

Zadanie 3
Systemy Operacyjne

Autor
Maciej Proszak

1 Koncept

1.1 Opis zadania

Mamy bufor FIFO na liczby całkowite. Procesy A1 generują kolejne liczby parzyste modulo 50, jeżeli w buforze jest mniej niż 10 liczb parzystych. Procesy A2 generują kolejne liczby nieparzyste modulo 50, jeżeli liczb parzystych w buforze jest więcej niż nieparzystych. Procesy B1 zjadają liczby parzyste pod warunkiem, że bufor zawiera co najmniej 3 liczby. Procesy B2 zjadają liczby nieparzyste, pod warunkiem, że bufor zawiera co najmniej 7 liczb. W systemie może być dowolna liczba procesów każdego z typów. Zrealizuj wyżej wymienioną funkcjonalność przy pomocy semaforów. Zakładamy, że bufor FIFO poza standardowym `put()` i `get()` ma tylko metodę umożliwiającą sprawdzenie liczby na wyjściu (bez wyjmowania) oraz posiada metody zliczające elementy parzyste i nieparzyste. Zakładamy, że semafory mają tylko operacje P i V.

1.2 Kolejka FIFO

Kolejka będzie stworzona w pamięci współdzielonej, by procesy miały dostęp do kolejki. Przy każdej operacji odczytu i zapisu do kolejki będziemy blokować innym procesom dostęp do operacji za pomocą semafora binarnego. Wszystkie operacje działają w sekcji krytycznej, dlatego należy zastosować tutaj mutex.

Wszystkie operacje dostępne na kolejce to:

- `put`
- `get`
- `top`
- `numberOfEven`
- `numberOfOdd`

Zostaną dodane dodatkowe zmienne zliczające liczbę dodanych liczb parzystych oraz nieparzystych.

1.3 Semafony

Dla kolejki zostanie stworzony jeden główny semafor (`mutex`) opisany wcześniej. Dla producentów i konsumentów zostanie użytych 8 semaforów. Dla całego programu zostaną użyte łącznie 9 semaforów.

1.4 Programy weryfikujące

Zostaną stworzone łącznie 4 programy: 2 programy produkujące wartości kolejno parzyste i nieparzyste oraz 2 programy konsumujące kolejne wartości w kolejce. Nowe procesy zostaną uruchomione za pomocą poleceń `fork()`. Semafony będą oczywiście wspólne dla wszystkich procesów jak i kolejka w pamięci współdzielonej.