

Zadanie poprawkowe: Sortowanie współbieżne w pamięci dzielonej

Termin wysyłania rozwiązań upływa 10 lutego 2018 roku o godz. 22:59 CET.

Pytania do zadania proszę kierować na adres `p.zuk@mimuw.edu.pl`

Grupa procesów składająca się z procesu nadzorczego `Sort` oraz procesów $A(0), A(1), \dots, A(N-1)$ i $B(0), B(1), \dots, B(N-2)$ realizuje sortowanie $2N$ ($N \geq 2$) liczb całkowitych.

Proces `Sort` wczytuje ze standardowego wejścia liczbę N , alokuje pamięć dzieloną, następnie wczytuje do niej ze standardowego wejścia tablicę $t[0..2N-1]$ liczb całkowitych. W kolejnym kroku `Sort` uruchamia procesy $A(i)$ oraz $B(i)$.

Każdy z procesów $A(i)$ cyklicznie:

- porównuje i w razie konieczności zamienia liczbę $t[2*i]$ oraz $t[2*i+1]$
- sygnalizuje procesom $B(i)$ oraz $B(i-1)$ ($A(0)$ sygnalizuje tylko procesowi $B(0)$, a $A(N-1)$ tylko $B(N-2)$) zakończenie kroku sortowania
- przed rozpoczęciem kolejnej iteracji, oczekuje na zakończenie kroku sortowania przez w/w procesy.

Każdy z procesów $B(i)$ cyklicznie:

- oczekuje na zakończenie kroku sortowania przez procesy $A(i)$ oraz $A(i+1)$
- porównuje i w razie konieczności zamienia liczbę $t[2*i+1]$ oraz $t[2*i+2]$
- sygnalizuje procesom $A(i)$ oraz $A(i+1)$ zakończenie kroku sortowania.

Po posortowaniu procesy $A(i)$, $B(i)$ kończą pracę, a proces `Sort` wypisuje posortowane liczby na standardowe wyjście (po jednej w wierszu).

Zaimplementuj w C przedstawiony schemat komunikacji. Do komunikacji między procesami należy wykorzystać pamięć dzieloną oraz semaforey.

Rozwiązanie należy przesłać pocztą elektroniczną jako załącznik na adres `p.zuk@mimuw.edu.pl`. Załączony plik powinien mieć nazwę `AB123456.tar.gz`, gdzie `AB123456` to login na maszynie `students`. Rozwiązanie musi zawierać plik `Makefile` albo `CMakeLists.txt`. Należy zadbać by rozwiązanie działało na komputerach w laboratorium.

Kryteria oceniania

Rozwiązania, w których procesy $A(i)$ oraz $B(i)$ wykonują ustaloną (zależną od N) liczbę kroków sortowania mogą otrzymać maksymalnie 8 pkt. Aby otrzymać 10 pkt. należy zaimplementować wykrywanie sytuacji, w której tablica staje się posortowana.

Przykład

Wejście:

2
6
5
3
7

Wyjście:

3
5
6
7