

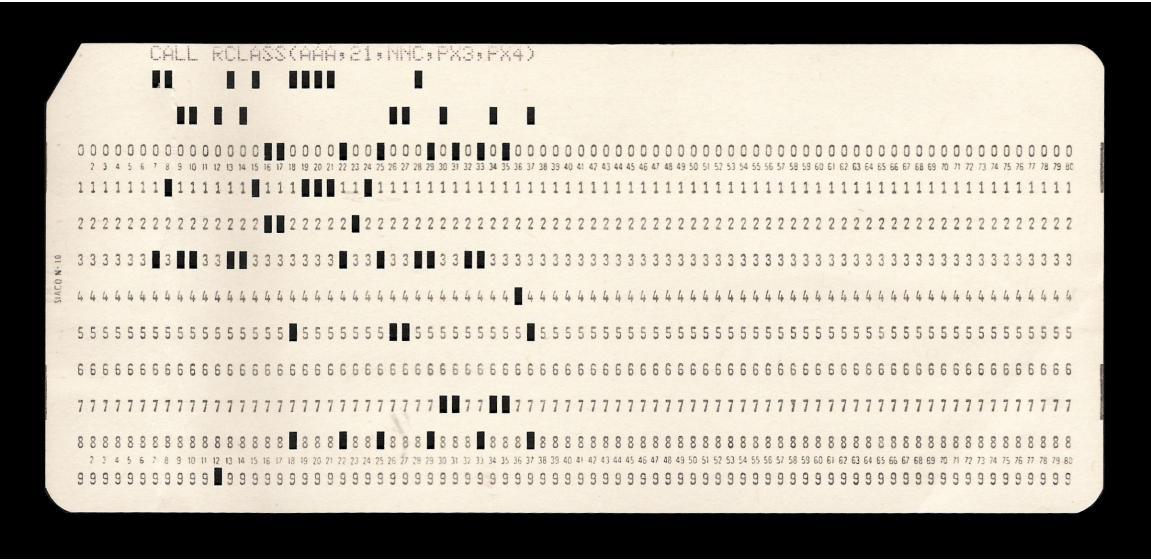
Kurs administrowania systemem Linux

Zajęcia nr 5: Podstawowe czynności administracyjne w Linuksie

Instytut Informatyki Uniwersytetu Wrocławskiego

25 marca 2024

Rekord zapisany na karcie Holleritha



Plik jest wirtualizacją dysku

Wiele mechanizmów w komputerach uległo *wirtualizacji* (np. proces jest wynikiem wirtualizacji CPU itp.)

Dawniej (lata 1960-te)

- Rekord — zapis na pojedynczej karcie perforowanej.
- Zbiór rekordów — plik kart.
- Pamięć bębnowa, potem dyskowa — bufor przechowujący plik kart.
- Pamięć dyskową zaczęto nazywać *dyskiem*.
- IBM VSAM (Virtual Storage Access Method) — możliwość jednoczesnego przechowywania kilku plików (kart) na jednym dysku — wirtualizacja dysku.
- Pojedynczy wirtualny dysk przechowujący plik rekordów (kart) zaczęto nazywać *plikem*.
- Koncepcja *katalogu* (*kartoteki*) — sposób organizacji plików na dysku.

Filozofia Uniksa: wszystko jest plikiem (prawie)

Wszystko jest plikiem

Unix jest *dyskowym systemem operacyjnym*. Przetwarzanie informacji przez jądro i procesy użytkowników polega na otwieraniu (`open(2)`), czytaniu (`read(2)`), pisaniu (`write(2)`) i zamykaniu (`close(2)`) plików. Większość plików jest nazwana, tj. posiada ścieżkę dostępu w drzewie katalogów (*rootfs*).

Rodzaje plików

- zwykłe
- katalogi
- dowiązania symboliczne
- urządzenia blokowe (o dostępie bezpośrednim)
- urządzenia znakowe (o dostępie sekwencyjnym)
- rurociągi (FIFO)
- gniazda domeny Uniksowej (lokalne)

Pliki tekstowe

Tam, gdzie to możliwe, używaj plików tekstowych.

Pliki tekstowe

- dane i wyniki programów
- logi
- pliki konfiguracyjne
- komunikacja między programami

Uniwersalne narzędzia do przetwarzania plików tekstowych

- powłoka (bash)
- edytory (vi, emacs)
- awk, sed, tr i dziesiątki innych narzędzi
- perl (*Practical Extraction and Report Language*)
- inne języki skryptowe

Por. zamknięte formaty binarne: pliki można przetwarzać *tylko* za pomocą dedykowanych programów i *tylko* w zakresie określonym przez te programy.

Obsługa systemów plików w Linuksie

- System plików — wirtualizacja urządzenia blokowego.
- Wiele różnych dyskowych systemów plików, np. `ext{2,3,4}`, `reiser{fs,4}`, `ufs`, `jfs`, `xf`s, `hfs`, `minix`, COW: `btrfs`, `zfs`, log-structured: `logfs`, `nilfs2`, `f2fs`, ROM: `iso9660`, `udf`, `cramfs`, `squashfs`, windowsowe: `FAT{12,16,32}`, `exFAT`, `NTFS` i wiele innych.
- Pseudosystemy plików — „symulowane” przez jądro, nie istnieją na fizycznym urządzeniu blokowym: `sysfs`, `procfs`, `udevfs`, `tmpfs`.
- Uwaga: we FreeBSD `procfs` domyślnie nie jest montowany, bo jest uważany za niebezpieczny. Jest montowany np. w trybie kompatybilności z Linuksem.
- Warstwa abstrakcji: `vfs`.
- W działającym systemie uniksowym: wiele systemów plików *zamontowanych* w różnych miejscach *jednego* drzewa katalogów.
- Por. tradycyjnie w Windows osobne urządzenia A:, B: (napędy dyskietek), C:, D: itd.

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, semaforey 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 (rozd. 3 man) i programy użytkowe (rozd. 1 i 8 man) opakowujące wywołania systemowe, dostęp do pseudosystemów plików i gniazd jądra

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 `udev`

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 przestrzeń 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.

Rootfs — główny 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*).

Mogą być niedostępne we wstępnej fazie rozruchu

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

<code>bin/</code>	pliki wykonywalne
<code>sbin/</code>	pliki wykonywalne dla roota
<code>lib*/</code>	biblioteki współdzielone, inne pliki zależne do arch. <code>lib/</code> — architektura rodzima, <code>libarch/</code> , np. <code>lib32/</code> — architektura obca
<code>share/</code>	pliki niezależne od architektury, w tym skrypty
<code>share/doc/</code>	dokumentacja systemu (w różnych formatach)
<code>share/doc-base/</code>	indeks dokumentacji systemu
<code>share/man/</code>	pliki podręcznika systemowego <code>man</code>
<code>share/info/</code>	pliki dokumentacji GNU <code>info</code>
<code>include/</code>	pliki nagłówkowe (niezbędne do kompilacji)
<code>src/</code>	pakiety źródłowe oprogramowania
<code>games/</code>	praca to nie wszystko
<code>local/</code>	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 `/usr/bin/`.
- 2 Programy należące do systemu operacyjnego (w Linuksie brak). W BSD `/usr/bin/`.
- 3 Oprogramowanie zarządzane systemem portów/pakietów. W Linuksie `/usr/bin/`.
W BSD `/usr/local/bin/`.
- 4 Oprogramowanie instalowane poza systemem portów/pakietów: `/opt/bin/`.
W Linuksie częściej `/usr/local/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/`

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

Rozwiązanie klasyczne

- Plik konfiguracyjny programu *prog* znajduje się w pliku `~/.progrc` (*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/`.