

Laboratorium 6. Sygnały

Pobierz plik lab6.tar.gz i rozpakuj go.

Zadanie 1. Obsługa sygnałów

- a) Zapoznaj się z programem `sigcatch.c`. Jakie jest zadanie funkcji `signal()` oraz `pause()`? Jak program zachowa się w przypadku otrzymania sygnału `INT` a jak w przypadku innego sygnału, np. `QUIT`, `TERM`?
- b) Zapoznaj się z programem `raise.c`. Jakie jest zadanie funkcji `signal()`, `pause()`, `kill()`?

Zadanie 2. Podstawowa obsługa sygnałów

Napisz program, który co n sekund wypisuje na ekranie komunikat: "Działam dalej". Naciśnięcie klawisza: `ctrl+c` (sygnał `INT`) wypisuje napis "Przechwycilem sygnał" i program nadal działa. Jakiej funkcji użyjesz do odmierzenia czasu?

Zadanie 3. Podstawowa obsługa sygnałów

Napisz program, który prosi o podanie hasła i kończy działanie, jeśli użytkownik nie wprowadzi hasła w określonym czasie. Wskazówka: użyj funkcji `alarm()`.

Zadanie 4. Sygnały niezawodne (POSIX)

Zapoznaj się z programem `sygnaly.c` z pliku `lab6.tar.gz`. Napisz wersję tego programu, która korzysta z funkcji `sigaction()`. Sprawdź obsługę sygnałów w przypadku użycia każdej z funkcji. Czy funkcja `sigaction()` z wartościami domyślnymi jest całkowicie równoważna funkcji `signal()`? Sprawdź:

- Czy przywracana jest obsługa domyślna sygnału?
- Co się dzieje, jeśli podczas obsługi sygnału `X` pojawi się nowy, taki sam sygnał `X`? Czy następny sygnał jest blokowany? Czy obsługa sygnału `X` jest przerywana, aby obsłużyć nowy sygnał `X`? Co się dzieje, jeśli pojawi się kilka takich samych sygnałów `X`? Czy ustawiane są w kolejce?
- Co się dzieje, jeśli podczas obsługi sygnału `X` pojawi się nowy sygnał `Y`? Czy obsługa sygnału `X` jest przerywana i przechodzisz do obsługi sygnału `Y`? Czy też jest kontynuowana a dopiero po obsłużeniu sygnału `X` obsługiwany jest sygnał `Y`?
- Co się dzieje z funkcją `read()`, gdy pojawi się sygnał? Czy zostaje ona przerwana? Sprawdź: wpisz fragment tekstu, nie naciskaj `Enter`, w zamian prześlij sygnał do procesu. Czy wczytany został cały tekst, czy tylko jego ostatni fragment? Czy po obsłudze sygnału program automatycznie ponownie wywołał funkcję `read()`, czy też wrócił z funkcji `read()` z błędem `EINTR`?

Zadanie 5. Obsługa sygnałów

Napisz własną wersję polecenia `kill`. Polecenie to powinno być wywoływane następująco:

```
kill [ -s nazwa-sygnału lub numer ] pid ...
```

Jeśli nie zostanie określony sygnał to wysyłany jest sygnał `SIGTERM`. Uwzględnić obsługę błędów (np. brak uprawnień, brak sygnału o podanym numerze/nazwie, brak procesu, do którego wysyłany jest sygnał).

Zadanie domowe

Zadanie 6.

Opisz wyniki uzyskane w zadaniu 4.

Zadanie 7.

Co to jest proces demon? Napisz własną procedurę tworzenia takiego procesu. Porównaj jej działanie z funkcją systemową `daemon()`.

Zadanie 8.

Napisz program działający jako demon i nadzorujący pracę wybranych użytkowników (liczbę procesów, liczbę otwartych plików, ich wielkość, inne wybrane zasoby). W razie przekroczenia limitów program wysyła pocztą komunikat do użytkownika i do administratora. Program działa w oparciu o plik konfiguracyjny.