1.Wątki i mutexy

by Paweł Paduch — Ostatnio zmodyfikowane 2014-11-13 11:39

1. Napisać (lub przerobić jeden z poprzednich) program w którym proces macierzysty tworzy 100 lub więcej procesów, które nic nie wykonują i się kończą. Proces macierzysty czeka na zakończenie wszystkich swoich procesów, poczym zwraca zmierzony czas całej operacji (np. za pomocą clock_gettime (patrz. man -s3 clock_gettime)). Napisać analogiczny program bazujący na wątkach. Porównać wyniki. Czy wątki są szybsze?
2. Napisać program w którym:

- Stworzymy sobie 5 wątków w odstępach 1 sekundowych. Każdy wątek powinien mieć 10s „sleepa".
- Za pomocą jednej struktury ze zmiennymi typu char* oraz int przekazać (przez parametr) tworzącym się wątkom ich unikalne nazwy i numery np.: „wątek1";0, „wątek2";1, „wątek3";3 itd. Zmiana tych danych ma następować po „sleepach" w procesie macierzystym. Wątki przed swoim zakończeniem powinny wypisać dane przekazane im przez parametr, wykorzystać dane specyficzne wątku. Porównać działanie z danymi specyficznymi wątku i bez.
- Pierwszy uruchomiony wątek powinien stworzyć klucz danych specyficznych wątku (wykorzystać pthread_once), natomiast, każdy z wątków, ustawić daną specyficzną wątku dla jego nazwy, zobaczyć jak działa.
- W procesie macierzystym zrezygnować z 1s. „sleepów" tak by wątki startowały wszystkie bez czekania. Czy coś się zmieniło?
- Wykorzystać mutexy do zsynchronizowania zakładania danych specyficznych wątku przed ich zmianą przez proces macierzysty.

3. Napisać program tworzący dwa wątki.

- W pierwszym wątku po 2s wywołać pthread_cancel z tidem wątku drugiego ponowić próbę po kolejnych 2s.
- W drugim wątku na samym początku za pomocą pthread_set_cancelstate zabezpieczyć się przed jego usunięciem, poczym po 3s pozwolić na usunięcie.
- Do obu wątków dodać procedurę porządkowania wyświetlającą jakiś komunikat.
- Przetestować funkcję usuwającą procedurę porządkującą z parametrem 1 i 0.

4. Oczywiście nie zapominamy o sprawozdaniu

2.