Sygnały

Sygnał jest przerwaniem programowym procesu, generowanym przez jądro systemu operacyjnego lub przez inny proces. Sygnał może pojawić się w dowolnym momencie w czasie pracy procesu (pojawia się asynchronicznie). Otrzymanie sygnału przez proces oznacza, że pojawiło się jakieś zdarzenie wyjątkowe, wymagające natychmiastowej reakcji ze strony procesu.

Reakcja procesu na otrzymany sygnał

- Wykonanie akcji domyślnej -(najczęściej zakończenie procesu z ewentualnym zrzutem zawartości segmentów pamięci na dysk, czasami zignorowanie sygnału),
- Zignorowanie sygnału
- Przechwycenie sygnału tj. podjęcie akcji zdefiniowanej przez użytkownika

W zależności od zdarzenia w systemie wysłane mogą być różne sygnały (jest ich 19, ☞ Tabela 1).

Tabela 1 Sygnały w systemie Unix

Nazwa	Nr	Opis	Akcja standardowa
SIGHUP	1	zerwanie łączności z terminalem	zakończenie procesu
SIGINT	2	przerwanie programu	zakończenie procesu
SIGQUIT	3	zakończenie programu	zakończ. proc. i utw. obrazu pam.
SIGILL	4	niedozwolona instrukcja	zakończ. proc. i utw. obrazu pam.
SIGTRAP	5	Śledzenie	zakończ. proc. i utw. obrazu pam.
SIGIOT	6	instrukcja IOT	zakończ. proc. i utw. obrazu pam.
SIGEMT	7	instrukcja emulatora EMT	zakończ. proc. i utw. obrazu pam.
SIGFPE	8	wyjątek zmiennopozycyjny	zakończ. proc. i utw. obrazu pam.
SIGKILL	9	zabicie procesu (unicestwienie)	zakończenie procesu
SIGBUS	10	błąd magistrali	zakończ. proc. i utw. obrazu pam.
SIGSEGV	11	naruszenie segmentacji	zakończ. proc. i utw. obrazu pam.
SIGSYS	12	nieprawidłowy arg. funkcji systemowej	zakończ. proc. i utw. obrazu pam.
SIGPIPE	13	pisanie do łącza nie otwartego do czytania	zakończenie procesu
SIGALRM	14	pobudka	zakończenie procesu
SIGTERM	15	zakończenie programowe	zakończenie procesu
SIGUSR1	16	sygnał definiowany przez użytkownika	zakończenie procesu
SIGUSR2	17	sygnał definiowany przez użytkownika	zakończenie procesu
SIGCLD	18	zakończenie procesu potomnego	odrzuć sygnał
SIGPWR	19	niedobór mocy	zakończenie procesu

Funkcje obsługujące sygnały zawarte są w bibliotece **<signal.h>**

Funkcja KILL

int kill(int pid, int signum) - Wysyłanie sygnału

pid – identyfikator procesu, do którego chcemy wysłać sygnał
signum – numer wysyłanego sygnału

Wynik:
-1 w przypadku błędu (szczegóły w **errno**)
0 OK.

- pid > 0 oznacza proces o identyfikatorze pid,
- pid = 0 oznacza grupę procesów do których należy proces wysyłający,
- pid = -1 oznacza wszystkie procesy, których rzeczywisty identyfikator użytkownika jest równy efektywnemu (obowiązującemu) identyfikatorowi procesu wysyłającego sygnał,
- pid < -1 oznacza procesy należące do grypy o identyfikatorze -pid.

Funkcja SIGNAL

void (*signal(int signum, void (*f)()))() – Określenie sposobu obsługi sygnału

signum – numer sygnału, którego obsługa ma zostać zmieniona
f – może obejmować jedną z trzech wartości:

- **SIG_DFL** -(wartość 0) standardowa reakcja na sygnał
- **SIG_IGN** -(wartość 1) ignorowanie sygnału
- **Wskaźnik do funkcji** - wskaźnik na funkcję, która będzie uruchomiona w reakcji na sygnał

Wynik:

-1 (SIG_ERR)	w przypadku błędu (szczegóły w **errno**)
Wskaźnik na poprzednią procedurą obsługi, lub odpowiednio SIG_IGN, SIG_DFL	identyfikator grupy procesów

1. Nie można przechwytywać, ani ignorować sygnałów SIGKILL i SIGSTOP.
2. Gdy w procesie macierzystym ustawiony jest tryb ignorowania sygnału SIGCLD to po wywołaniu funkcji. `exit` przez proces potomny, proces zombi nie jest zachowywany i miejsce w tablicy procesów jest natychmiast zwalniane

Przykłady użycia funkcji signal():

```
void (*f)();

f=signal(SIGINT,SIG_IGN); /* ignorowanie sygnału sigint*/

signal(SIGINT,f); /*przywrócenie poprzedniej reakcji na syg.*/

signal(SIGINT,SIG_DFL); /*ustaw. standardowej reakcji na syg.*/
```

```
void moja_funkcja() {
   printf("Został przechwycony sygnał\n");
   exit(0);
}

main(){
   signal(SIGINT,moja_funkcja); /* przechwycenie sygnału */
   ⋮
}
```

Funkcja PAUSE

void pause() -oczekiwanie na sygnał

Zawiesza wywołujący proces aż do chwili otrzymania dowolnego sygnału. Jeśli sygnał jest ignorowany przez proces, to funkcja `pause` też go ignoruje. Najczęściej sygnałem, którego oczekuje `pause` jest sygnał pobudki SIGALARM.

Funkcja ALARM

unsigned alarm (unsigned int sek) - Funkcja wysyła sygnał SIGALARM po upływie czasu podanym przez użytkownika.

sek - ilość sekund po których wysyłany jest sygnał SIGALARM

Wynik:

0	w przypadku gdy alarm nie był wcześniej ustawiony
Ilość sekund pozostałą do wysłania sygnału	W przypadku wcześniejszego wywołania funkcji `alarm`, w wyniku której nie został jeszcze wysłany sygnał SIGALARM