Sieci komputerowe

Wstęp do systemu Linux/UNIX, część I

Ewa Burnecka / Janusz Szwabiński

ewa@ift.uni.wroc.pl / szwabin@ift.uni.wroc.pl

Plan wykładu

- Wstęp do systemu Linux/UNIX
- Modele ISO/OSI i TCP/IP
- Warstwa dostępu do sieci
- Warstwa Internetu
- Warstwa transportowa
- Warstwa aplikacji
- Bezpieczeństwo

Literatura

- Æleen Frisch, *„UNIX. Administracja systemu”*
- Arnold Robbins, *„Sed i awk. Leksykon kieszonkowy”*
- Karol Krysiak, *„Sieci komputerowe. Kompedium”*

(Niepełna) Historia systemu UNIX

1969 K. Thompson, D. Ritchie i J. F. Ossanna przestają pracować nad systemem MULTICS i rozpoczynają pracę nad nowym systemem operacyjnym

1970 B. Kernighan nazywa ten system *Uniplexed Information and Computing System* (UNICS → UNIX)

1971 Rozpoczynają się prace nad językiem C

1973 Jądro Uniksa w języku C (przenośność). System zostaje udostępniony uczelniom

1974 UCB (Uniwersytet Kalifornijski w Berkeley) otrzymuje kopię systemu

- **1975** Powstaje PWB/UNIX 1.0, pierwsza komercyjna wersja Uniksa
- **1977** Bell Labs wydaje UNIX edycja 6. UCB rozsyła ok. 30 kopii swoich udoskonaleń jako 1 BSD
- **1978** 3 BSD zawiera pamięć wirtualną i vi :-)
- **1979** UNIX edycja 7
- **1982** UNIX System III
- **1983** AT&T wypuszcza UNIX System V. Richard Stallman inicjuje projekt GNU
- **1991** Linus Torvalds rozwija jądro Linuksa

Cechy systemu UNIX

- wielozadaniowość
- wieloużytkowość
- przenośność
- bezpieczeństwo
- stabilność
- skalowalność
- wydajność

Logowanie się do systemu

```
Welcome to SuSE Linux 8.0 (i386) - Kernel 2.4.18-4GB

defiant login: szwabin
password:
```

Hasło dostępu

`passwd` - zmiana hasła

Nie używamy:

- fragmentów imienia i nazwiska swojego lub członków rodziny
- imion ulubionych zwierząt
- liczb mających znaczenie w życiu naszym lub naszych bliskich
- nazw ważnych dla nas rzeczy (np. ulubione danie)
- słów znajdujących się w słownikach komputerowych

- nazwisk sławnych ludzi, nazw sławnych miejsc
- tytułów filmów, piosenek itp.
- publikowanych przykładów haseł

Przykład

Tworzymy hasło ze słowa „StarTrek”

złe	dobre	b. dobre
StarTrek	sTarteRk	$$Tarker^T
stArtrEk	stItTruk	st^TrTre#
StarDrek	JeTrekdi	Yetr^Ekd^I
trekstar	sttraerk	st^TraeRk

Uzyskiwanie pomocy

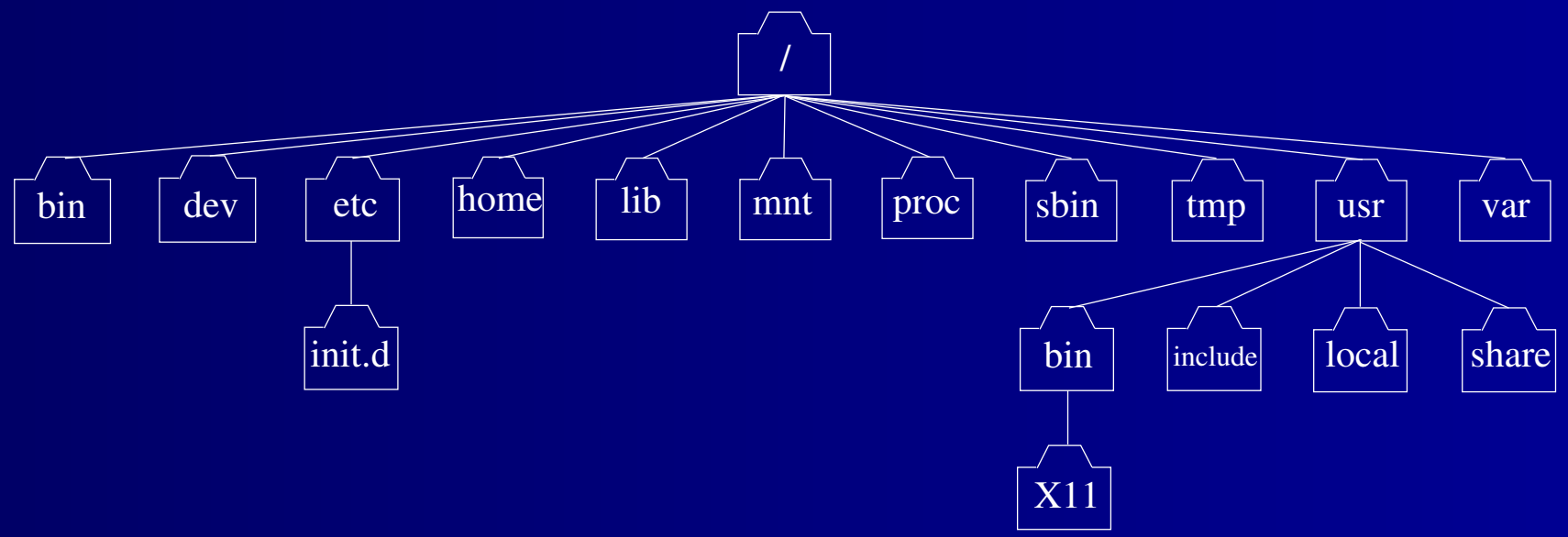
```
man [opcje] nazwa_polecenia
```

Przykład

```
man ls
man 8 intro
man -a intro
man -f cd (whatis cd)
man -k mail (apropos mail)
man -t grep > drukuj.ps
```

Pliki

```
man hier
```



Typy plików:

- „zwykłe” (`-`)
- katalogi (`d`)
- specjalne (`c` lub `b`)
- dowiązania (`l`)
- sockety (`s`)
- nazwane potoki (`p`)

Podstawy nawigacji

- `pwd` wyświetla nazwę bieżącego katalogu
- `cd` zmienia katalog
- `ls` wyświetla zawartość katalogu
- wygodne skróty

.	-	katalog bieżący
..	-	katalog nadrzędny
~	-	katalog domowy
–	-	poprzedni katalog

- `less/more` służą do odczytu plików tekstowych
- `file` pozwala stwierdzić rodzaj pliku

Obsługa `less`

Polecenie	Akcja
Page Up lub **b**	idź stronę do tyłu
Page Down lub **space bar**	idź stronę do przodu
G	idź do końca pliku tekstowego
1G	idź do początku pliku tekstowego
/string	szukaj „string” w tekście
n	powtórz poprzednie przeszukiwanie
q	zakończ program

Operacje na plikach

- `cp` - kopiowanie
- `mv` - przesuwanie
- `rm` - usuwanie
- `mkdir` - tworzenie katalogów
- `ln` - tworzenie dowiązań

Przekierowywanie wejścia/wyjścia

Przykład

```
ls > file_list.txt
ls >> file_list.txt
sort  < file_list.txt
```

Potoki . . .

. . . czyli przekierowanie standardowego wyjścia jednego procesu na standardowe wejście innego

Przykład

```
ls -l | less
ls -lt | head
du | sort -nr
find . -type f -print | wc -l
```

Filtrowanie potoków

`sort`	sortuje standardowe wejście
`uniq`	z posortowanego strumienia danych usuwa duplikaty
`grep`	szuka podanego wzorca w każdej linii na wejściu
`fmt`	formatuje tekst
`pr`	dzieli tekst na strony
`head`	wyświetla kilka pierwszych linii
`tail`	wyświetla kilka ostatnich linii
`tr`	zamienia lub usuwa znaki
`sed`	jak wyżej, tylko bardziej zaawansowany
`awk`	język programowania do tworzenia filtrów

Przykład

```
cat niesformatowany_tekst.txt | fmt | pr | lpr
cat lista.txt | sort | uniq | pr | lpr
tar tzvf archiwum.tgz | less
```

Prawa dostępu do plików

```
-rw-r--r-- 1 szwabin users 5330 wrz 30 23:14 sk-1.tex
```

- grupa właściciela
- właściciel
- prawa dostępu dla pozostałych
- prawa dostępu dla grupy
- prawa dostępu dla właściciela

```
chmod [ugoa][+-=][rwx] nazwa_pliku
```

u	właściciel	r	odczyt
g	grupa właściciela	w	zapis
o	pozostali	x	wykonywanie
a	wszyscy		

Prawa dostępu w postaci numerycznej

- `rwxrwxrwx` $\longrightarrow$ 111 111 111
- `rw-rw-rw-` $\longrightarrow$ 110 110 110
- ...

czyli

- `rwx` $\longrightarrow$ 111 $\longrightarrow$ 7
- `rw-` $\longrightarrow$ 110 $\longrightarrow$ 6
- `r-x` $\longrightarrow$ 101 $\longrightarrow$ 5
- `r--` $\longrightarrow$ 100 $\longrightarrow$ 4

Przykład

```
chmod 600 sk-1.tex
```

777	rwxrwxrwx
755	rwxr-xr-x
700	rwx------
666	rw-rw-rw-
644	rw-r--r--
600	rw-------

Zmiana właściciela i grupy

```
chown nowy_wlasciciel nazwa_pliku
chgrp nowa_grupa nazwa_pliku
```

Uwaga!

- zmienić właściciela może tylko `root` (w systemach typu BSD) lub `root`/właściciel (w systemach typu System V)
- zmienić grupę może właściciel pliku

Rozszerzone prawa dostępu

Kod	Nazwa	Znaczenie
t	sticky bit	zachowanie w pamięci programu po zakończeniu jego wykonywania
s	SUID	ustawienie identyfikatora użytkownika dla procesu w czasie wykonywania
s	GUID	ustawienie identyfikatora grupy dla procesu w czasie wykonywania

Domyślne prawa dostępu

Przykład

```
umask 023
```

$\Rightarrow$ nowe pliki będą miały prawo dostępu $777 - 023 = 754$ (`rwxr-xr--`)

Procesy

- interakcyjne (pierwszoplanowe lub tła)
- kolejkowane
- demony

Atrybuty procesu

- PID (Process ID)
- PPID (Parent Proces ID)
- liczba nice (od -20 do 19)
- TTY
- RUID/EUID
- RGID/EGID

Kontrola procesami

- `bg`
- `fg`
- `jobs`
- `kill`
- `ps`

Polecenie `kill`

- lista możliwych sygnałów: `kill -l`

	Nazwa	Opis
1	SIGHUP	zawieszenie
9	SIGKILL	zabicie (nie może być zignorowany)
15	SIGTERM	przerwanie

Polecenie `ps`

- `ps -u szwabin`
- `ps aux`

USER	PID	% CPU	% MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
szwabin	1659	95.6	0.7	2340	828	pts/3	R	23:13	1:11	./class

USER	nazwa właściciela procesu
PID	numer identyfikacyjny procesu
% CPU	szacunkowy ułamek czasu używania procesora
% MEM	szacunkowy ułamek używanej pamięci
VSZ	wielkość użytej pamięci wirtualnej
RSS	wielkość użytej pamięci rzeczywistej
TTY	terminal kontrolny procesu
STAT	stan procesu
START	czas rozpoczęcia działania procesu
TIME	całkowity użyty czas procesu
COMMAND	wykonywana linia poleceń

- `pstree`
- `top`