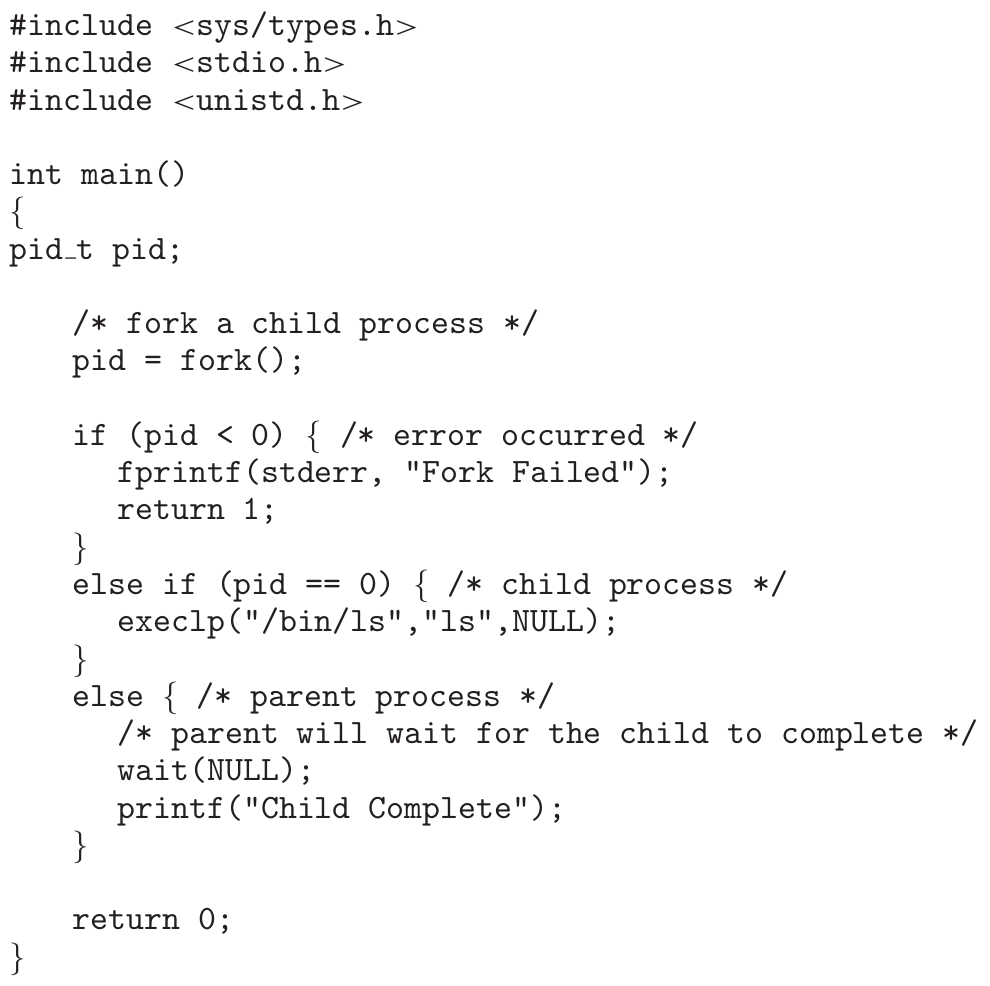
**Przedmiot:** Systemy operacyjne - laboratorium

**Tematyka:** Procesy

**Informacje wstępne:**

* Proszę zapoznać się z działanie fukcji *fork()* języka C (np.: man fork).
* Proszę zapoznać się z działaniem programu (przekleić, skompilować, uruchomić):



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

**Zadanie 1.**

Podany program proszę rozbudować w następujący sposób:

* Po wykonaniu funkcji *fork()* odczytać i wyświetlić PID: zarówno procesu macierzystego, jak i potomnego (obie te informacje wyświetlić zarówno z poziomu procesu macierzystego, jak i potomnego).
* Zawiesić działanie programu macierzystego na funkcji odczytującej z klawiatury.
* Dla procesu potomnego zamienić funkcję *execlp()* na funkcję odczytania z klawiatury zmiennej całkowitej.
* Przed wykonaniem *fork()* zarezerwować zasoby:
  + Otworzyć do pisania plik
  + Zarezerwować miejsce w pamięci (statycznie i dynamicznie)

Na tej podstawie proszę przedstawić:

* Odszukać w *pstree* procesy (macierzysty i potomny) i przedstawić je
* Zinterpretować dostęp do zasobów w procesie macierzystym i potomnym
* Który z procesów odczyta wprowadzone dane z klawiatury (dla czego?) ?
* Co się wydarzy z procesem potomnym po usunięciu procesu macierzystego ?
  + W dwojakim postępowaniu:
  + 1/ proces macierzysty zakończył działanie
  + 2/ proces macierzysty otrzymał kill -9

**Zadanie 2.**

Napisz program przechwytujący przesłane sygnały z programu *kill* i wyświetli ich otrzymanie.

Podpowiedź: funkcja *signal()*

Czy wszystkie sygnały można przechwycić ?

**Zadanie na podsumowanie**

Proszę sprawdzić, ile procesów potomnych uda się utworzyć, zanim system ulegnie zawieszeniu ? Proszę tą informacją podzielić się na ‘forum technicznym’.

Proszę tam też wkleić kod, którym następowało sprawdzenie.