sem. A fenti példában az USB-beállítások menüje csak akkor jelenik meg, ha a CONFIG\_SND és a CONFIG\_USB beállítást is kiválasztottuk; a függést a depends on SND != n && USB != n sor adja meg. A szükséges gépelés mennyiségének csökkentése érdekében minden beállítási lehetőség a CONFIG kulcsszóval kezdődik, amit a beállítónyelv nem használ. A beállításoknak a következő értékek adhatók meg:

- y – a beállítás engedélyezve van,
- n – a beállítás nincs engedélyezve,
- m – az adott dolog modulként kerül lefordításra.

Ha a CONFIG\_SND és a CONFIG\_USB beállítások nincsenek n értékre állítva (vagyis belefördítjük őket a rendszermagba, vagy modulok lesznek), a CONFIG\_SND\_USB\_AUDIO beállítás jelenik meg a felhasználó előtt. A beállítás a három lehetséges érték közül egyet vehet fel, vagyis „háromállású” kapcsolóként viselkedik. A felhasználó a következő szöveget látja: „USB Audio/MIDI driver:”

### A háromállású USB Audio/MIDI driver beállítás

A beállítási lehetőségek megadásakor az alábbi típusok közül választhatunk:

- bool – a változó az y és az n (igen, illetve nem) értéket veheti fel.
- tristate – a változó y, n vagy m értéket kaphat.
- int – a változó bármilyen számértéket felvehet.

Ezt a beállítási lehetőséget egy depends logikai sor szabályozza, ami a menüpontokéval azonos logika szerint működik. A CONFIG\_SND\_USB\_AUDIO lehetőség a CONFIG\_SND és a CONFIG\_USB beállításoktól függ, vagyis ha ezek valamelyikét



4. kép Az újonnan hozzáadott FooBar USB hangszóró

modulnak állítjuk be, akkor a CONFIG\_SND\_USB\_AUDIO beállítás is modulértéket vesz fel. Ha mindkét vezérlőbeállítás le van tiltva (vagyis n értéket kaptak), a függő beállítási lehetőség nem jelenik meg. Ha mindkettőnek y értéket adunk, a függő beállítás n, y vagy m értéket kaphat. Ezt a viselkedést egyetlen sor írja elő:

```
depends on SND && USB
```

A rendszermag kódjából a beállítási érték látható lesz (jelen esetben a CONFIG\_SND\_USB\_AUDIO), így a kód ennek és bármely beállítási értéknek a meglétét ellenőrizni tudja. Meg kell jegyezni, hogy a #ifdef-nek az egy .c fájlban belül valamilyen beállítási érték ellenőrzésére történő használata ellenkezik a rendszermagra vonatkozó programozási stílusbeli alapelvekkel. Ezzel már korábban is foglalkoztam egy cikkemben, lásd „Proper Linux Kernel Coding Style” (Linux Journal, 2002. július, ↗ <http://www.linuxjournal.com/article/5780>).

Az #ifdef parancsokat csak a .h fájlokban használjuk, a .c fájlok maradjanak tiszták és könnyen olvashatók.

A sűgő szövege korábban egyetlen nagy Configuration.help fájlba került. Most a leírás a Kconfig állományban, közvetlenül a depends sor után található. Egy help vagy ---help--- tartalmú sorral kezdődik, ezt további sorok követik, a help sorhoz képest két szóközzel beljebb.

### Új beállítási lehetőség hozzáadása

Ha új beállítási lehetőséget kívánunk hozzáadni, a meglévő Kconfig fájl csupán a kívánt helyen a megfelelő sorokkal kell bővítenünk. Ha például új USB illesztőprogram készül az ALSA hangrendszerhez, akkor az a sound/usb könyvtárba fog kerülni, és egy sound/usb/Kconfig fájl kell hozzáadni. Az alábbi illesztőprogram a képzeletbeli FooBar USB-s hangszóró vezérlését végzi. A CONFIG\_SND és a CONFIG\_USB beállítások engedélyezésétől, továbbá a CONFIG\_SND\_USB\_AUDIO beállítástól függ, ugyanis az új illesztőprogram felhasználja ennek az illesztőprogramnak bizonyos szolgáltatásait. Az új beállítási lehetőség az SND\_USB\_AUDIO pont után kap helyet, de még a záró endmenu parancs előtt, valahogy így:

```
config SND_USB_FOOBAR
    tristate "USB FooBar hangsz r
illesztiprogramja"
    depends SND_USB_AUDIO
    help
        `ll tsd Y 0rt0kre, ha FooBar USB
```

```
hangsz r t szeretnØl hasznÆlni.
```

```
A k d modulkØnt is hasznÆlhat .
(= Ilyenkor sz ksØg szerint
illesztheti
a fut rendszermaghoz, illetve
eltÆvol that .)
A modul a k vetkezi nevet kapja:
usbfoobar.o
```

A beállítási lehetőség akkor jelenik meg, ha az SND\_USB\_AUDIO beállítást választottuk ki (4. kép).

### A rendszermag fordítása

A rendszermag fordítása különálló Makefile állományok alapján történik, ezek összekapcsolása fordítás közben történik meg, így egy nagyméretű Makefile jön létre. A különálló Makefile állományok a megszokottól eltérő, kifejezetten a rendszermag fordítási folyamatára jellemző formátumot követnek. A Makefile feladatköre a szükséges fájlok – a megadott beállításoktól függő, megfelelő formátumú, vagyis az eredmény vagy modul lesz, vagy beépül a rendszermagba – létrehozatalára korlátozódik. Például a 2.5.59-es rendszermagbeli *drivers/usb/misc/Makefile* állomány így néz ki:

```
#
# Makefile for the rest of the USB drivers
# (the ones that don't fit into any other
# categories)
#
obj-$(CONFIG_USB_AUERSWALD) += auerswald.o
obj-$(CONFIG_USB_BRLVGER) += brlvger.o
obj-$(CONFIG_USB_EMI26) += emi26.o
obj-$(CONFIG_USB_LCD) += usblcd.o
obj-$(CONFIG_USB_RIO500) += rio500.o
obj-$(CONFIG_USB_SPEEDTOUCH) += speedtch.o
obj-$(CONFIG_USB_TEST) += usbttest.o
obj-$(CONFIG_USB_TIGL) += tiglusb.o
obj-$(CONFIG_USB_USS720) += uss720.o
```

```
speedtch-objs := speedtouch.o atmsar.o
```

Amennyiben a CONFIG\_USB\_LCD beállítás m értékét kapott, az *usblcd.c* modulba történő fordításáról az alábbi sor gondoskodik:

```
obj-$(CONFIG_USB_LCD) += usblcd.o
```

Egyéb esetben, ha a beállítás y értékét kapott, a kódrészlet közvetlenül a rendszermagba épül. Ha a megfelelő modul egyetlen .c fájlból jön létre, akkor mindössze ennyi kell ahhoz, hogy új rendszermag Makefile-t hozzunk létre.

Ha az illesztőprogram több .c állományból épül össze, az állományok neveit különálló sorokban kell felsorolni, az illesztőprogram, avagy a modul nevével párosítva. Az előbbi példában a fájlok és az illesztőprogramnevek listája a következőképpen alakul:

```
obj-$(CONFIG_USB_SPEEDTOUCH) += speedtch.o
```

és a másik a

```
speedtch-objs := speedtouch.o atmsar.o
```

Az első sor adja meg, hogy a speedtch modul létrejön-e. Ha igen, a sor jelzi, hogy a rendszermagba kerül-e vagy modul lesz. A második sor tudatja, hogy a *speedtouch.c* és az *atmsar.c* fájlok .o fájlokba fordítandók, és összekapcsolva fogják alkotni a speedtch.o modult.

A régebbi rendszermagoknál, ha egy fájl szimbólumokat is közzétett, akkor közvetlenül is szerepelnie kellett a rendszermag Makefile állományaiban. A 2.5-ös és újabb rendszermagoknál ez a kötelezettség már nem áll fenn.

### Új illesztőprogram hozzáadása a fordítási folyamathoz

Ha új illesztőprogrammal szeretnénk bővíteni a rendszermag fordításának folyamatát, és az illesztőprogram egyetlen fájlban található, akkor mindössze egy sorra van szükségünk. A FooBar USB hangszóró példájára utalva az

```
obj-$(CONFIG_SND_USB_FOOBAR) += usbfoobar.o
```

sort kell hozzáadni a *sound/usb/Makefile* állományhoz.

Ha az illesztőprogram két fájlban található, például a *foobar1.c* és a *foobar2.c* fájlból, akkor egy további sorra is szükség van:

```
usbfoobar-objs := foobar1.o foobar2.o
```

### Összegzés

A 2.5-ös rendszermagnál a beállítások megadásának, illetve a fordításnak a folyamata jelentősen leegyszerűsödött és rugalmasabbá vált a korábbi változatokhoz képest. Roman Zippel és Kai Germaschewski munkájának hála a rendszermag fejlesztői több figyelmet fordíthatnak magára a programozásra, és kevesebbet kell törődniük a rendszermag fordításának bonyolult folyamatával.

A Kbuild folyamatról további tudnivalókat *Sam Ravnborg*

kiváló írásában találhatunk, amely a

➔ <http://marc.theaimsgroup.com/?l=linux-kernel&m=104162417329638> címen érhető el.

*Linux Journal* 2003. május, 109. szám



**Greg Kroah-Hartman** (greg@kroah.com)

Jelenleg a Linux USB és a PCI Hot Plug rendszermag felelőse. Az IBM-nél dolgozik, ahol számos, a Linux rendszermagjával kapcsolatos kérdéssel foglalkozik.

## KAPCSOLÓDÓ CÍMEK

A Linux-rendszermag otthona

➔ <http://www.kernel.org>

A 2.5-ös fejlesztői rendszermag letöltési helye

➔ <ftp://ftp.kernel.org/pub/linux/kernel/v2.5/>

Rendszermag-levelezési lista

➔ <http://www.cs.helsinki.fi/linux/linux-kernel/>

2.5-ös rendszermag-beállítási tippek

➔ <http://www.linux-sxs.org/administration/25xconf.html>