

Temat zajęć	Procesy w systemie Linux
Zakres materiału	Tworzenie i wykonywanie procesów w systemie Linux

Materiał teoretyczny

- funkcje *fork()*, *waitpid()*, *execv()*, argumenty linii poleceń i kod zakończenia procesu

Treść zadania

Napisz w języku C program spełniający poniższe wymagania:

- program akceptuje wywołanie z co najmniej jednym argumentem – uruchomienie go bez argumentów ma spowodować wyprowadzenie na *stderr* stosownej informacji i zakończenie pracy z kodem powrotu *201*;
- wszystkie podane programowi argumenty muszą być liczbami całkowitymi z przedziału $[0,100]$ – uruchomienie go w inny sposób ma spowodować wyprowadzenie na *stderr* informacji o błędzie i zakończenie pracy z kodem powrotu *202*;
- jeżeli program uruchomiono z jednym argumentem, program kończy pracę z kodem powrotu równym wartości otrzymanego argumentu;
- jeżeli program uruchomiono z dwoma argumentami, to program kończy pracę z kodem powrotu równym wartości większego z otrzymanych argumentów;
- jeżeli program uruchomiono z trzema lub więcej argumentami, program dzieli otrzymane argumenty na dwie równoliczne listy, a jeśli liczba argumentów jest nieparzysta, to na listy, z których druga jest o jeden element dłuższa od pierwszej;
- program uruchamia dwa procesy potomne (funkcja *fork()*);
- każde z dzieci zastępuje swój kod (funkcja *execv()*), uruchamiając ten sam program, który jest wykonywany przez rodzica i przekazuje mu jako argument odpowiednio: do pierwszego dziecka – lewą listę otrzymanych argumentów, do drugiego dziecka – prawą listę otrzymanych argumentów;
- rodzic czeka na zakończenie obu procesów potomnych (funkcja *waitpid()*), a następnie wypisuje na *stdout* trzy linie tekstu:
 1. swój PID, PID pierwszego zakończonego dziecka, kod powrotu pierwszego dziecka, listę argumentów, z którymi zostało uruchomione pierwsze dziecko;
 2. swój PID, PID drugiego zakończonego dziecka, kod powrotu drugiego dziecka, listę argumentów, z którymi zostało uruchomione drugie dziecko;
 3. swój PID i większy z kodów powrotów obu dzieci (wartość tę należy wyprowadzić w tej samej kolumnie, co kody powrotu potomków);
- rodzic kończy pracę, zwracając kod powrotu równy większemu z kodów powrotu obu dzieci;

- w efekcie w ostatniej linii wyjścia powinna znaleźć się wartość największego argumentu spośród wszystkich przekazanych do programu przy jego uruchomieniu.

Uwaga: poprawna sekwencja operacji to *fork-fork-waitpid-waitpid* (czyli rodzic **najpierw** uruchamia **dwa** procesy potomne, a potem czeka na zakończenie obu); konstrukcja kodu, w której operacje układają się w ciąg *fork-waitpid-fork-waitpid* (procesy potomne pracują po kolei) dyskwalifikuje rozwiązanie!

Przykładowe wyjście z programu:

```
$. /z08 10 5 32 76 12 99

19998 20002 10 | 10
19998 20003 32 | 5 32
19998          32

19999 20000 76 | 76
19999 20001 99 | 12 99
19999          99

19997 19998 32 | 10 5 32
19997 19999 99 | 76 12 99
19997          99
```

Uwaga! Kod źródłowy programu (1 plik) po zaprezentowaniu prowadzącemu zajęcia laboratoryjne musi zostać jako **załącznik** przesłany na adres `sos1@wi.zut.edu.pl`:

- plik z kodem źródłowym musi mieć nazwę: `numer_indeksu.so.lab08.c` (np. `66666.so.lab08.c`),
- plik musi zostać wysłany z poczty uczelnianej (domena `zut.edu.pl`),
- temat maila musi mieć postać: `SO IS1 999X LAB08` gdzie `999X` to numer grupy laboratoryjnej (np. `SO IS1 210B LAB08`),
- w pierwszych trzech liniach kodu źródłowego w komentarzach (każda linia komentowana osobno) musi znaleźć się:
 - informacja identyczna z zamieszczoną w temacie maila,
 - imię i nazwisko osoby wysyłającej maila,
 - adres e-mail, z którego wysłano wiadomośćnp.:

```
// SO IS1 210B LAB08
// Jan Nowak
// nj66666@zut.edu.pl
```

- e-mail nie może zawierać żadnej treści (tylko załącznik).

Dostarczone kody programów będą analizowane pod kątem wykrywania plagiatów. Niewysłanie wiadomości, wysłanie jej w formie niezgodnej z powyższymi wymaganiami lub wysłanie pliku, który nie będzie się kompilował i uruchamiał, będzie traktowane jako brak zadania i skutkowało otrzymaniem oceny niedostatecznej.