

Wydział Informatyki Politechniki Białostockiej Przedmiot: Systemy Operacyjne	Data: 16.06.2023
Temat: Projekt 2 Grupa: PS 1 Imię i nazwisko: Szymon Siłkin Paweł Siemieniuk Marcel Herman Michał Wysocki	Prowadzący: dr. inż. Wojciech Kwedło

Temat 1 - czytelnicy i pisarze

Czytelnicy i pisarze. Z czytelni korzysta na okrągło pewna ilość czytelników i pisarzy, przy czym jednocześnie może w niej znajdować się albo dowolna ilość czytelników, albo jeden pisarz, albo nikt - nigdy inaczej. Problem ten ma trzy rozwiązania - z możliwością zagłócenia pisarzy, z możliwością zagłócenia czytelników oraz wykluczające zagłócenie. Napisać:

- a) dwa programy symulujące dwa różne rozwiązania tego problemu, bez korzystania ze zmiennych warunkowych [17 p], lub
- b) dwa programy symulujące dwa różne rozwiązania tego problemu, przy czym jeden z nich musi korzystać ze zmiennych warunkowych (condition variable). [27 p], lub
- c) trzy programy symulujące trzy różne rozwiązania tego problemu, przy czym przynajmniej jeden z nich musi korzystać ze zmiennych warunkowych [34 p].

Ilość wątków pisarzy R i czytelników W można przekazać jako argumenty linii poleceń. Zarówno czytelnicy jak i pisarze wkrótce po opuszczeniu czytelni próbują znów się do niej dostać. Program powinien wypisywać komunikaty według poniższego przykładu:

ReaderQ: 11 WriterQ: 10 [in: R:0 W:1]

Oznacza to, że w kolejce przed czytelnią czeka 10 pisarzy i 11 czytelników a sama czytelnia zajęta jest przez jednego pisarza. Komunikat należy wypisywać w momencie zmiany którejkolwiek z tych

Realizacja punktów

Zrealizowane zostały wszystkie punkty

- Rozwiązanie z możliwością zagłódzenia pisarzy bez używania zmiennych warunkowych
- Rozwiązanie z możliwością zagłódzenia czytelników z użyciem zmiennych warunkowych
- Rozwiązanie wykluczające zagłódzenie z użyciem zmiennych warunkowych

Najistotniejsze algorytmy

Zagłódzenie pisarzy

Pisarze chcąc dostać się do biblioteki muszą czekać aż wszyscy czytelnicy będą poza biblioteką i kolejką, podczas, gdy czytelnicy sprawdzają tylko czy nie ma żadnego pisarza w bibliotece. W danej sytuacji czytelnicy będą cały czas wracać po wyjściu przed czekającymi pisarzami i nigdy nie dojdzie do sytuacji, w której ani w bibliotece, ani w kolejce nie będzie żadnego czytelnika. Ten stan doprowadza do zagłódzenia pisarzy.

Zagłódzenie czytelników

Pisarz po wyjściu z biblioteki sygnalizuje innemu pisarzowi możliwość dostępu, dopiero gdy w kolejce nie ma innych pisarzy, ten informuje wszystkich czytelników o możliwości dostępu. Jako, że pisarze cały czas powracają do kolejki, nigdy nie dojdzie do sytuacji, w której czytelnicy otrzymają sygnał dostępu, co doprowadzi do ich zagłódzenia.

Wykluczenie zagłódzenia

Pisarz po wyjściu z biblioteki informuje kolejnego pisarza o możliwości dostępu, zapamiętując, ilu pisarzy miało już dostęp w danym cyklu. Ostatni pisarz w cyklu informuje wszystkich czytelników o możliwości dostępu. Po opuszczeniu biblioteki przez czytelników, ostatni wysyła sygnał do pisarza, po czym cykl się powtarza. Taka sytuacja daje równy dostęp wszystkim wątkom.

Zmienne Globalne i Funkcje

Zmienne Globalne - "var.h"

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <pthread.h>
4 #include <stdlib.h>
5 #include <unistd.h>
6 #include <stdbool.h>
7 #include <signal.h>
8 #include "func.h"
9
10 #ifndef VAR
11 #define VAR
12
13 extern unsigned int MAX_TIME;
14 extern unsigned int REST_TIME;
15
16 typedef struct Library
17 {
18     unsigned int readers;
19     unsigned int writers;
20 }Library;
21
22 typedef struct Queue
23 {
24     unsigned int readers;
25     unsigned int writers;
26 }Queue;
27
28 extern Library lib;
29 extern Queue que;
30
31 extern pthread_mutex_t lock;
32 extern pthread_cond_t readers_lock;
33 extern pthread_cond_t writers_lock;
34
35
```

```

36 // Zlicza ilu pisarzy miało dostęp do biblioteki w jednym cyklu (wykluczenie ↵
    zagłódzenia)
37 extern unsigned int writers_counter;
38
39 extern unsigned int readers_num;
40 extern unsigned int writers_num;
41
42 extern time_t *threads_last_entry;
43 extern time_t *threads_max_wait_time;
44
45 extern time_t start_time;
46
47 #endif

```

Funkcje - "func.h"

```

1 #ifndef FUNC
2 #define FUNC
3 void printStatus();
4 /*
5  * Deklaracja funkcji do wyświetlania aktualnego stanu kolejek
6  * Wyświetla aktualny stan kolejek czytelników i pisarzy.
7  */
8 void signalHandler();
9 /*
10 * Deklaracja funkcji do obsługi sygnału
11 * Funkcja jest wywoływana w momencie obsługi sygnału.
12 * Przeprowadza ona analizę czasu działania wątków i wyświetla
13 * statystyki dotyczące oczekiwania w kolejce dla czytelników i pisarzy.
14 */
15 void logEntryTime(int id);
16 /*
17 * Deklaracja funkcji do logowania czasu wejścia wątku do kolejki
18 * Funkcja jest wywoływana, gdy wątek wchodzi do kolejki.
19 * Zapisuje ona czas wejścia wątku do kolejki oraz aktualizuje najdłuższy czas ↵
    oczekiwania w kolejce.
20 */
21 #endif

```

Instrukcja

Kompilacja

Trzy program są kompilowane przy pomocy narzędzia make następującą komendą:

```
1 make all
```

Lub każda wersja oddzielnie:

Zagłódzenie pisarzy

```
1 make build-wstarv
```

Zagłódzenie czytelników

```
1 make build-rstarv
```

Wykluczenie zagłódzenia

```
1 make build-exclude
```

Skompilowane programy trafiają do katalogu /bin

Uruchamianie i Zakańczanie

Program uruchamiany jest przy pomocy skryptu "run.sh" w następujący sposób:

```
1 ./run.sh [ OPCJE ] ... WERSJA CZYTELNICY PISARZE
```

CZYTELNICY – Liczba wątków z rolą czytelnika jakie ma stworzyć program.

PISARZE – Liczba wątków z rolą pisarza jakie ma stworzyć program.

WERSJA

"1" lub "wstarv" – Zagłódzenie pisarzy.

"2" lub "rstarv" – Zagłódzenie czytelników.

"3" lub "exclude" – Wykluczenie zagłódzenia.

OPCJE

[-m M] – Ustawienie maksymalnego czasu spędzania w bibliotece na M sekund (domyślnie 5).

[-r R] – Ustawienie czasu odpoczynku po wyjściu z biblioteki, przed dołączeniem do kolejki na R sekund (domyślnie 1).

Przykłady użycia:

```
1 ./run.sh 2 5 10
```

```
1 ./run.sh -m 10 exclude 7 13
```

Zakańczanie

Jako że program działa w nieskończonej pętli, zalecanym sposobem zamknięcia jest przerwanie go przyciskając "Ctrl + C". Program po otrzymaniu sygnału wypisuje statystyki związane z jego przebiegiem a następnie poprawnie zakańcza.

Instrukcja jest również dostępna w języku angielskim, uruchamiając skrypt z opcją "-h".