UDev - informacje ogólne

Położenie: (nie dotyczy)

Trochę historii

Dawniej w katalogu */dev* znajdowały się specjalne pliki oznaczające urządzenia **blokowe** (odczyt i zapis są buforowane) i urządzenia **znakowe** (odczyt i zapis nie są buforowane), np. /dev/hda. Każdy z tych plików posiadał dwie cyfry, tzw. *major-minor pair* (major to główny typ urządzenia, np. dysk twardy, a *minor* to szczegółowy typ, np. partycja na tym dysku), które były wskazówką dla kernela, gdzie szukać fizycznego urządzenia wskazywanego przez plik. Problem w tym, że znajdowały się tam pliki wszystkich możliwych urządzeń (nawet tych nie podłączonych), co powodowało pewne problemy w zarządzaniu nimi. Były też trudności w przypadku nowego technologicznie urządzenia: trzeba było je dodawać „ręcznie" za pomocą polecenia *makedev*.

Problem miał rozwiązać mechanizm *Devfs* (plik konfiguracyjny znajdował się w */etc/devfs.conf*). Tworzył on pliki tylko istniejących fizycznych urządzeń i umożliwiał automatyczne dodanie nowych w czasie działania systemu. Nie był już potrzebny *major-minor pair*. Dla przejrzystości urządzenia grupowane były w katalogach, np. tam, gdzie pierwotnie mieliśmy do czynienia z */dev/hda4*, po wprowadzeniu *Devfs* było: */dev/ide/host0/bus0/target0/lun0/part4*.

Dla zachowania kompatybilności ze starymi programami, *devfsd* tworzył przy starcie także stare pliki w starym schemacie, ale wskazywały one nie bezpośrednio na urządzenia fizyczne, lecz były symlinkami do nowych plików urządzeń w systemie *devfs*. Przy ładowaniu nowego modułu, *devfsd* automatycznie tworzył nowy plik urządzenia.

Obecnie ten mechanizm został zastąpiony przez *udev*.

Jak działa udev?

Jądro (a w zasadzie moduły jądra) za pomocą funkcji *kobject_uevent_env* oraz *kobject_uevent* wysyła informacje o podpiętych urządzeniach przez *Netlink socket*[1] (komunikacja dwukierunkowa za pomocą mechanizmu AF_NETLINK[2]) do pliku */sys/*/uevent* (strefa *userspace*). *Udev*[3] (zwane „*user helper*") zczytuje informacje z tego pliku za pomocą mechanizmu *inotify* i (na przykład) tworzy odpowiednie urządzenia w */dev*. Programy, które chcą komunikować się z jądrem, korzystają z biblioteki „*libnl*":
 # emerge libnl
Istnieją obecnie dwa mechanizmy *hotplug* (wzajemnie wykluczające się):

- *usermode_helper / uevent_helper(/sbin/hotplug)*[4]*;*
- *netlink (preferowany).*

Aby sprawdzić, czy netlink działa, należy:
 # ls /proc/net/ *(powinien tam być „netlink")*
 # cat /proc/net/protocols *(powinien być na wykazie)*
 # dmesg | grep netlink
Oprócz dynamicznego tworzenia urządzeń, *udev* korzysta także z mechanizmu *coldplug* (statyczne) uruchamianego przy starcie (runlevel „*boot*"). *Udev* jest wstecznie kompatybilny z *devfs*, tzn. katalog */dev* jest statyczny, żebyśmy nigdy nie doświadczyli braku jakiegoś węzła urządzenia. Aby już teraz wyłączyć statyczny *DevFS*, należy ustawić następującą opcję w pliku */etc/conf.d/rc*:
RC_DEVICE_TARBALL="no"
Można także ustawić parametr startowy przy bootowaniu: *gentoo=nodevfs.*

1 Dawniej był to mechanizm *hotplug, system calls, ioctls* lub *proc.* Wszystkie te mechanizmy były skomplikowane i pracowały w trybie synchronicznym, podczas gdy *Netlink* używa trybu asynchronicznego (jest wielowątkowy). Zaletą *Netlink* jest także *multicasting* (wysyła tę samą wiadomość do wielu klientów) oraz buforowanie I/O,

2 AF jest skrótem od „*Address Family*".

3 Napisany został w języku C.

4 Jest to stary niezalecany mechanizm (uruchamia dużą ilość procesów podczas bootowania), który można uaktywnić w jądrze: *Device Drivers → Generic Drivers Option → Support for uevent helper.* Aby dokładnie określić ścieżkę do mechanizmu *uevent_helper*, należy wpisać ją do:
echo /sbin/hotplug > /proc/sys/kernel/hotplug
echo /sbin/hotplug > /sys/kernel/uevent_helper (lub przypisać ścieżkę bezpośrednio w konfiguracji jądra).

Konfiguracja jądra

File systems --->
Pseudo Filesystem --->
[*] /dev/pts file system for Unix98 PTYs
[*] /dev file system support (OBSOLETE)
[ ] Automatically mount at boot (powinna być wyłączona)
Device Drivers --->
Generic Driver Option --->
[*] Maintain a devtmpfs filesystem to mount at /dev (opcja CONFIG_DEVTMPFS=y)

Inne ustawienia:
 # cat /usr/src/linux/.config
 # General Setup
CONFIG_HOTPLUG=y
 # Networking option
CONFIG_NET=y
CONFIG_UNIX=y (Unix Domain Sockets)
CONFIG_NETFILTER_NETLINK=y
CONFIG_NETFILTER_NETLINK_QUEUE=y
 # Pseudo Filesystems
CONFIG_PROC_FS=y
CONFIG_SYSFS=y
CONFIG_SYSFS_DEPRECATED=n*
CONFIG_TMPFS=y
CONFIG_TMPFS_POSIX_ACL=y
 # Inne
CONFIG_UEVENT_HELPER_PATH=""
CONFIG_INOTIFY=y
CONFIG_SIGNALFD=y

Inne informacje

1. Jeśli występują problemy z *Udev*, należy zaktualizować pakiet *baselayout*.

2. Od wersji 1.97, *udev* ma nową metodę nadawania nazw interfejsom sieciowym. Robi to za pomocą pliku */etc/udev/rules.d/80-net-name-slot.rules* (domyślnie zawiera on tylko komentarz). Jeśli chemy go uaktywnić, możemy wypełnić go treścią poprzez:

 # cp /lib/udev/rules.d/80-net-name-slot.rules /etc/udev/rules.d/80-net-name-slot.rules

Jeśli nie chcemy z tego korzystać, można:
 # ln -s /dev/null /etc/udev/rules.d/80-net-name-slot.rules
W tym przypadku prawdopodobnie będą wczytywane reguły z pliku */etc/udev/rules.d/70-my-net-names.rules*

Więcej o nowej konwencji nazw: *http://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames*

Reguły udev

Reguły można tworzyć w plikach umieszczonych w: */etc/udev/rules.d/**
Przykład reguły:
ACTION=="add", DRIVERS=="usb-storage", ATTRS{serial}=="0x00d80029", RUN+="mount /mnt/usb"

Poniższa reguła ma zmieniać domyślną nazwę interfejsu sieciowego (niestety, u mnie nie działa):
ACTION=="add", SUBSYSTEM=="net", ATTR{address}=="00:e0:4c:53:44:58", NAME=="eth0"

Parametr ACTION ma następujące znaczenie:

ACTION==”add” - gdy pojawi się urządzenie (*de facto*: gdy pojawi się wpis spełniający kryteria reguły);
ACTION==”remove” - gdy zniknie urządzenie (*de facto*: gdy istniejący wpis zniknie).

Uwaga: Niestety, „*udevadm monitor*” nie wykrywa wszystkich eventów, np. wpinania wtyczki eth, czy też słuchawek (przyczyna nieznana).

Objaśnienie: Czyli w przypadku dodania urządzenia (akcja „*add*”), za pomocą sterownika „*usb-storage*” wykonana zostanie akcja montowania pendrive'a. Jeśli nie byłoby parametru **ACTION**, reguła byłaby wykonana za każdym razem, gdy spełnione byłyby warunki. Numer seryjny urządzenia możemy sprawdzić za pomocą polecenia:
/usr/bin/udevadm info /dev/usb/lp0
/usr/bin/udevadm info --query=path /dev/sdb1 (sprawdzamy ścieżkę do urządzenia blokowego w /sys)
/usr/bin/udevadm info –query=property /sys/block/sdb/sdb1 (pokazuje numer seryjny i inne atrybuty, według których można identyfikować urządzenie w regułach udev)
udevadm info -a -p /sys/class/net/enp0s20f0u1

Można stosować także:
= nadpisuje wartość nową wartością
+= dodaje nowe działanie / wartość, ale nie usuwa (nie nadpisuje) innych wartości dotyczących tego eventu
:= przypisuje wartość do klucza na stałe (inne reguły nie mogą jej zmienić)
!= sprawdza, czy klucz nie ma wartości
GOTO=”odnośnikDoEtykieta”
LABEL=”nazwaEtykiety”

Aby załadować utworzone reguły:
udevadm control --reload

Żeby uaktywnić je w tej chwili:
udevadm trigger --verbose

Żeby przetestować / zdebugować działanie reguł:
udevadm test /sys/class/net/enp2s0

Monitorowanie podłączanych urządzeń, m.in. czy urządzenie jest podłączane, usuwane, czy zmieniane (*add*, *remove* czy *change*):
udevadm monitor

Wpinanie wtyczki RJ-45

SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="00:e0:4c:53:44:58", RUN="/home/robert/.skrypty/./wifi_off"

SUBSYSTEMS=="usb", ACTION=="remove", ATTRS{idProduct}=="9700", RUN="/home/robert/.skrypty/./wifi_on"

Drukarka USB

SUBSYSTEM=="usb",**ACTION**=="add",**ATTR**{idVendor}=="03f0",**ATTR**{idProduct}=="4117", **RUN**="*/home/adi/.skrypty/hp_1018_firmware*"

SUBSYSTEM=="usb",**ACTION**=="remove",**ENV**{MODALIAS}=="usb:v03F0p4117d0100dc00dsc00dp00ic07isc01ip02in00",**RUN**+="*/home/adi/.skrypty/hp_1018_rm_locker*"

Uwaga: O ile akcja „*add*” akceptowała atrybuty „*idVendor*” (polecenie *lsusb -v*) o tyle akcja „*remove*” bazuje raczej na nazwie urządzenia, w tym wypadku aliasie nazwy (polecenie „*udevadm monitor --property*”), czyli MODALIAS (lub nazwie samej: ID_MODEL).

Różne

Jak sprawdzić, czy kabel jest odłączony?
cat /sys/class/net/enp2s0/operstate

Ostatnia aktualizacja: 7 marca 2023.