

Systemy Operacyjne 2017/18

[Strona główna](#) [Moje kursy](#) [SO2018](#) [Laboratorium 4](#) [Zadania - Zestaw 4](#)

Zadania - Zestaw 4

Otwarto: niedziela, 1 marca 2015, 01:00

Zestaw 4. Sygnały

Rodzaje sygnałów: SIGINT, SIGQUIT, SIGKILL, SIGTSTP, SIGSTOP, SIGTERM, SIGSEGV, SIGHUP, SIGALARM, SIGCHLD, SIGUSR1, SIGUSR2

Sygnały czasu rzeczywistego: SIGRTMIN, SIGRTMIN+n, SIGRTMAX

Przydatne polecenia Unix: kill, ps

Przydatne funkcje systemowe: kill, raise, sigqueue, signal, sigaction, sigemptyset, sigfillset, sigaddset, sigdelset, sigismember, sigprocmask, sigpending, pause, sigsuspend

Zadanie 1 (25%)

- Napisz program wypisujący w pętli nieskończonej aktualną godzinę. Po odebraniu sygnału SIGTSTP (CTRL+Z) program zatrzymuje się, wypisując komunikat "Oczekuję na CTRL+Z - kontynuacja albo CTR+C - zakończenie programu". Po ponownym wysłaniu SIGTSTP program powraca do pierwotnego wypisywania.
Program powinien również obsługiwać sygnał SIGINT. Po jego odebraniu program wypisuje komunikat "Odebrano sygnał SIGINT" i kończy działanie. W kodzie programu, do przechwycenia sygnałów użyj zarówno funkcji signal, jak i sigaction (np. SIGINT odbierz za pomocą signal, a SIGTSTP za pomocą sigaction).
- Zrealizuj powyższe zadanie, tworząc program potomny, który będzie wywoływał jedną z funkcji z rodziny exec skrypt shellowy zawierający zapętlone systemowe polecenie date. Proces macierzysty będzie przychwytywał powyższe sygnały i przekazywał je do procesu potomnego, tj. po otrzymaniu SIGTSTP kończy proces potomka, a jeśli ten został wcześniej zakończony, tworzy nowy, wznowiając działanie skryptu, a po otrzymaniu SIGINT kończy działanie potomka (jeśli ten jeszcze pracuje) oraz programu.

Zadanie 2 (35%)

Napisz program, który tworzy N potomków i oczekuje na ich prośby na rozpoczęcie działania (sygnały SIGUSR1). Po uzyskaniu K prośb, proces powinien pozwolić kontynuować działanie wszystkim procesom, które poprosiły o przejście (wysłać im sygnał pozwolenia na rozpoczęcie pracy) i niezwłocznie akceptować każdą kolejną prośbę. Program powinien wypisać listę wszystkich otrzymanych sygnałów wraz numerem PID potomka oraz kodem zakończenia procesu (w przypadku otrzymania sygnału zakończenia pracy potomka).

Program kończy działanie po zakończeniu pracy ostatniego potomka albo po otrzymaniu sygnału SIGINT (w tym wypadku należy zakończyć wszystkie działające procesy potomne).

N i M są argumentami programu.

Zachowanie dzieci powinno być następujące: Każde dziecko najpierw symuluje pracę (usypia na 0-10 sekund). Następnie wysyła sygnał SIGUSR1 prośby o przejście, a po uzyskaniu pozwolenia losuje jeden z 32 sygnałów czasu rzeczywistego (SIGRTMIN, SIGRTMAX), wysyła go do rodzica i kończy działanie, zwracając liczbę sekund jaką proces wylosował do uśpienia.

Program główny powinien mieć możliwość śledzenia informacji na temat: (dla czytelności w łatwy sposób powinno dać się je grupami włączyć i wyłączyć)

- tworzenia procesu potomnego (jego nr PID),
- otrzymanych prośb od procesów potomnych,
- wysłanych pozwoleń na wysłanie sygnału rzeczywistego
- otrzymanych sygnałów czasu rzeczywistego (wraz z numerem sygnału)
- zakończenia procesu potomnego (wraz ze zwróconą wartością)

Zadanie 3 (40%)

Napisz program który tworzy proces potomny i wysyła do niego L sygnałów SIGUSR1, a następnie sygnał zakończenia wysyłania SIGUSR2. Potomek po otrzymaniu sygnałów SIGUSR1 od rodzica zaczyna je odsyłać do procesu macierzystego, a po otrzymaniu SIGUSR2 kończy pracę.

Proces macierzysty w zależności od argumentu Type (1,2,3) programu wysyła sygnały na trzy różne sposoby:

- SIGUSR1, SIGUSR2 za pomocą funkcji kill (15%)
- SIGUSR1, SIGUSR2 za pomocą funkcji kill, z tym, że proces macierzysty wysyła kolejny sygnał dopiero po otrzymaniu potwierdzenia odebrania poprzedniego (15%)
- wybrane 2 sygnały czasu rzeczywistego za pomocą kill (10%)

Program powinien wypisywać informacje o:

- liczbie wysłanych sygnałów do potomka
- liczbie odebranych sygnałów przez potomka
- liczbie odebranych sygnałów od potomka

Program kończy działanie po zakończeniu pracy potomka albo po otrzymaniu sygnału SIGINT (w tym wypadku od razu wysyła do potomka sygnał SIGUSR2, aby ten zakończył pracę. Wszystkie pozostałe sygnały są blokowane w procesie potomnym).

L i Type są argumentami programu.