

# IPC - pamięć wspólna, semaforey

## Przydatne funkcje:

System V:

`<sys/shm.h>` `<sys/ipc.h>` - `shmget`, `shmctl`, `shmat`, `shmdt`

POSIX:

`<sys/mman.h>` - `shm_open`, `shm_close`, `shm_unlink`, `mmap`, `munmap`

## Zadanie

Wykorzystując semaforey i pamięć wspólną z IPC Systemu V napisz program symulujący działanie sklepu wysyłkowego.

W sklepie wysyłkowym pracują 3 typy pracowników:

- 1) odbierający zamówienie oraz przygotowujący odpowiednie paczki
- 2) pakujący zamówienie do paczki
- 3) zaklejający paczki oraz wysyłający je kurierem

Po uruchomieniu, pracownik 1) cyklicznie odbiera zamównienia - losuje dodatnią liczbę całkowitą (wielkość zamówienia/paczki) i umieszcza ją w tablicy przechowywanej w pamięci wspólnej. Po odebraniu nowego zamównienia pracownik 1) wypisuje na ekranie komunikat postaci:

(pid timestamp) Dodałem liczbę : n. Liczba zamównień do przygotowania: m. Liczba zamównień do wysłania: x.

gdzie pid to PID procesu pracownika 1), timestamp to aktualny czas (z dokładnością do milisekund), n to wylosowana liczba (wielkość zamówienia), m to liczba zamównień do przygotowania w pamięci wspólnej (licząc z utworzonym zamówieniem), a x to liczba zamównień do wysłania w pamięci wspólnej.

Po pobraniu zamówienia pracownik 2) pomnaża liczbę przez 2 i wypisuje na ekranie komunikat:

(pid timestamp) Przygotowałem zamówienie o wielkości n. Liczba zamównień do przygotowania: m. Liczba zamównień do wysłania: x.

gdzie n to liczba pomnożona przez 2, a m jest bez aktualnie przygotowanego.

Po pobraniu przygotowanego zamówienia pracownik 3) pomnaża liczbę przez 3 i wypisuje na ekranie komunikat:

(pid timestamp) Wysłałem zamówienie o wielkości n. Liczba zamównień do przygotowania: m. Liczba zamównień do wysłania: x.

gdzie n to liczba pomnożona przez 3, a x jest bez aktualnie wysłanego.

Zakładamy, że równocześnie pracuje wielu pracowników 1), 2) i 3). Rozmiar tablicy z zamówieniami (w pamięci wspólnej) jest ograniczony i ustalony na etapie kompilacji. Tablica ta indeksowana jest w sposób cykliczny - po dodaniu zamówienia na końcu tablicy, kolejne zamówienia dodawane są od indeksu 0. Korzystając w odpowiedni sposób z semaforów należy zagwarantować, że liczba oczekujących zamówień nie przekroczy rozmiaru tablicy oraz że tablica nie będzie modyfikowana przez kilka procesów równocześnie. Rozmiar tablicy zamówień dobierz tak, aby mogła zajść sytuacja, w której tablica jest całkowicie zapełniona. W pamięci wspólnej oprócz tablicy można przechowywać także inne dane dzielone pomiędzy procesami. Kolejni pracownicy są uruchamiani w pętli przez jeden proces macierzysty (za pomocą funkcji fork oraz exec).

Zrealizuj powyższy problem synchronizacyjny, wykorzystując mechanizmy synchronizacji procesów oraz pamięć współdzieloną ze standardu:

1. IPC - System V (50%)
2. IPC - Posix (50%)