Systemy operacyjne

Mariusz Żynel
mariusz@math.uwb.edu.pl
http://math.uwb.edu.pl/~mariusz/

Uniwersytet w Białymstoku

2024/2025

Dostęp do systemu i jego zasobów

- Tylko zarejestrowani użytkownicy mają dostęp do systemu
- Zarejestrowany użytkownik posiada swoją unikalną nazwę i hasło
- Autentykacja, czyli logowanie do systemu, polega na podaniu nazwy użytkownika i hasła, które są weryfikowane przez system
- Uzyskanie dostępu do konkretnych zasobów (plików) wymaga autoryzacji

Użytkownicy, grupy i hasła

`/etc/passwd` – spis kont użytkowników

```
mariusz:x:1001:10:Mariusz Zynel:/export/home/mariusz:/bin/bash
username:password:uid:gid:gcos-field:home-dir:login-shell
```

`/etc/shadow` – spis haseł

```
mariusz:A4e.zrtAKMo8Y:17116::::::
username:password:lastchg:min:max:warn:inactive:expire:flag
```

`/etc/group` – spis grup

```
sys::3:root,bin,adm
groupname:password:gid:user-list
```

Superuser

- Superuser to uprzywilejowany użytkownik zarządzający systemem
- W Unixie
  - Superuser to użytkownik, którego UID to 0
  - Katalogiem domowym superusera jest / stąd nazwa `root`
  - Superuser może wszystko
- Konto superusera używamy wyłącznie wtedy, gdy jest to konieczne
- W Solaris można zalogować się jako `root` wyłącznie z konsoli
- W systemach stosujących model RBAC (Role Based Access Control) zamiast użytkownika jest rola superuser
- Nie wszystkie wieloużytkownikowe systemy operacyjne mają konto superusera, np. Plan 9

Zarządzanie kontami i grupami

`useradd`	dodawanie nowego konta do systemu
`usermod`	modyfikacja definicji istniejącego konta w systemie
`userdel`	usuwanie konta z systemu
`groupadd`	dodawanie nowej grupy do systemu
`groupmod`	modyfikacja definicji istniejącej grupy w systemie
`groupdel`	usuwanie grupy z systemu
`passwd`	zmiana hasła
`passmgmt`	aktualizacja kont użytkowników na niskim poziomie
`pwck`	sprawdzenie poprawności spisu użytkowników `/etc/passwd`
`grpck`	sprawdzenie poprawności spisu grup `/etc/group`

Programy narzędziowe związane z kontami i grupami

`id`	zwraca UID oraz GID użytkownika wraz z ich nazwami
`groups`	zwraca nazwy grup, których użytkownik jest członkiem
`logname`	zwraca nazwę zalogowanego użytkownika
`who`	kto jest zalogowany (who is on the system)
`whodo`	co, kto robi (who is doing what)
`whoami`	zwraca efektywną nazwę użytkownika (`/usr/ucb/whoami`)

Przełączanie użytkowników

`su` (switch user) pozwala zostać innym użytkownikiem bez potrzeby wylogowywania się z systemu

- Jeśli wywołującym jest superuser to nie jest pytany o hasło
- Jeśli hasło jest poprawne, to uruchamiany jest nowy shell z UID, GID oraz dodatkowymi grupami podanego użytkownika
- Domyślna nazwa użytkownika to `root`
- Polecenie `su -` powoduje ustawienie środowiska tak jakby nastąpiło logowanie na konto podanego użytkownika

```
su
su root -c "rm /var/log/syslog.7"
su clamav -c /opt/cfw/bin/freshclam
su - john
```

`sudo` wykonuje podane polecenie z prawami innego użytkownika, nawet nie znając jego hasła, jeśli superuser to skonfigurował

```
sudo su
```

Prawa własności plików

- Każdy plik, a więc także każdy katalog, urządzenie, proces, wszystko w systemie Unix ma swego właściciela
- Właścicielem pliku jest użytkownik oraz grupa
- Polecenie `ls -l` pokazuje właściciela:

  kolumna 3 – użytkownik
  kolumna 4 – grupa

```
sirius$ ls -lh
total 967605
drwxr-xr-x  11 mariusz  staff          11 Aug  5  2010 archive
drwx------   2 mariusz  staff          89 Aug  3  2017 bin
drwxr-xr-x  12 mariusz  staff          12 Dec 21  2014 documents
drwxr-xr-x  17 mariusz  staff          17 Dec 14 23:42 download
drwxr-xr-x  54 mariusz  staff          54 Jun  4  2016 guides
-rw-r--r--   1 mariusz  staff        6.3M Jan  2 20:38 httpd-2.4.29.tar.bz2
drwxr-xr-x  45 mariusz  staff          45 Aug 18  2017 projects
drwxr-xr-x   2 mariusz  staff          46 Dec 28 23:32 public_html
drwx------   9 mariusz  staff           9 Feb 20  2008 tex
```

Zmiana właściciela pliku

`chown` zmienia właściciela wskazanego pliku na podanego użytkownika

```
chown mariusz potwierdzenie.pdf
chown mariusz:staff docs/wniosek-wilno.odt
chown root:bin bin/*.sh
chown -R mysql:mysql /data/mysql
```

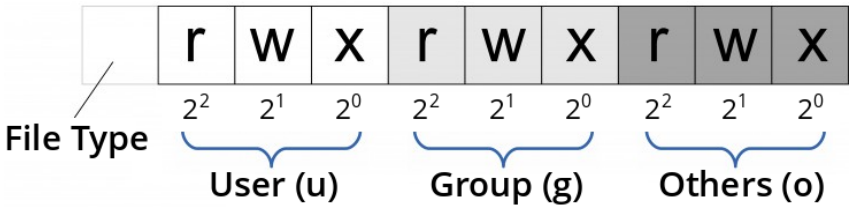
`chgrp` zmienia właściciela wskazanego pliku na podaną grupę

```
chgrp staff docs/wniosek-wilno.odt
chgrp bin bin/*.sh
chgrp -R mysql /data/mysql
```

Prawa dostępu do plików

- Każdy plik, a więc także każdy katalog, urządzenie, proces, wszystko w systemie Unix ma określone prawa dostępu
- Prawa dostępu nie są dziedziczone
- Polecenie `ls -l` pokazuje prawa dostępu w pierwszej kolumnie

```
drwxr-x---   2 root     adm          2 Oct  2 18:35 adm
drwxr-xr-x   2 bin      bin          6 Oct  2 18:56 bin
-r-xr-xr-x   1 root     root         0 Oct  2 18:56 bin/date
```



Dostęp	Symbol	Wartość
odczyt	r	4
zapis	w	2
wykonanie	x	1
brak	-	0

Symbol	Typ pliku
-	zwykły plik
d	katalog
l	dowiązanie symboliczne
b	specjalny plik blokowy
c	specjalny plik znakowy
p	potok FIFO
s	gniazdo (socket)
D	door
P	port zdarzenia

Zmiana praw dostępu

`chmod` zmienia prawa dostępu do wskazanego pliku

```
chmod u=rw,g=r,o=- potwierdzenie.pdf
chmod u+rw,g-rw,o-rwx docs/wniosek-wilno.odt
chmod a+x bin/*.sh

chmod 0644 /var/log/*
chmod u=rw,g=r,o=r /var/log/*

chmod 0755 /var/log
chmod u=rwx,g=rx,o=rx /var/log

chmod -R o+r /var/log
```

Lepkie bity (sticky bit)

- W systemie mogą być wspólne katalogi z prawem zapisu dla wszystkich, np.: `/tmp`, `/var/tmp`, `/var/mail`, w których każdy użytkownik może mieć swoje pliki, ale nie ma uprawnień do usuwania i zmiany nazwy plików pozostałych użytkowników
- Takie katalogi mają ustawioną flagę sticky bit o symbolu `t`

```
sirius$ ls -ld /tmp /var/tmp
drwxrwxrwt   8 root  sys    1368 Feb 25 15:26 /tmp
drwxrwxrwt   8 root  sys   15872 Feb 25 15:45 /var/tmp
```

- Flagę sticky bit ustawia się poleceniem `chmod`

```
chmod +t /tmp/test
chmod -t /tmp/test
chmod 1777 /tmp/test
```

Flagi SUID i SGID

- Niektóre operacje w systemie wymagają wyższych uprawnień niż posiadane przez użytkownika, np.: zmiana hasła, nagranie płyty CD
- Ze względów bezpieczeństwa niektóre operacje wykonywane są z niższymi uprawnieniami, np.: wysyłanie i odbieranie poczty
- Uruchomienie programu z ustawioną flagą SUID powoduje, że proces uzyska efektywne UID właściciela pliku

```
-r-sr-sr-x 1 root sys 22644 sier 7 2009 /usr/bin/passwd
chmod 4555 /usr/bin/passwd
```

```
-rwsr-xr-x 1 root bin 54K maj 22 2008 /usr/bin/cdrw
chmod 4755 /usr/bin/cdrw
```

- Uruchomienie programu z ustawioną flagą SGID powoduje, że proces uzyska efektywne GID właściciela pliku

```
-r-xr-sr-x 1 root smmsp 1,3M lip 24 2015 /usr/sbin/sendmail
chmod 2555 /usr/sbin/sendmail
```

- Ustawienie flagi SGID dla katalogu powoduje, że pliki tworzone w tym katalogu odziedziczą GID tego katalogu

ACL – Access Control List

- Klasyczny mechanizm uprawnień przy bardziej skomplikowanym schemacie praw dostępu okazuje się nieefektywny
- ACL pozwala nadawać uprawnienia indywidualnie
- Dwa standardy ACL: POSIX i NFSv4

```
alpha$ ls -lvd biuro
drwxr-x---+ 99 root     root          146 Feb 24 14:29 biuro
     0:user:ewa:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/delete/read_acl/synchronize
         :file_inherit/dir_inherit:allow
     1:user:wojtek:list_directory/read_data/add_file/write_data
         /add_subdirectory/append_data/read_xattr/write_xattr/execute
         /delete_child/read_attributes/write_attributes/delete/read_acl
         /synchronize:file_inherit/dir_inherit:allow
     2:user:sysadm:list_directory/read_data/add_file/write_data
         /add_subdirectory/append_data/read_xattr/write_xattr/execute
         /delete_child/read_attributes/write_attributes/delete/read_acl
         /write_acl/write_owner/synchronize:file_inherit/dir_inherit:allow
     3:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/read_attributes
         /write_attributes/read_acl/write_acl/write_owner/synchronize:allow
     4:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```