

# System operacyjny Unix; *administrowanie*

- pliki konfiguracyjne Unix-a z katalogu /etc

– plik `"/etc/passwd"`;

zawiera linie następującej postaci:

`name : passwd : uid : gid : gecos : home_dir : shell`

← początkowy shell

← początkowa grupa użytkowników (id grupy)

← zamiast hasła może być "x" co oznacza że  
hasło jest w pliku shadow

przykład:

`/etc/passwd`

```
user1:x:502:503::/home/user1:/bin/bash
user2:x:503:504::/home/user2:/bin/bash
user3:x:504:505::/home/user3:/bin/bash
```

`/etc/shadow`

```
user1:$1$IbhRrsb$LvbSB4fYWEQ1YrxRNWCxY.:12386:0:99999:7:::
user2:$1$nEM0IHuV$SdJR98gE.EHMPVfCYMuDc0:12386:0:99999:7:::
user3:$1$CXtZnR6B$V8W5KbGwcrGA6mX9XJQlQ.:12386:0:99999:7:::
```

# System operacyjny Unix; *administrowanie*

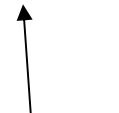
- pliki konfiguracyjne Unix-a z katalogu /etc

- plik "/etc/group";

zawiera linie następującej postaci:

name : passwd : gid : user1, user2, user3 ...

nazwa  
grupy



lista użytkowników należących do grupy  
nie trzeba go podawać jeśli użytkownik jest  
jawnie wymieniony na liście ...

- polecenia związane z grupami:

- groups – wyświetla wszystkie grupy do których należymy
- newgrp *nazwa\_grupy* – zmienia bieżącą grupę (być może zostaniemy zapytani o hasło !)
- chgrp – zmiana grupy właścicieli pliku
- groupadd – dodawanie nowej grupy (tylko *root*; użytkowników grupy dopisywać ręcznie)
- gpasswd – zmiana hasła do grupy
- id – wyświetla uid, gid, oraz grupy do których należymy

- różnica w działaniu grup użytkowników w SV i BSD:

- SysV: w danej chwili użytkownik należy do jednej grupy
- BSD: w danej chwili użytkownik ma prawa przysługujące wszystkim grupom do których należy

# System operacyjny Unix; *administrowanie*

- pliki konfiguracyjne Unix-a z katalogu /etc
  - plik "/etc/group" – eksperyment
    - mamy użytkowników: *user1*, *user2*, *user3*
    - mamy grupy użytkowników:
      - *test1i2* zawierającą użytkowników *user1* i *user2*
      - *test3* zawierającą użytkownika *user3*
      - *test1i3* zawierającą użytkowników *user1* i *user3*
    - mamy pliki

```
-rw-rw-r-- 1 mhanckow test1i2    16 maj 31 08:41 plik1.txt
-rw-rw-r-- 1 mhanckow test1i2     1 maj 31 08:19 plik2.txt
-rw-rw-r-- 1 mhanckow test3       1 maj 31 08:21 plik3.txt
-rw-rw-r-- 1 mhanckow test1i3     1 maj 31 08:21 plik4.txt
```
    - wejść do systemu jako *user1* i zobaczyć jakie operacje możemy wykonywać na plikach *plik?.txt* ...
    - przełączyć się na inną grupę poleceniem *newgrp* (listę bieżących grup wyświetlimy poleceniem *id*)
    - grupy linux-owe działają jak w BSD ?!?!?! (= mamy dostęp wynikający ze wszystkich grup do których należymy wyświetlanych przez *id*)
    - można wchodzić do grup do których "nie należymy" podając hasło !!!

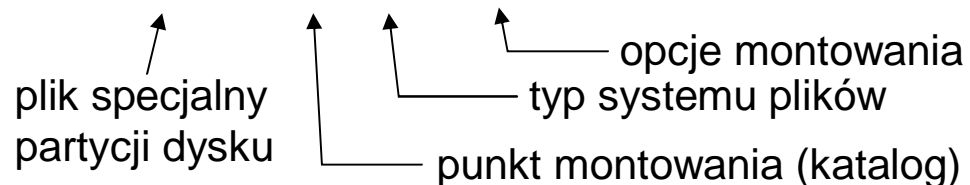
# System operacyjny Unix; *administrowanie*

- pliki konfiguracyjne Unix-a z katalogu /etc

## – plik `"/etc/fstab"`;

zawiera systemy plików montowane podczas uruchamiania;  
każda linia wygląda jak argumenty polecenia "mount"

```
/dev/hda3 /mnt/hda3 ntfs exec,dev,suid,ro,uid=501,gid=501 1 1  
/dev/hda1 / ext2 noatime 1 1
```



# System operacyjny Unix; *administrowanie*

- pliki konfiguracyjne Unix-a z katalogu /etc

## – plik "/etc/inittab";

- plik używany podczas zmiany *poziomu działania* [ang. runlevel]
- określa które procesy (demony) mają być uruchomiona na danym poziomie
- plik inittab jest interpretowany przez proces *init* (pid=1 !)

- każda linia tego pliku wygląda tak:

**id : run levels : action : command**

np:

```
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
```

skrypty uruchamiane przy  
przełączaniu na dany  
poziom działania  
(np. podczas startu systemu)

- poziomy działania:

1 – single user mode (tryb do wykonywania czynności adm)  
2-5 – normalne działanie  
6 – reboot

- zmiana poziomu działania: polecenie "telinit poziom"

... pokazać plik /etc/inittab Linuxa

# System operacyjny Unix; *administrowanie*

- jak użytkownicy są wpuszczani do systemu
  - mamy na myśli użytkowników korzystających z systemu uniksowego poprzez terminale tekstowe ... (Linux: Ctrl+Alt+F?)
  - proces *init* uruchamia procesy *getty* dla każdego terminala

uruchamiany przez proces *init*  
na podstawie pliku */etc/inittab*;  
dla każdego terminala osobny  
proces *getty*

