

# LINUX

**LINUX IS NOT UNIX**

Jądro linuxa

LINUX KERNEL

O'REILLY

Daniel P. Bovet & Marco Cesati

- 1991 – Linus Torvalds
  - Intel 80386, inne architektury: Alpha, SPARC, Motorola MC680x0, PowerPc, IBM System/390
  - Kod źródłowy dostępny w ramach licencji GNU (<http://www.kernel.org>)
- Kod źródłowy Linuksa
  - 4500 plików w C i asemblerze
  - w 270 podkatalogach
  - 2 000 000 linii kodu
  - 58MB na dysku (książka – 2MB)

„Dijkstra prawdopodobnie mnie nienawidzi” [L. Torvalds]

# jądro Linuksa a jądra uniksowe

- Monolityczne – jak większość komercyjnych odmian Uniksa
- Dobre wsparcie dla modułów. Tradycyjne jądra kompilowane i łączone statycznie; dynamiczne ładowanie pewnych modułów (sterowniki urządzeń)
- używa wątków jądra w ograniczonym zakresie. Niektóre nowoczesne jądra bazują na wątkach jądra
- Wsparcie dla aplikacji wielowątkowych (procesy lekkie)
- Jądro bez wywłaszczania
- Wsparcie dla SMP (Symetryczna wieloprocessorowość)
- Standardowy system plików prosty, można przenieść JFS, XFS

# Zalety Linuksa

- Darmowy
- GPL – można modyfikować kody
- Działa na prostych, tanich platformach sprzętowych
- Szybki, w pełni wykorzystuje możliwości sprzętu
- Stabilny – rzadko zdarzają się błędy
- jądro może być bardzo małe
- Kompatybilny z różnymi so
  - Montowanie różnych systemów plików
  - Biblioteki do uruchamiania programów napisanych dla innych systemów
- Dobre wsparcie, szybko dostępne sterowniki

- Wielozadaniowość
- Wielodostęp
- Wieloprocessorowość (od.2.0)

# Wersje jądra Linuksa

## stabilne

- 2.2.14
- 2.6 (2003r)
- 2.6.11.x
- 2.6.39

3.0 (2011)

3.5,....3.16

4.0

5.0

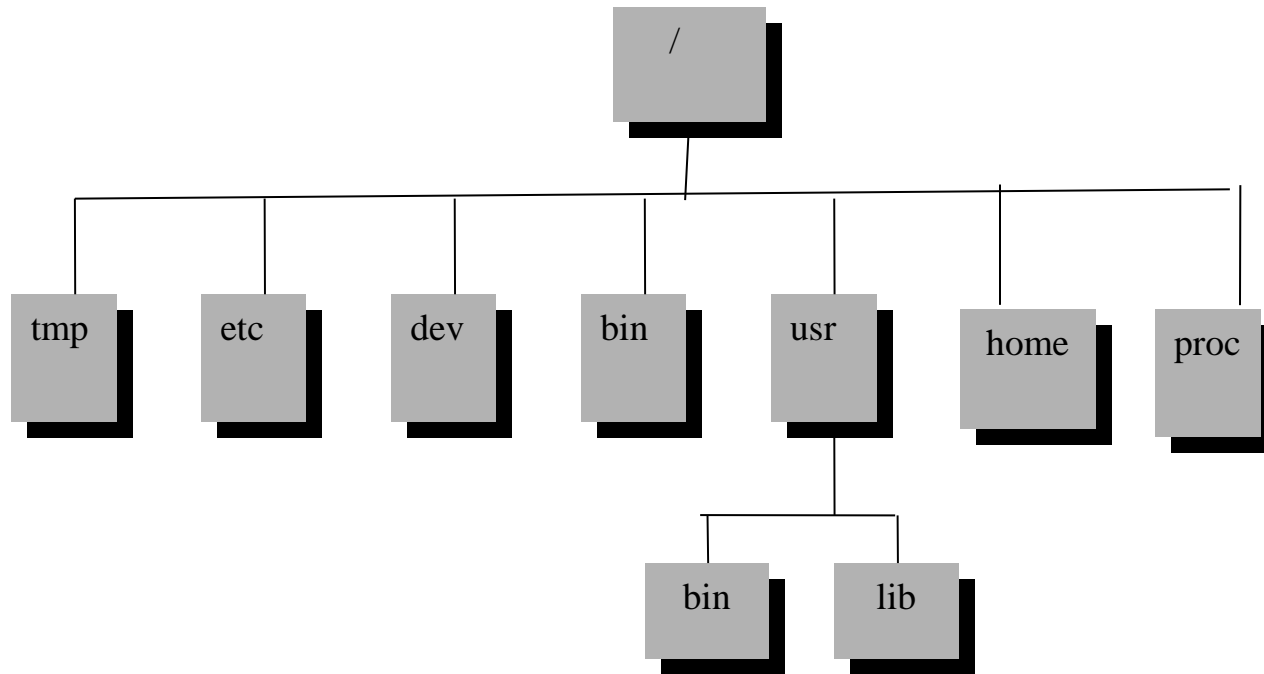
## rozwojowe

2.3.51

3.4-rc1

- Jądro
- Powłoka
- System plików
- Programy użytkowe





## ułatwienia

<tab>

<Ctrl+d>

↑ ↓

pol1;pol2

pol1 \

arg1 \

arg2

~/.....

\$HOME

## pliki ukryte

*.nazwa*

ls -a

## metaznaki

\*

?

[...]

maskowanie \

# polecenie\_opcje\_argumenty

passwd

pwd

clear

cal

who

touch

ls -ld

tree

cd

mkdir

mkdir -p

cat

more

less

head

tail

wc

cut

sort

touch

rm

rmdir

rm -r

mv

cp

cp -r /katalogi


cp a1 a2 a3

-i

```
[anka@mars anka]$ ls -la
```

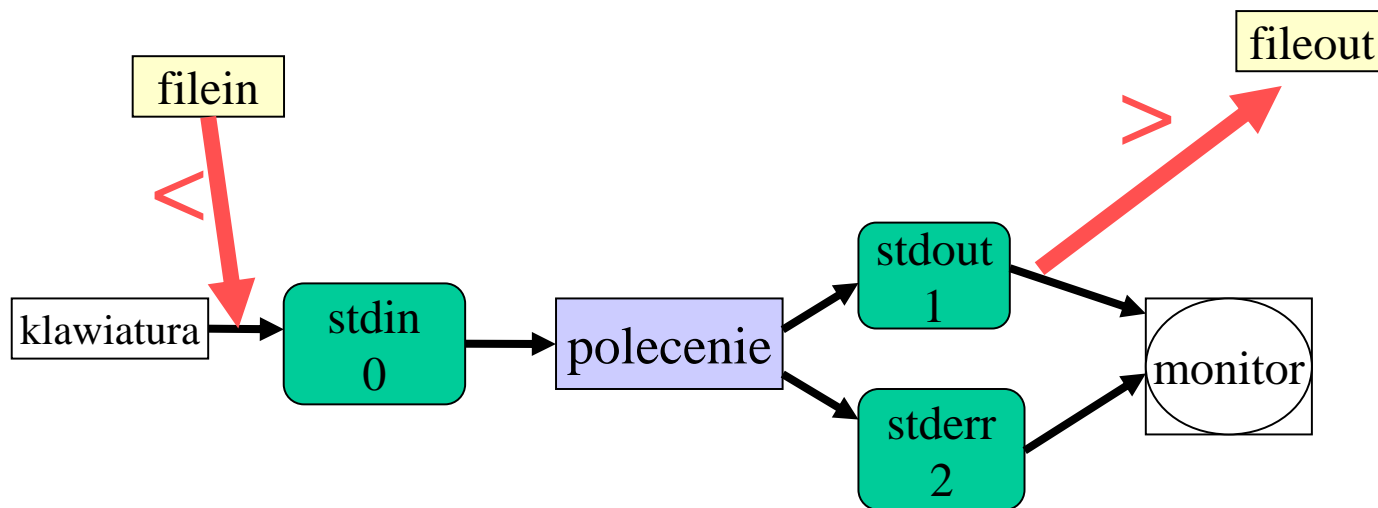
```
total 188
```

drwx-----	12	anka	workers	4096	Sep 15 08:56	.
drwxr-xr-x	22	root	root	4096	Sep 29 09:13	..
-rw-r--r--	1	anka	workers	788	May 17 16:37	A.c
-rw-r--r--	1	anka	workers	0	Mar 12 2004	.addressbook
-rw-----	1	anka	workers	4210	Oct 7 17:38	.bash_history
-rw-r--r--	1	anka	workers	24	Sep 22 2003	.bash_logout
-rw-r--r--	1	anka	workers	191	Sep 22 2003	.bash_profile
-rw-r--r--	1	anka	workers	124	Sep 22 2003	.bashrc
-rw-r--r--	1	anka	workers	755	May 17 16:36	B.c



typ pliku: -, d, b, c, l, s, p

# Przekierowania <, >, >>



cat <plik

cat <plik1 > plik2

cat plik 1>wynik 2>&1

cat>plik

cat plik 2>/dev/null

ls -l >plik /rozmiar pliku

cat>>plik

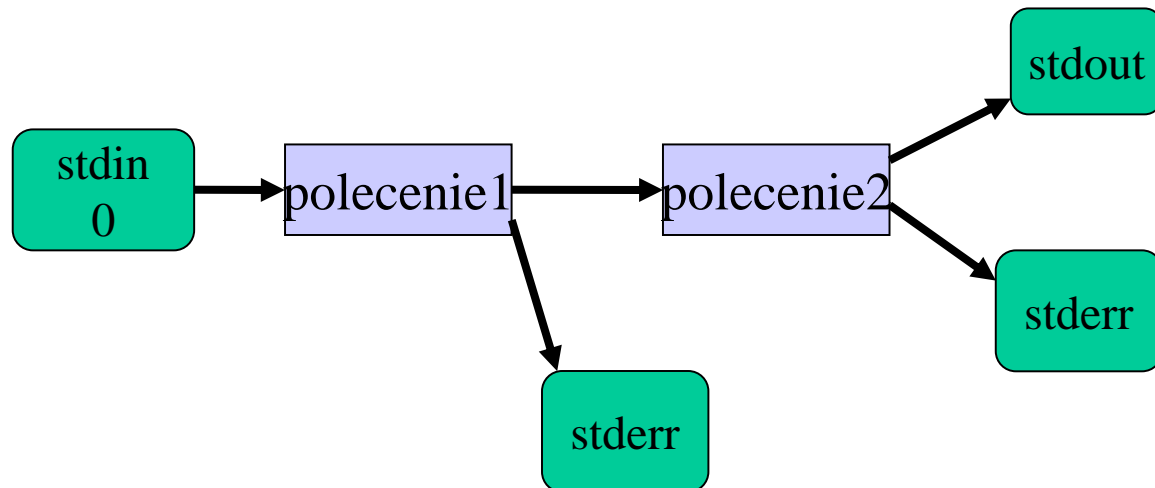
cat plik 1>wynik 2>blad

cat plik>plik

.....>& plik

# Potoki |

polecenie1|polecenie2

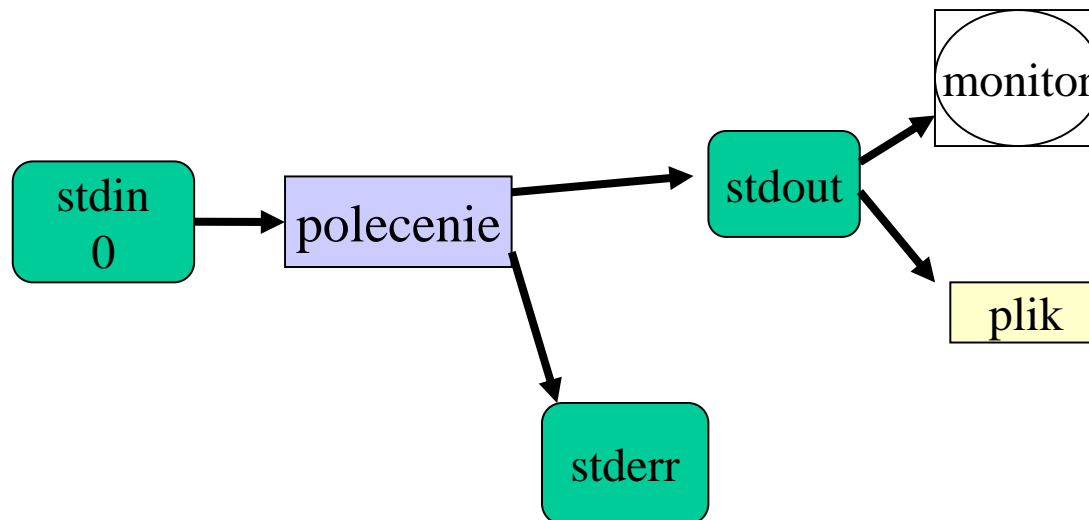


ls -l|more

who|wc -l

# tee

```
polecenie1|tee plik
```



```
cat plik1 plik2|sort|cat -n|tee wynik
```

```
cat plik1 plik2|sort|cat -n>wynik
```

# Porównywanie plików

cmp

diff patch

diff -u plik1 plik2>lata

patch<lata

patch -R <lata



# wyszukiwanie

- Ciągu znaków w pliku(ach)

**grep** opcje wzorzec plik

- Plików wg. atrybutów

**which**

**whereis**

**find** kat\_startowy opcje kryterium

# grep

## wzorzec

.

[a-z]

^..... - początek

.....\$ - koniec

[^...] - negacja

„.....” ‘.....’

\\ - lub

## opcje

-i – ignor. wlk. liter

-v - inwersja

-c – zlicza linie

-n – numeruje linie

```
grep ^[^d-] plik
```

```
ls -l|grep ^[^d-]
```

```
grep -v ^[cC] plik.f>plikbk.f
```

```
grep '\.$' plik
```

```
grep 'int\\|long' *.c
```

# find kat\_start { opcja kryterium }

## Opcje:

- name
- type [dfbcs]
- size [+ -]n[bcwk] np.. -100c
- mtime [+ -]n
- atime
- user
- perm [+ -]sugo s,u,g,o - <0;7>
- exec np. -exec cat { } \;
- ok
- newer

operatory logiczne: -a; -o; !

find . -type l -o -type f

# Pomoc

man

whatis

apropos

info

# katalog

nazwa	nr i-węzła
plik1	1099
plik2	1080
plik3	1065

# i-węzeł

- typ pliku:
  - **-** – plik zwykły,
  - **d** – katalog,
  - **p** – łącze nazwane FIFO,
  - **b** – plik specjalny blokowy,
  - **c** – plik specjalny znakowy,
  - **l** – link symboliczny,
  - **s** – gniazdo (semafor),
- identyfikator właściciela oraz grupy pliku,
- prawa dostępu,
- rozmiar pliku w bajtach/porcjach 512B (32bity);  
pow(2,41) - max. rozmiar pliku
- ostatni czas dostępu, modyfikacji,
- czas utworzenia i skasowania,
- liczba dowiązań,
- liczba bloków dyskowych zajmowanych przez plik,
- adresy dyskowe (12+3)---ext4 – 4 extenty

# linki

- **twarde**

**ln plik linktw** (liczba dowiązań, numery i-węzłów,  
ograniczenia)

- **symboliczne**

**ln -s plik linksymb** (atrybuty linku, link do katalogu, linku)

ln -s kat kat2

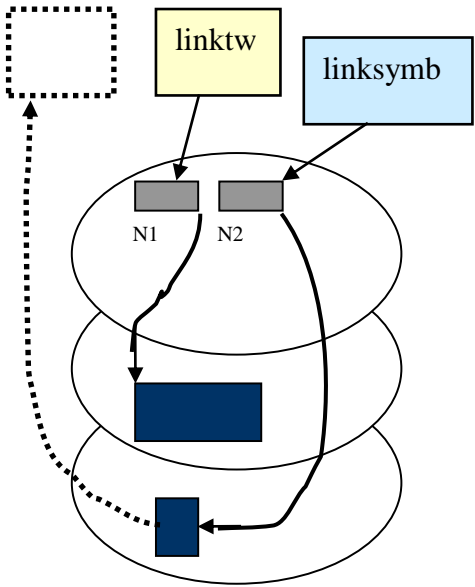
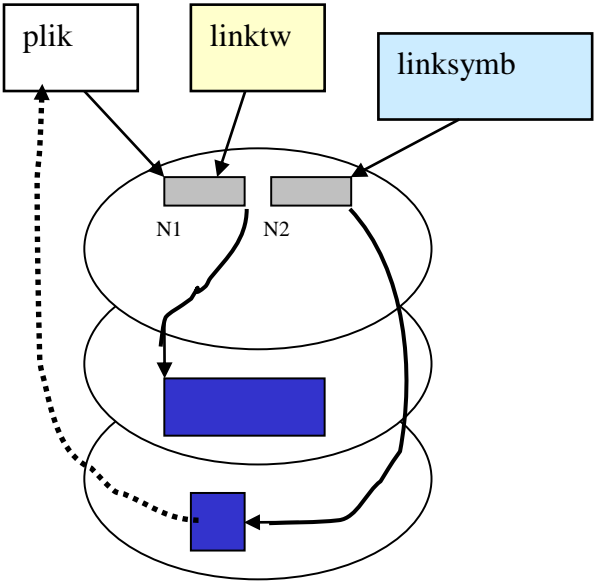
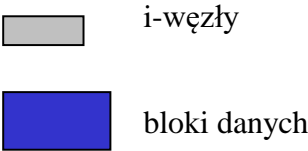
ls kat2

ls -l kat2

cd kat2

In plik linktw

In -s plik linksymb

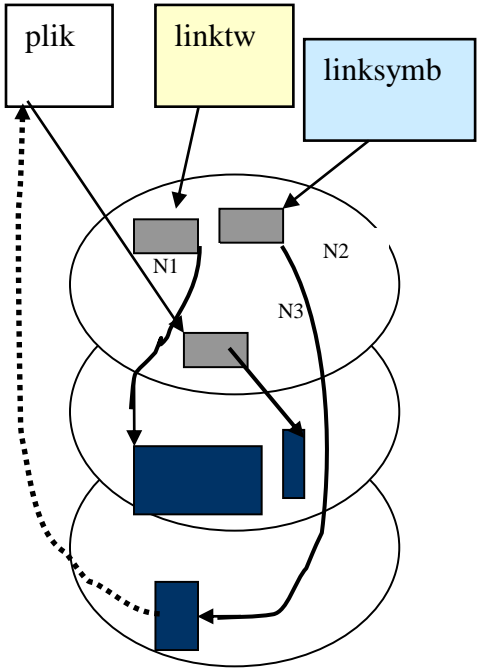


rm plik

cat > plik

.....

<ctrl+d>





<b>atrybut</b>	<b>link twardy</b>	<b>link symboliczny</b>
<b>typ</b>	-	1
<b>rozmiar</b>	jak plik	=długości nazwy
<b>nr i-węzła</b>	jak plik	nowy
<b>ograniczenia</b>	pliki zwykłe w tym samym file systemie	brak

# Prawa dostępu

## DO PLIKU

- r** - oglądanie zawartości pliku, prawo do kopiowania
- w** – zezwolenie na modyfikację zawartości
- x** – prawo do uruchomienia pliku wykonywalnego

## DO KATALOGU

- r** – umożliwia przeszukiwanie zawartości katalogu, jest interpretowane jako prawo wypisywania zawartości (komenda **ls**)
- w** – umożliwia modyfikowanie zawartości katalogu, dodawanie oraz usuwanie plików z katalogu
- x** – pozwala na dostęp do plików zapisanych w nim oraz na wejście do danego katalogu – uczynienie go katalogiem bieżącym (komenda **cd katalog**)

chmod **7777** plik



rozszerzone prawa dostępu do pliku

**rwXrWxrWx**



s

SUID



s

SGID



t

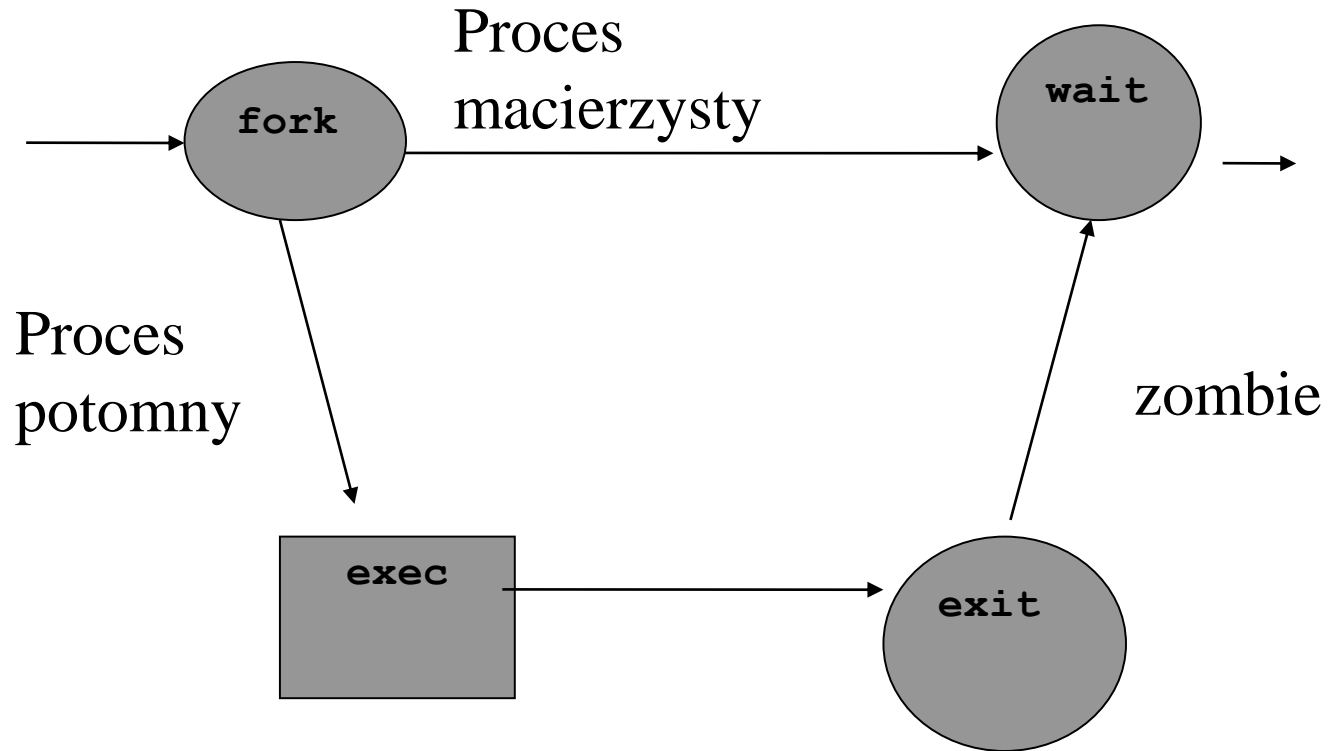
sticky  
bit

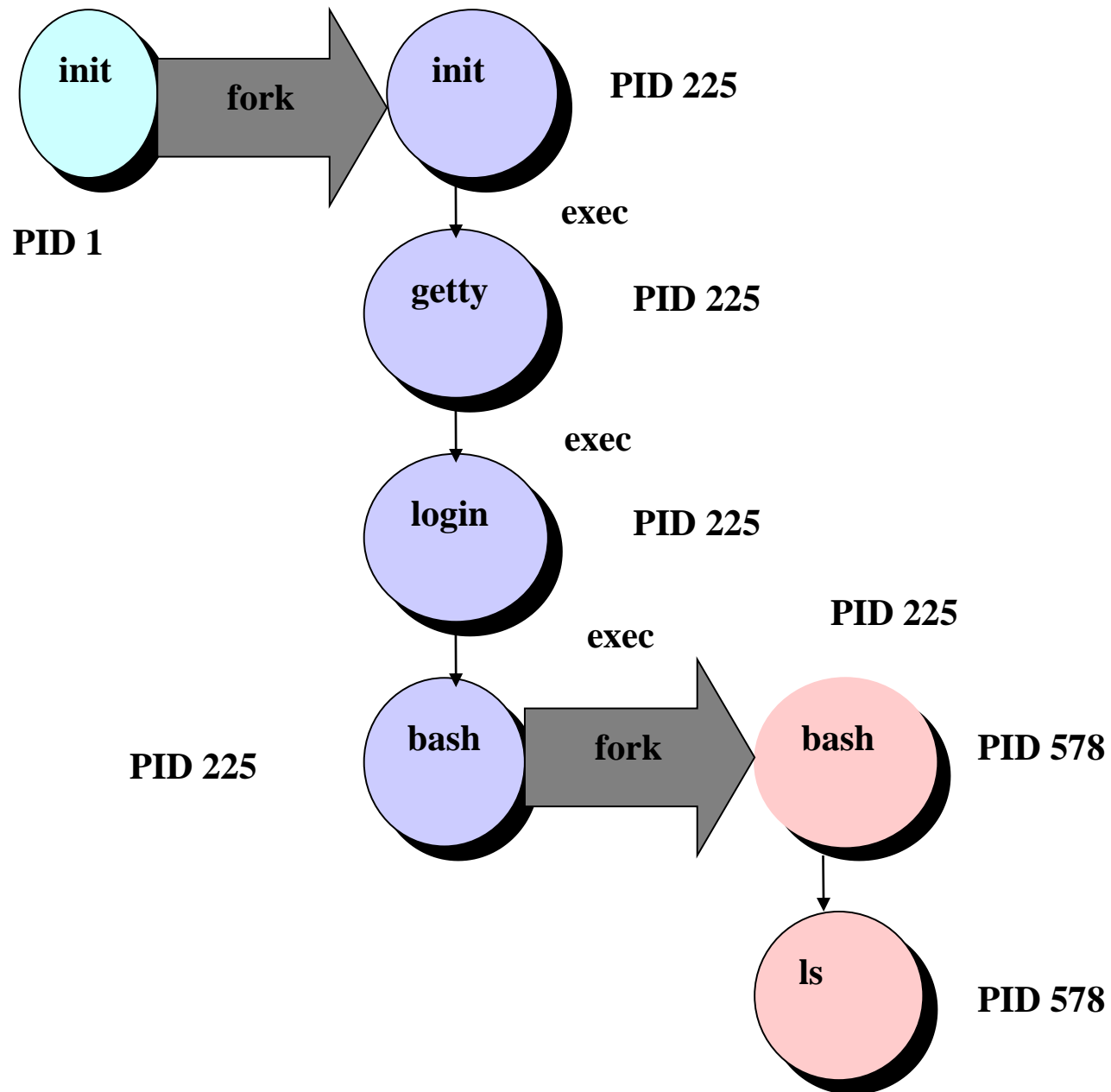
```
chmod u+w,og+r-w plik
```

```
chmod a=rw plik
```

```
chmod o= plik
```

# PROCESY





# Atrybuty procesu

**PID**–identyfikator procesu (process ID)

**PPID**–identyfikator procesu przodka

**UID**–identyfikator użytkownika, który proces uruchomił (jest on właścicielem procesu)

**EUID**–efektywny identyfikator użytkownika (effective user ID), który określa jakie prawa przysługują danemu procesowi

**RUID**–rzeczywisty identyfikator użytkownika (real user ID), który rozpoczął proces; RUID różni się od EUID, jeżeli uruchomiony program miał ustawione rozszerzone prawa dostępu – SUID (s zamiast x na pozycji użytkownika)

**RGID**–rzeczywisty identyfikator grupy użytkownika (real group ID)

**EGID**–efektywny identyfikator grupy użytkownika (effective group); Effective Group ID – różni się od RGID, jeśli uruchomiono program z ustawionym prawem SGID (s zamiast x na pozycji grupy)

**TIME**–czas trwania

**TTY**–terminal

**COM, CMD**–faktyczne polecenie, które uruchomiło proces

**NI**–Liczba **nice** mająca wpływ na priorytet procesu, określa poziom uprzejmości procesu

**PRI**–aktualny priorytet procesu (obliczany dynamicznie)

**SIZE**–wielkość pamięci wirtualnej procesu

**RSS**–wielkość użytej pamięci rzeczywistej

**STIME**–czas rozpoczęcia procesu.

# Atrybuty procesu-cd.

**STAT**—aktualny stan procesu;

**R** – run (działający),

**S** – sleep (uśpiony),

**D** – oczekujący na operację dyskową,

**T** – stopped lub traced (zatrzymany lub śledzony),

**Z** – zombie (proces, który zakończył swoje działanie, zwolnił wszystkie używane zasoby, ale nie otrzymał potwierdzenia przyjęcia sygnału zakończenia od procesu rodzica)

Status dodatkowo może być oznaczony symbolami:

**W**—nie ma strony zaalokowanej w pamięci operacyjnej,

**L** —ma stronę w pamięci,

**<** —wysoki priorytet,

**N** —obniżony priorytet



ps aux

ps l

ps -U *user*

ps -eo pid,ppid,user,ruid,euid,cmd --sort pid

top ( interakcyjne - r,h,k,q)

kill -l - lista sygnałów

kill -9 *pid*

echo \$? /128+*nr sygnału*

# sygnały

- 1 – SIGHUP
- 2 – SIGINT ctrl+c
- 3 – SIGQUIT ctrl+\
- 9 – SIGKILL
- 14 – SIGALARM
- 15 – SIGTERM
- 19 – SIGSTOP ctrl+z

# procesy

- Interakcyjne
  - Pierwszoplanowe
  - Tła
- Wsadowe
- Demony

# Procesy pierwszoplanowe i tła

polecenie &

jobs

fg %*n*

bg %*n*

<ctrl+z> ; bg %

<del> - usuwa proces tła

<ctrl+c> - usuwa proces pierwszoplanowy