

udev – állandó eszköznevek a felhasználói térben

Vessünk véget major és minor számok okozta kavalkádnak!

A 2.5-ös rendszermagtól kezdve, a rendszer minden fizikai és virtuális eszköze a felhasználói térből is látható, és a `sysfs` hierarchikus szerkezeten keresztül érhető el. Az `/sbin/hotplug` segítségével figyelmeztést kaphatunk, ha a rendszer új eszközzel bővül vagy eltávolítottuk valamelyiket. E két lehetőség használatával végre dinamikus, rugalmas eszközneveket használó `/dev` rendszert hozhatunk létre.

Cikkünk témája az `udev` program, amely a `devfs` feladatait vállalja át. Ez a megoldás bármely pillanatban képes a rendszer eszközeihez `/dev` bejegyzéseket szolgáltatni. Továbbá olyan képességekkel is rendelkezik, amelyeket ezidáig kizárólag a `devfs` segítségével nem lehetett megvalósítani: képes az eszköznevek folyamatos megőrzésére, mialatt az eszköz szabadon mozoghat az eszközfában, rugalmas eszköz elnevezési megoldásra, figyelmeztet ha az eszköz külső rendszerei megváltoznak, valamint a teljes névkiosztási szabályrendszert a rendszermagon kívülre helyezi.

A Linux gépeken a rendszer valamennyi eszközállományát a `/dev` könyvtárban találjuk. Az eszközállomány adja meg, hogyan érhető el a felhasználói program egy bizonyos alkatrész vagy függvényt. Például, a `/dev/hda` állományt hagyomány szerint a rendszer első IDE meghajtójának elnevezésére használjuk. A `hda` név megfeleltethető egy major és minor számnak, melyek alapján a rendszermag meg tudja állapítani, melyik alkatrészrel vegye fel a kapcsolatot. Jelenleg rengeteg nevet alkalmazunk, melyek a különféle major és minor számoknak felelnek meg.

Minden egyes major és minor számpárhoz valamilyen nevet rendelünk ami az adott alkatrésztípushoz tartozik. Ezt a hozzárendelést a Linux „*assigned names and numbers authority*” szervezete (röviden LANANA, magyarul „Linux név és szám összerendelési hivatal”) végzi, melynek jelenlegi eszközlístáját a honlapján találjuk (lásd a hálózati forrásokat).

Ahogy a Linux kezdett újfajta eszközöket is támogatni, ezeket is major és minor számtartományokhoz kellett rendelni, hogy aztán a felhasználók a `/dev` könyvtáron keresztül elérhessék azokat. Alternatív megoldásként az eszközöket elérhetjük a fájlrendszeren keresztül is (lásd a hálózati forrásokat). A 2.4-es és korábbi verzióban a major számok érvényes tartománya 1–255-ig terjedt, míg a minor számok 1–255 között lehettek. A korlátozott tartományok miatt, az új major

és minor számok kiosztását a 2.3-as fejlesztési ciklus idejére befagyasztották. Azóta a zárlatot már feloldották és a 2.6-os rendszermag érvényes major számtartományát 4,095-re növelték. Egyazon major számhoz pedig több mint egymillió minor szám rendelhető.

Melyik `/dev` bejegyzés melyik eszköz?

Amikor a rendszermag egy új eszköztípust talál, általában az eszköztípushoz tartozó következő major/minor párost osztja ki. Így aztán rendszerindításkor az első felderített USB nyomtató kapná a 180-as major számot és a 0-s minor számot, amire a `/dev` könyvtár alatt a `/dev/usb/lp0` hivatkozik. A második USB nyomtató a 180-as major számot és az 1-es minor számot kapná, amit a `/dev` alatt a `/dev/usb/lp1` azonosít. Ha a felhasználó újraszervezi az USB hálózatot, esetleg behelyez egy új USB elosztót, hogy több USB eszközt kezelhessen a rendszeren, a számítógép következő indításakor a nyomtatók USB keresési sorrendje megváltozhat, megcserélve ezzel a két nyomtató minor szám hozzárendelést. Ugyanez érvényes szinte minden eszközre melyet a számítógép működése közben eltávolíthatunk vagy behelyezhetünk. A *hot-plug* rendszerű PCI megoldások és buszok megjelenésével (ilyen például az IEEE1394, USB és a CardBus), majdhogynem minden eszköz szembekerül ezzel a problémával.

A 2.5-ös rendszermag megjelenésével jelentősen leegyszerűsödött annak megállapítása, melyik eszköz minor számát rendeltük melyik fizikai eszközhöz. Ha egy rendszerhez két USB nyomtatót csatlakoztattunk, a `sysfs/sys/class/usb` al-könyvtár szerkezete a következőképpen néz ki:

```
/sys/class/usb/
|-- lp0
|   |-- dev
|   |-- device ->
└-- ../../../../devices/pci0/00:09.0/usb1/1-1/1-1:0
    |-- `-- driver -> ../../../../bus/usb/drivers/usb1p
    `-- lp1
        |-- dev
        |-- device ->
        ../../../../devices/pci0/00:0d.0/usb3/3-1/3-1:0
            `-- driver -> ../../../../bus/usb/drivers/usb1p
```

```
$ cat /sys/class/usb/lp0/device/serial
HXOLL0012202323480
$ cat /sys/class/usb/lp1/device/serial
W09090207101241330
```

Az egyes USB eszközkönyvtárakon belül, amelyekre az *lp0/device* és az *lp1/device* közvetett hivatkozások mutatnak, számos USB-vel kapcsolatos információt találunk, ilyen például a gyártó azonosítója valamint a (remélhetőleg egyedi) sorozatszám. Amint az a fenti meghatározás állományaiból kiderül, a */dev/usb/lp0* eszközállomány a HXOLL0012202323480 sorozatszámú USB nyomtatóhoz tartozik, míg a */dev/usb/lp1* eszközállomány a W09090207101241330 számú nyomtatónak felel meg. Ha ezeket áthelyezzük, például mindkettőt egy USB elosztó mögé rakjuk, könnyen előfordulhat, hogy átneveződnek, hiszen induláskor más sorrendben próbáljuk őket elérni:

```
$ tree /sys/class/usb/
/sys/class/usb/
|-- lp0
|   |-- dev
|   |-- device ->
|   .../.../devices/pci0/00:09.0/usb1/1-1/1-1.1/
|   ↳ 1-1.1:0
|       `-- driver -> .../.../bus/usb/drivers/usb1p
|           |-- lp1
|               |-- dev
|               |-- device ->
|               .../.../devices/pci0/00:09.0/usb1/1-1/1-1.4/
|               ↳ 1-1.4:0
|                   `-- driver -> .../.../bus/usb/drivers/usb1p
$ cat /sys/class/usb/lp0/device/serial
W09090207101241330
$ cat /sys/class/usb/lp1/device/serial
HXOLL0012202323480
```

Mint a leírásból is látszik, az eltérő próbálkozási sorrend következtében a */dev/usb/lp0* eszköz most a W09090207101241330 sorozatszámú USB nyomtatóhoz tartozik. A *sysfs* segítségével viszont a felhasználó meg tudja állapítani, hogy a rendszer mag melyik eszközhöz melyik eszközfájlt rendelt. Ez pedig nagyon hatékony összerendelés, amit azelőtt nem lehetett egyszerűen megoldani. Mindazonáltal a felhasználót általában nem érdekli, hogy a */dev/usb/lp0* és a */dev/usb/lp1* felcserélődött, és most ezért valamilyen beállításfájlban meg kéne változtatni valamit. A felhasználó mindössze annyit szeretne, hogy a helyes nyomtatóra nyomtasson, függetlenül attól, hogy az hol helyezkedik el éppen az USB fán.

Tülméretes /dev

A legtöbb terjesztésben a */dev* könyvtár nem minden eszközállományához tartozik ténylegesen a géphez csatlakoztatott fizikai eszköz. A */dev* könyvtár ugyanis az operációs rendszer telepítésekor jön létre, amikor belekerül az összes ismert név. Egy Red Hat Fedora 1 verziót futtató gépen a */dev* könyvtár nem kevesebb mint 18,000 különféle bejegyzést tartalmaz. Ilyen mennyi-

ségű bejegyzés hamar kényelmetlenné válik, ha a felhasználó meg szeretné állapítani, milyen eszközök vannak éppen jelen.

devfs

A */dev* könyvtárban található számtalan eszközállomány miatt sok operációs rendszer inkább magára a rendszer-magra bízta a könyvtár kezelését, hiszen az mindig pontosan tudja, milyen eszközök vannak jelen. Ezt pedig a *devfs* nevű, memória alapú fájlrendszeren keresztül érhetjük el. A Linux is rendelkezik ilyen megoldással, amely idővel több különféle terjesztésben, például Gentoo Linuxban, vált kedvelté.

A *devfs* sok ember számára jelent megoldást a közvetlen problémákra. Ugyanakkor a Linux-alapú *devfs* megoldásnak máig van néhány megoldatlan problémája. Legnagyobb hiányossága, hogy nem képes állandó neveken létrehozni eszközcsomópontokat.

Az udev céljai

A korábban bemutatott problémák miatt indult el az *udev* projekt. Célja, hogy a felhasználói térben fusson, dinamikus */dev* könyvtárat hozzon létre, egységes eszközneveket használjon ha szükséges, és felhasználó térbeli API-t biztosítson az aktuális rendszereszközök adatainak eléréséhez. Az *udev* és a *devfs* összehasonlításáról bővebben olvashatunk a hátlózi források részben.

Az első pontot, a felhasználói térben való működést két lehetőség kihasználásával tudja teljesíteni. Egyrészt, az */sbin/hotplug* eseményeket generál valahányszor egy eszközt eltávolítunk vagy hozzáadunk a rendszerhez, másrészt a *sysfs* bármilyen szükséges információt meg tud adni a kiválasztott eszközről.

A második pontot, azaz a *dynamic /dev* szolgáltatást, úgy éri el, hogy valamennyi */sbin/hotplug* eseményt elfogja, a *sysfs*-ből kikeresi a felvett eszközhöz tartozó major és minor számokat, majd az eszközhöz tartozó rendszer mag névvel létrehozza a */dev* állományt. Ha az eszközt eltávolítottuk, az eszközhöz tartozó */dev* bejegyzést már nem túl nagy feladat törölni.

Az *udev* e két első célját már 2003 áprilisában képes volt végrehajtani mégpedig egészen apró, lefordított állapotban mindössze 6Kb kód segítségével, annak köszönhetően, hogy a hot-plug események elfogására és a *sysfs* használatára épülő elképzelés jól megvalósítható és egyszerűen elkészíthető volt. A 2003-as szerény kezdeteket követően az *udev* végül valamennyi célját elérte. Felhasználóinak rugalmas szabály-alapú rendszer alapján teszi lehetővé, hogy az eszközökhöz állandó neveket rendeljenek.

Az *udev* szabályait a */etc/udev/udev.rules* állományban találjuk, amely valamennyi olyan eszközt tartalmazza, amelyet a felhasználó az alapértelmezett névtől eltérő módon szeretne megadni.

Nézzünk egy *udev.rules* állomány példát:

```
# ha a /sbin/scsi_id visszatérési értéke
# "OEM 0815" az eszközt
# disk1-nek nevezzük
BUS="scsi", PROGRAM="/sbin/scsi_id",
↳ RESULT="OEM 0815", NAME="disk1"
```

```
# USB nyomtatót lp_color-nak nevezzük
BUS="usb", SYSFS_serial="w09090207101241330",
↳ NAME="lp_color"

# Az adott gyártó és modellszámmal ellátott SCSI
# lemezt boot-nak nevezzük
BUS="scsi", SYSFS_vendor="IBM",
↳ SYSFS_model="ST336", NAME="boot"

# A 00:0b.0 azonosítóju PCI hangkártyát dsp-nek
# nevezzük
BUS="pci", ID="00:0b.0", NAME="dsp"

# A második elosztó harmadik kapujára csatlakozta-
# tott USB egeret
# mouse1-nek hívjuk
BUS="usb", PLACE="2.3", NAME="mouse1"

# A ttyUSB1 eszközt mindig pda-nak nevezzük
# két további közvetett hivatkozással
KERNEL="ttyUSB1", NAME="pda",
↳ SYMLINK="palmtop handheld"

# USB webkameráinkat és a hozzájuk tartozó
# hivatkozásokat
# webcam0, webcam1, ... névvel illetjük
BUS="usb", SYSFS_model="xv3", NAME="video%n",
↳ SYMLINK="webcam%n"
```

Az udev szabályok az eszköz tulajdonságai és a kívánt eszköznév közti összerendelést adják meg. Az összerendelés megadásához számos kulcsadatot kérdezhetünk le az eszköztől. Ha az *udev.rules* állományban nem talál megfelelő bejegyzést, az alapértelmezett rendszermag nevet fogja felhasználni. Az udev által megértett kulcsok listáját alább olvashatjuk:

- **BUS:** az eszköz busztípusára keres; példák: PCI, USB vagy SCSI.
- **KERNEL:** a rendszermag által adott névre keres.
- **ID:** a busz eszközszáma; például a PCI busz ID vagy az USB eszköz ID.
- **PLACE:** A buszon elfoglalt topológiai helyre keres, mint például a fizikai kapu amelyre az USB eszközt csatlakoztattuk.
- **SYSFS_fájlnev**, **SYSFS{ fájlnev}** : segítségével az udev bármilyen sysfs eszközjellemezőt kikereshet, például a címkét, gyártót, az USB sorozatszámot vagy a SCSI UUID értéket. Egyetlen szabályban maximum öt sysfs állományt vizsgálhatunk, melyek mindegyikének teljesülnie kell a szabály érvényesüléséhez.
- **PROGRAM:** az udev külső programot is meghívhat és ellenőrizheti az eredményt. A kulcs a program sikeres visszatérése esetén lesz érvényes. A program által visszaadott karaktersorozatot aztán tovább vizsgálhatjuk a **RESULT** kulccsal.
- **RESULT:** Az utolsó PROGRAM hívás eredményét teszti. Ezt a kulcsot a PROGRAM hívás utáni bármelyik kulcsban használhatjuk.

A különféle kulcsok után a **NAME** és az elhagyható **SYMLINK** szavakat adhatjuk meg. A szabály teljesülésekor az udev

a **NAME** részben megadott nevet használja fel az eszköz nevéként, míg a **SYMLINK**, amennyiben létezik, a szükséges további közvetett hivatkozásokat határozza meg. Szóköz karakterekkel elválasztva egyszerre több hivatkozás is megadható. A */dev* könyvtár egyszerű szerkezetének megtartása érdekében a **NAME** és a **SYMLINK** állományok könyvtárneveket is tartalmazhatnak.

Példa

Térjünk kicsit vissza a két nyomtatós példánkhoz. Ha mindkét nyomtatót egyedi névvel szeretnénk ellátni, a következő két udev szabályt használhatjuk:

```
BUS="usb", SYSFS_serial="w09090207101241330",
↳ NAME="lp_color"
BUS="usb", SYSFS_serial="HXOLL0012202323480",
↳ NAME="lp_plain"
```

A fenti szabályok hatására az udev mindkét nyomtatót megpróbálja megkeresni a sysfs fájlisorozatában, majd a fájl értékétől függően *lp_color* vagy *lp_plain* névvel látja el az eszközt. Ezáltal, függetlenül attól, melyiket csatlakoztattuk előbb mindkét nyomtató ugyanazt az állandó nevet kapja. Az sem okoz zavart, ha más USB nyomtatót is csatlakoztatunk.

Különleges udev szabályok

Az udev alatt a *udev.rules* állomány **NAME**, **SYMLINK** és **PROGRAM** mezőinek megadásához különféle printf-szerű szöveg helyettesítést is használhatunk. A használható mezők a következők:

- **%n:** az eszköz rendszermag száma; például az sda3 eszköz rendszermag száma 3.
- **%k:** az eszköz rendszermag-neve.
- **%M:** az eszköz rendszermag szerinti major száma.
- **%m:** az eszköz rendszermag szerinti minor száma.
- **%b:** az eszköz sín azonosítója.
- **%c:** a PROGRAM által visszaadott szöveg. A szövegből kiemelhetjük valamelyik szót, ha a módosítót számmal egészítjük ki. Ez a mező magától értetődő módon a PROGRAM mezőben nem működik.
- **%%:** maga a % karakter.

Ezen kívül bizonyos kulcsokkal héjprogramszerű mintakeresést végezhetünk. A keresőminták a következők:

- *****: nulla, egy vagy több karaktert keres.
- **?**: egy darab tetszőleges karakter, amely nem lehet nemlétező (nulla hosszú).
- **[]**: a zárójelek közt felsorolt bármelyik karakter; például a **tty[SR]** minta egyaránt megtalálná a **ttys** és a **ttyR** szöveget. A keresés a **-** jel használatával támogatja a tartományokat is. Például ha valamilyen számot keresünk, a **[0-9]** mintát használjuk. Amennyiben a **[** jelet a **!** karakter követi, csak az itt fel nem sorolt karakter lesz érvényes találat.

A fent bemutatott szöveg helyettesítési és szövegkeresési módszerek alkalmazása, valamint az, hogy az udev tetszőleges programot képes futtatni majd eredményét felhasználni, rendkívül rugalmas eszközzé teszik az eszköznév kiosztás terén. E hatékonyság érzékeltetésére nézzük meg a következő szabályt:


```
KERNEL="[hs]d[a-z]", PROGRAM="name_cdrom.pl %M %m",
↳ NAME="%1c", SYMLINK="cdrom"
```

A fenti szabály bármely blokkos eszközt megtalál, majd az eszköz major és minor számával meghívja a name_cdrom.pl Perl parancsfájlt. Ha a program sikeresen lefutott, az udev a kimenet első szavát használja fel eszköznévként, majd létrehozza a cdrom közvetett hivatkozást.

A name_cdrom.pl parancsfájlt megtaláljuk az udev terjesztésben. A parancsfájl feladat a következő: megállapítja, hogy az eszköz CD-ROM eszköz-e. Amennyiben igen, a Free CDDDB adatbázisra küldött lekérdezéssel ellenőrzi, hogy az eszközben használt CD-ROM létezik-e az adatbázisban. Amennyiben létezik, ennek alapján nevezi el a CD-t. Például a saját /dev könyvtárban a szabály alkalmazását követően a következőképpen néz ki:

```
$ ls -l /dev/s* /dev/cdrom
brw----- 1 root root 22, 64 Feb 15 08:26
↳ /dev/Samiam-Astray
lrwxrwxrwx 1 root root 8 Feb 15 08:26
↳ /dev/cdrom ->
/dev/Samiam-Astray
```

Megfigyelhettük hogyan képes az udev internetes adatbázisok alapján elnevezni az eszközüket.

Igen, ez valóban kicsit őrült névhasználati szabály, de jól példázza mire képes és milyen rugalmas tud lenni az udev.

Köszönetnyilvánítás

A szerző szeretne köszönetet mondani *Daniel Stekloff*-nak az IBM munkatársának, aki segített formába önteni az udev rendszert. Az ő kitalálása nélkül az udev nem jöhetett volna létre. Valamint *Kay Sievers*-nek, aki az udev különleges képességeinek kialakításában játszott fontos szerepet, különös tekintettel a karaktersorozat módosítókra és mintakeresésre, melyek nélkülözhetetlenek az udev éles használatához. Segítése nélkül az udev közel sem lenne olyan hatékony és használható mint amilyen manapság. *Pat Mochel* sysfs és meghajtó modell-magja nélkül az udev szintén nem lett volna megvalósítható. A szerző lekötelezve érzi magát, hogy magára vállalta amit a legtöbben megvalósíthatatlannak tartottak és, hogy lehetővé tette másoknak az egységes keretrendszerre való építkezést, lehetővé téve minden felhasználónak a hogy átlássák a rendszermagban használt „belőtt pók szötte hálót” („Web woven by a spider on drugs”, a rendszermag-meghajtókat nyilvántartó adatszerkezet neve – a ford.). A cikk az 2002-es Ottawa-i Linux Symposium udev témájú publikációja nyomán készült (lásd a forrásokat).

Linux Journal 2004. június, 122. szám



Greg Kroah-Hartman jelenleg több különféle meghajtó alrendszer Linux rendszermag gazdája. Az IBM-nél dolgozik, ahol Linux rendszermaggal kapcsolatos dolgokkal foglalkozik és a greg@kroah.com címen érhető el.

