

Systemy Operacyjne

Zestaw I

Grupa 4.

Proszę napisać odpowiedni plik `Makefile` oraz `README` dotyczący wszystkich programów/zadań w tym zestawie. Zadania umieścić w katalogu `Zestaw1`. Nazwy plików wykonywalnych powinny posiadać rozszerzenie `.x`.

1. kopiuj

Proszę napisać program, który kopiuje zawartość jednego pliku do drugiego.

Użyć funkcji `open`, `read`, `write`. Obsłużyć możliwe błędy, użyć funkcji `perror`. Program przyjmuje nazwy plików z linii komend. W przypadku złej liczby argumentów wypisać komunikat o użyciu programu (użyć aktualnej nazwy programu).

2. procinfo

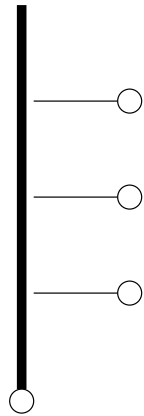
- Proszę napisać funkcję, która wypisuje identyfikatory `UID`, `GID`, `PID`, `PPID` i `PGID` dla aktualnego procesu oraz nazwę programu przekazaną jako argument.
- Implementację funkcji umieścić w osobnym pliku `".c"` oraz stworzyć odpowiedni plik nagłówkowy `".h"` z deklaracjami.

3. procesy

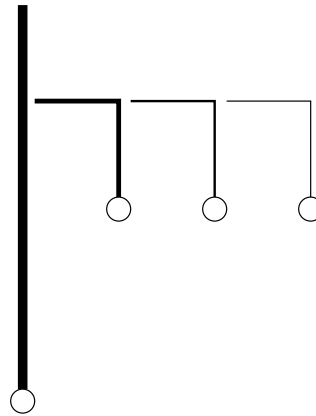
- Napisać program wypisujący identyfikatory `UID`, `GID`, `PID`, `PPID` i `PGID` dla danego procesu - użyć funkcji z poprzedniego zadania.
- Wywołać funkcję `fork` 3 razy (np. w pętli) i wypisać powyższe identyfikatory dla wszystkich procesów potomnych.
- Wstawić funkcję `sleep` tak, aby procesy pojawiały się na ekranie grupowane pokoleniami od najstarszego do najmłodszego.
- Na podstawie wyników programów narysować „drzewo genealogiczne” tworzonych procesów (z zaznaczonymi identyfikatorami).
- *Ile powstaje procesów i dlaczego?*
- Zmienić nazwę programu `argv[0]` na `"parent"` dla procesu macierzystego oraz `"child"` dla procesu potomnego.
Wynik sprawdzić używając komend `ps u` i `ps -a`. Proszę pamiętać o nie przekroczeniu długości argumentu `argv[0]`. Drzewko wszystkich procesów można zobaczyć komendą `ps -AHF`.

4. **drzewko1, drzewko2** Proszę napisać program jak powyższy ale generujący następujące „drzewka genealogiczne” procesów:

1)



2)



Andrzej Görlich
goerlich@ibis.if.uj.edu.pl
<http://th.if.uj.edu.pl/~atg/SO>