

Sygnaly i procesy demony

Polecenie *kill*

kill [opcje] *id_procesu*

- ❑ Niektóre znaki z terminala:
interrupt character Ctrl_C, Del -> SIGINT
quit character Ctrl_\
suspend character Ctrl_Z -> SIGSTP
- ❑ Błędy sprzętowe
- ❑ Błąd operacji zmiennoprzecinkowej - SIGFPE.
- ❑ Niektóre sytuacje wykrywane przez oprogramowanie systemowe.

Funkcja systemowa *kill*

***int* kill(*int* *id_procesu*, *int* *sygnal*)**

- ❑ Jeżeli *id_procesu*=0, to *sygnal* jest wysyłany do wszystkich procesów w grupie nadawcy.
- ❑ Jeżeli *id_procesu*=-1 i nadawca nie jest nadzorcą, to *sygnal* zostanie rozesłany do wszystkich procesów, których rzeczywisty identyfikator użytkownika jest taki sam jak efektywny identyfikator użytkownika procesu nadawcy.
- ❑ Jeżeli *id_procesu*=-1 i nadawca jest nadzorcą, to *sygnal* zostanie rozesłany do wszystkich procesów różnych od procesów systemowych (*id*=0 lub 1).
- ❑ Jeżeli *id_procesu* <-1, to *sygnal* zostanie wysłany do wszystkich procesów, których *id_grupy* jest równy *|id_procesu|* nadawcy.
- ❑ Jeżeli *sygnal*≠0, to następuje sprawdzenie, czy wystąpił błąd. *Sygnal* nie jest wysyłany.

Funkcja systemowa *signal*

```
#include <signal.h>
int (*signal (int sygnal, void (*funkcja) (int))) (int)

signal (SIGUSR1, SIG_IGN);
signal (SIGUSR1, SIG_DFL);

extern void obsluga();

if (signal (SIGINT, SIG_IGN) != SIG_IGN)
    signal (SIGINT, obsluga);
```

Blokowanie sygnałów

```
BSD: int maska;  
maska = sigmask(SIGQUIT) | sigmask(SIGINT)  
  
int sigblock( int maska);  
int sigsetmask( int maska);  
  
int stara_maska;  
stara_maska = sigblock( sigmask(SIGQUIT) |  
                        sigmask(SIGINT));  
  
/*  obszar chroniony */  
sigsetmask(stara_maska);  
  
System V:  
int sighold( int sygnał );  
int sigrelse( int sygnał );
```

Procesy demony

Demon (*daemon*) jest procesem drugoplanowym (nie ma przydzielonego terminala ani *shella*), który czeka na wystąpienie pewnego zdarzenia.

- Zamknij wszystkie otwarte deskryptory plików
- Zmień roboczy katalog bieżący
- Ustaw maskę trybu dostępu do plików
- Działaj jako proces drugoplanowy
- Odłącz się od grupy procesów
- Ignoruj sygnały terminala sterującego
- Odłącz się od terminala sterującego
- Nie odzyskuj terminala sterującego
- Obsłuż sygnał SIGTERM
- Obsłuż sygnał SIGCLD

Budowanie procesu demona

Zamykanie otwartych deskryptorów plików

- funkcja `close`,
- stała `NOFILE` lub `_NFILE` w System V,
- funkcja `getdtablesize` w BSD,

Zmiana roboczego katalogu bieżącego

- funkcja `chdir`

Ustawianie maski trybu dostępu do plików

- funkcja `umask`

Przejdźcie do pracy w tle

- przejęcie działania w charakterze potomka
- funkcja `fork` i `exit`

Budowanie procesu demona cd

Odłączenie od grupy procesów

- funkcja `setpgrp`

Ignorowanie sygnałów terminala sterującego (BSD)

- sygnały `SIGTTOU`, `SIGTTIN`, `SIGTSTP`
- funkcja `signal`

Odłączenie od terminala sterującego w System V

- funkcja `setpgrp` w System V

Odłączenie od terminala sterującego w BSD

- funkcja `open` („/dev/tty”, ...)
- funkcja `ioctl` (... , `TIOCNOTTY`, ...)

Budowanie procesu demona cd

Blokada odzyskiwania terminala sterującego

- ❑ w BSD nie jest możliwe odzyskanie terminala, bo identyfikator grupy procesów demona jest różny od zera
- ❑ w System V należy proces pozbawić przywództwa w grupie powołując kolejnego potomka
 - sygnał *SIGHUP* i funkcja *signal*
 - funkcja *fork* i *exit*

Obsługa sygnału *SIGCLD*

- ❑ w System V – ignorowanie
- ❑ w BSD – funkcja *wait3* i opcja *WNOHANG*

Przykłady

plik – [Sygnały-przykłady](#)