

## Procesy – zadania:

1. Napisz program tworzący dwa procesy. Każdy ze stworzonych procesów powinien utworzyć proces – potomka. Należy wyświetlać identyfikatory procesów rodziców i potomków po każdym wywołaniu funkcji `fork`.
2. Napisz program wypisujący napis „*Początek*”, następnie wywołujący funkcję `fork` i wypisujący napis „*Koniec*”. Co jest efektem uruchomienia tego programu?
3. Napisz program wypisujący napis „*Początek*”, następnie wywołujący funkcję `exec` powodującą wydruk zawartości bieżącego katalogu i następnie wypisujący napis „*Koniec*”. Co jest efektem uruchomienia tego programu?
4. Napisz program którego rezultatem będzie wydruk zawartości bieżącego katalogu poprzedzony napisem „*Początek*” a zakończony napisem „*Koniec*”
5. Napisz program, którego wynikiem jest sformatowana lista procesów:

-----początek listy-----

proces 1

proces2

.....

-----koniec listy-----

6. Napisz program tworzący równocześnie trzy procesy zombi.

## Procesy – zadania 2:

7. Zaimplementuj program wykonujący polecenie `ps ax`, i zapisujący jego wynik w pliku o nazwie podanej jako parametr wywołania programu.
8. Napisz program tworzący proces sierotę.
9. Stwórz grupę procesów (3 procesy – jeden z nich jest liderem). Niech lider czeka na zakończenie dwóch procesów z jego grupy. Mogą się przydać funkcje: `setgrp()` oraz `setpgid()`