Kurs administrowania systemem Linux

Zajęcia nr 5: Podstawowe czynności administracyjne w Linuksie

Instytut Informatyki Uniwersytetu Wrocławskiego

29 marca 2022

Komunikacja międzyprocesowa

Komunikacja pomiędzy programami (lokalna i zdalna)

- pliki zwykłe
- rurociągi (anonimowe lub nazwane, por. `mkfifo(3)`)
- gniazda lokalne i zdalne (zob. `socket(2)`)
- `ipc`: kolejki wiadomości, semafory i pamięć współdzielona
- `rpc`: *Remote Procedure Call*
- D-Bus i in. wysokopoziomowe

Komunikacja programów z jądrem

- wywołania systemowe (zob. rozdz. 2 man)
- pseudosystemy plików (`/proc`, `/sys`, `/dev` itd.)
- gniazda (`netlink(7)`, `rtnetlink(7)` itd.)

Obwoluty

- funkcje biblioteczne (rozdz. 3 man) i programy użytkowe (rozdz. 1 i 8 man) opakowujące wywołania systemowe, dostęp do pseudosystemów plików i gniazd jądra

Pseudosystemy plików

Interfejsy jądra w pseudosystemach plików

- `procfs` — pliki tekstowe z konfiguracją procesów i jądra
- `sysfs` — pliki tekstowe z konfiguracją urządzeń
- `udevfs` — urządzenia jako pliki; zwykle zarządzany przez demona `udevd`

Przykład: jak sprawdzić stan naładowania akumulatora?

- Konsument IT: sprawdzi ikonkę baterii w pasku narzędzi.
- Informatyk: uruchomi jedno z poleceń:

```
acpitool -B
acpi -bi
```

- Guru: napisze skrypt zawierający polecenie:

```
echo $((100 * \
   $(< /sys/class/power_supply/BAT0/energy_now) / \
   $(< /sys/class/power_supply/BAT0/energy_full)))%
```

Ważne katalogi pseudosystemów plików i ich obwoluty

`/proc/[PID]/` — informacje o każdym procesie

- Obwoluty: `ps(1)`, `top(1)` i in.
- Link symboliczny `/proc/self`

`/proc/sys/` — zmienne konfiguracyjne jądra

- Ponad 1000 pseudoplików tekstowych, niektóre do zapisu.
- Obwoluta: `sysctl(8)`.
- Trwała konfiguracja startowa: `/etc/sysctl.{conf,d/}` (zob. `sysctl.conf(5)`).
- Przykład: włączenie przekazywania pakietów IPv4:

  ```
  echo 1 > /proc/sys/net/ipv4/ip_forward
  sysctl net.ipv4.ip_forward=1
  ```

  Trwała konfiguracja: dodać wiersz

  ```
  net.ipv4.ip_forward=1
  ```

  do pliku `/etc/sysctl.conf`.

Por. `proc(5)`.

Informacje o systemach plików i punktach montażowych

`/proc/mounts`

- Lista czynnych punktów montażowych.
- Jeśli jądro obsługuje przestrzenie nazw punktów montażowych, to `/proc/mounts` jest linkiem do `/proc/self/mounts`.
- Dawniej `/etc/mtab` zarządzany przez `mount(8)` (obecnie link symboliczny do `/proc/mounts`).

`mount(8)`

- Wypisuje nieznacznie zmodyfikowaną zawartość pliku `/proc/mounts`.
- Opcja `-l` dodatkowo ujawnia etykiety wolumenów (wymaga praw administratora).
- Konfiguruje punkty montażowe zgodnie z podanymi opcjami.
- Plik konfiguracyjny: `/etc/fstab`.
  Uwaga: to nie jest *bieżąca* konfiguracja!
- `systemd` konfiguruje punkty montażowe za pomocą jednostek montowania (*mount unit configuration*), zob. `systemd.mount(5)`.

Inne programy ujawniające punkty montażowe

`findmnt`

- Wypisuje zamontowane systemy plików.
- Opcja `-A` — wszystkie.

`df`

- Wypisuje zamontowane systemy plików wraz ze statystykami użycia.
- Opcja `-a` — także pseudosystemy.
- Opcja `-h` — *human readable*.

`lsblk`

- Wypisuje urządzenia blokowe (zamontowane bądź nie).
- Opcja `-f`, `--fs` — informacje o systemach plików (wymaga praw administratora).
- Wiele możliwości wyboru informacji i formatowania wyniku.

Programy które odczytują informacje tylko z `/sys` są przeważnie dostępne dla zwykłego użytkownika. Jeśli także odpytują urządzenia, to wymagają uprawnień administratora.

Filesystem Hierarchy Standard

- Definiuje podział *rootfs* w dystrybucjach Linuksa na podkatalogi, ich strukturę, zawartość i przeznaczenie.
- Zarządzany przez Linux Foundation.
- Obecna wersja: 3.0 (2015).
- Bazuje na strukturze katalogów tradycyjnego Uniksa.
- Opis: `hier(7)` (podobnie jak w Uniksach).
- Hierarchia nie jest częścią *Single Unix Specification* (w szczególności POSIX-a). Różnice pomiędzy Uniksami.
- Problemy: przyzwyczajenie jest drugą naturą — sporo dziwolągów odziedziczonych po przodkach (np. `/etc` itp.).
- W celu zachowania kompatybilności sporo linków symbolicznych.
- Większość dystrybucji jest zgodna. Znane wyjątki: GoboLinux i NixOS.

Filesystem Hierarchy Standard, cd.

Rootfs — głowny system plików i pseudosystemy

`/etc/` pliki konfiguracyjne systemu
`/bin/` *istotne* pliki wykonywalne
`/sbin/` *istotne* pliki wykonywalne dla roota
`/lib*/` *istotne* biblioteki współdzielone
`/root/` katalog domowy roota
`/proc/` punkt montażowy `procfs`
`/sys/` punkt montażowy `sysfs`
`/dev/` pliki urządzeń, obecnie punkt montażowy `udevfs`
`/run/` zmienne ulotne, zwykle punkt montażowy `tmpfs`
`/tmp/` pliki tymczasowe, może być punktem montażowym `tmpfs`

Istotne = potrzebne nawet wówczas, gdy inne systemy plików poza *rootfs* nie są dostępne (np. *single user mode*).

Filesystem Hierarchy Standard, cd.

Mogą być niedostępne we wstępnej fazie rozruchu

`/usr/`	część stała plików systemowych, drugi poziom hierarchii
`/usr/local/`	część stała plików systemowych, trzeci poziom hierarchii
`/var/`	część zmienna plików systemowych
`/boot/`	pliki potrzebne do rozruchu (jądro, `initramfs`, bootloader). W Linuksie zwykle punkt montażowy niezaszyfrowanej partycji.
`/home/`	katalogi domowe użytkowników, może być punktem montażowym osobnej partycji
`/media/`	punkty montażowe urządzeń wymiennych
`/mnt/`	tymczasowy punkt montażowy
`/opt/`	oprogramowanie opcjonalne
`/srv/`	dane lokalne używane przez serwery

Przodkowie komputerów osobistych: stacje robocze

Standalone station

- *rootfs* na dysku lokalnym.
- `/home` na dysku sieciowym, uwierzytelnianie zdalne.
- Opcjonalnie: `/usr` na dysku sieciowym — scentralizowane zarządzanie oprogramowaniem. Bardzo popularne dawniej, gdy dyski były małe, a instalowanie i aktualizacja oprogramowania pracochłonne.

Diskless station

- *rootfs* w `tmpfs` (lokalnie w RAM), zwykle kopiowany z serwera TFTP przez *bootloader*.
- `/home` i `/usr` koniecznie na dysku sieciowym.

Thin client

- Lokalna maszyna jest tylko terminalem graficznym.
- Zwykle bezdyskowa. Emulator terminala ładowany z dysku sieciowego.

Drugi poziom hierarchii na osobnej partycji

- Pozwala na montowanie poprzez sieć (NFS).
- Wiele pakietów Debiana nie wspiera późnego montowania `/usr`!

Drugi poziom hierarchii części stałej systemu: /usr/

bin/	pliki wykonywalne
sbin/	pliki wykonywalne dla roota
lib*/	biblioteki współdzielone, inne pliki zależne do arch. lib/ — architektura rodzima, lib*arch*/, np. lib32/ — architektura obca
share/	pliki niezależne od architektury, w tym skrypty
share/doc/	dokumentacja systemu (w różnych formatach)
share/doc-base/	indeks dokumentacji systemu
share/man/	pliki podręcznika systemowego man
share/info/	pliki dokumentacji GNU info
include/	pliki nagłówkowe (niezbędne do kompilacji)
src/	pakiety źródłowe oprogramowania
games/	praca to nie wszystko
local/	trzeci poziom hierachii

Każdy zainstalowany pakiet ma zwykle własne podkatalogi w /usr/share/ i /usr/share/doc/.

Gdzie co się instaluje?

Rodzaje programów

1. Programy istotne `/{,s}bin/`.
2. Programy należące do systemu operacyjnego (w Linuksie brak). W BSD `/usr/{,s}bin/`.
3. Oprogramowanie zarządzane systemem portów/pakietów. W Linuksie `/usr/{,s}bin/`. W BSD `/usr/local/{,s}bin/`.
4. Oprogramowanie instalowane poza systemem portów/pakietów: `/opt/{,s}bin/`. W Linuksie częściej `/usr/local/{,s}bin/`.

Jak zapytać w Debianie?

- Kto zainstalował dany plik: `dpkg -S` *`plik`*
- Jakie pliki instaluje dany pakiet: `dpkg -L` *`pakiet`*
- Każdy plik jest instalowany przez dokładnie jeden pakiet (por. `dpkg-divert`).

Część zmienna systemu: /var/

backups/	backupy ważnych baz danych systemu (passwd itp., bazy systemu pakietów)
cache/	kopie tymczasowe
lib/	dane tworzone przez programy
log/	logi systemowe
mail/	pliki poczty elektronicznej (zwykle jeden mbox na użytkownika)
spool/	kolejki danych różnych programów (np. demona drukowania)
tmp/	pliki tymczasowe o dłuższym czasie życia niż w /tmp
opt/	dane oprogramowania opcjonalnego
local/	dane oprogramowania lokalnego
lock/	obecnie link do /run/lock
run/	obecnie link do /run

Hierarchia plików użytkownika

Rozwiązanie klasyczne

- Plik konfiguracyjny programu *prog* znajduje się w pliku `~/.`*prog*`rc` (`rc` = *run commands*).
- Dane programu *prog* znajdują się w katalogu `~/.`*prog*`/`
- Ewentualnie katalogi `~/.tmp/`, `~/.bin/` (i dostosowane zmienne `TEMP` oraz `PATH`).
- Aplikacje okienkowe: katalog `~/.thumbnails/`, pliki `~/.Xdefaults`, `~/.Xresources` itp.

Rozwiązanie współcześnie lansowane

- Odwzorowuje część hierarchii plików systemu.
- Pliki programów: `~/.local/share/`*prog*`/`
- Konfiguracja programów: `~/.config/`*prog*`/`
- Pliki tymczasowe programów: `~/.cache/`*prog*`/`
- Jak zwykle zachowanie kompatybilności wymaga utworzenia masy linków symbolicznych.

Katalogi XDG

- Katalogi DESKTOP, DOWNLOAD, TEMPLATES, PUBLIC, SHARE, DOCUMENTS, MUSIC, PICTURES i VIDEOS, zwykle ~/Desktop/, ~/Downloads/, ~/Templates/, ~/Public/, ~/Share/, ~/Documents/, ~/Music/, ~/Pictures/ i ~/Videos/ używane przez wiele aplikacji okienkowych.
- Pliki z ~/Desktop/ są wyświetlane na pulpicie wielu systemów okienkowych.
- Przeglądarki WWW zapisują pliki z Internetu do ~/Downloads/.
- Edytory tekstu zapisują pliki do ~/Documents/.
- Pliki z ~/Public/ są udostępniane przez serwery WWW i Samba.
- Zob. xdg-user-dir(1).
  Domyślna konfiguracja w /etc/xdg/user-dirs.defaults.
  Lokalna w ~/.config/user-dirs.dirs.
- Moje rozwiązanie: pojedynczy katalog ~/MyFiles/.

System plików (*rootfs*)

Składniki systemu plików Uniksa

- *Namespace*: mechanizm nazywania i hierarchicznego grupowania obiektów (to już omówiliśmy).
- *Access control*: mechanizmy ograniczania dostępu do obiektów.
- *API*: sposoby komunikacji użytkownika z jądrem w celu uzyskania informacji o i dostępu do obiektów.
- Implementacja.

Montowanie systemów plików

Montowanie

`mount -t` *`typ`* `-o` *`opcje urządzenie katalog`*

Typ zwykle wykrywany automatycznie.

Konfiguracja punktów montowania w `/etc/fstab`

`urządzenie katalog typ opcje backup fsck`

- Wpis w `/etc/fstab` pozwala montować podając tylko katalog lub urządzenie.
- Dodatkowy katalog `/etc/fstab.d/` — po jednym pliku na urządzenie.
- *`backup`* = 1: `dump(8)` robi backup, 0: nie robi.
- *`fsck`* — czy wołać `fsck(8)` = 2: tak, 0: nie, 1: *rootfs*.

Identyfikacja urządzenia podczas montowania

- Ścieżka dostępu do urządzenia, np. `/dev/nvme0n1`.
- UUID systemu plików, np. `UUID=ce4e0e19-1e52-cf948-ad434c55ce294de`.
- etykieta systemu plików, np. `LABEL=Backup_disk` (uwaga na spacje w nazwach etykiet!)
- Narzędzia: `blkid`, `lsblk -f`, `findfs`.
- Zalety `UUID`/`LABEL`: wolne od hazardów czasowych przy inicjalizacji systemu. Nie zależy od kolejności podłączania urządzeń USB itp.