

Tworzenie procesów. Środowisko procesu, sterowanie procesami

Zadanie 1

Napisz program, który przyjmuje jeden argument: *argv[1]*. Program ma utworzyć *argv[1]* procesów potomnych. Każdy proces potomny ma wypisać na standardowym wyjściu w jednym wierszu dwa identyfikatory: identyfikator procesu macierzystego i swój własny. Na końcu standardowego wyjścia proces macierzysty ma wypisać *argv[1]*. **Wskazówka:** aby program na pewno wypisywał *argv[1]* jako ostatni wiersz standardowego wyjścia, należy użyć funkcji systemowej `wait()`.

Zadanie 2

Napisz program, który przyjmuje jeden argument: *argv[1]* — ścieżkę katalogu. Program powinien wypisać na standardowym wyjściu swoją nazwę, korzystając z funkcji `printf()`. Zadeklaruj zmienną globalną *global*, a następnie zmienną lokalną *local*. W zależności od zwróconej wartości przez `fork()` dokonaj obsługi błędu, wykonaj proces rodzica / proces potomny. W procesie potomnym:

1. wyświetl komunikat „child process”,
2. dokonaj inkrementacji zmiennych *global* i *local*,
3. wyświetl komunikat „child pid = %d, parent pid = %d”
4. wyświetl komunikat „child's local = %d, child's global = %d”
5. wykonaj program `/bin/ls` z argumentem *argv[1]*, korzystając z funkcji `execl()`, zwracając przy tym jej kod błędu.

W procesie rodzica:

1. wyświetl komunikat „parent process”
2. wyświetl komunikat „parent pid = %d, child pid = %d”
3. wyświetl komunikat „Child exit code: %d”
4. wyświetl komunikat „Parent's local = %d, parent's global = %d”
5. zwróć stosowny kod błędu.