Przetwarzanie współbieżne

Laboratorium 2

Jan Bartula

# Implementacja funkcji

Funkcja **program.c**

int main(){  
 printf("Jan Bartula %d\n", getpid());  
}

## Uruchomienie funkcji

### Uruchomienie clone:

Jan Bartula 8629

Jan Bartula 8630

Jan Bartula 8631

Jan Bartula 8632

Jan Bartula 8633

Jan Bartula 8634

Jan Bartula 8635

Jan Bartula 8636

Jan Bartula 8637

Jan Bartula 8638

czas standardowy = 0.000430

czas CPU = 0.000430

czas zegarowy = 0.018668

Zmienna Globalna: 10

### Uruchomienie fork:

Jan Bartula 8643

Jan Bartula 8644

Jan Bartula 8645

Jan Bartula 8646

Jan Bartula 8647

Jan Bartula 8648

Jan Bartula 8649

Jan Bartula 8650

Jan Bartula 8651

Jan Bartula 8652

Zmienna Globalna: 0

czas standardowy = 0.000628

czas CPU = 0.000626

czas zegarowy = 0.023260

### Wniosek

Jak możemy zauważyć przestrzeń adresowa w fork jest inna niż w procesie macierzystym. W clone, zmienna globalna ma wartość równą ilości uruchomionych clonów. Co oznacza, że przestrzeń adresowa jest wspólna.

## Zmiana funkcji clone:

Aby funkcja clone działała jak fork, należy usunąć flagi podczas tworzenia kopii procesu:

# define CLONE\_SIGHAND 0x00000800 /\* Set if signal handlers shared. \*/

# define CLONE\_VM 0x00000100 /\* Set if VM shared between processes. \*/

Po usunięciu obu flag, kompilacji i ponownym uruchomieniu:

Jan Bartula 8670

Jan Bartula 8671

Jan Bartula 8672

Jan Bartula 8673

Jan Bartula 8674

Jan Bartula 8675

Jan Bartula 8676

Jan Bartula 8677

Jan Bartula 8678

Jan Bartula 8679

czas standardowy = 0.000603

czas CPU = 0.000603

czas zegarowy = 0.023371

Zmienna Globalna: 0

Oznacza to, że flagi działają zgodnie z definicją w dokumentacji. Dzielenie się pamięcią wirtualną pomiędzy procesami klonowanymi, a także sygnał obsługi współdzielonej. Para tych dwóch flag powoduje dzielenie się pamięcią, a zatem wcześniej mogły inkrementować tą samą zmienną, w obecnej sytuacji, znajduje się to w innym miejscu fizycznej pamięci.

Uwaga: należy pamiętać, że pointery „teoretycznie” wskazują na tę samą adres pamięci w procesie macierzystym i pochodnym. Ale jest to adres pamięci wirtualnej dla każdego z procesów.   
Ponieważ zostały one sklonowane, zmienne otrzymują one ten sam adres wirtualny, ale nie oznacza to, że jest to w tym samym adresie fizycznym RAM.