|  |  |  |
| --- | --- | --- |
| Łukasz Oprych  Gr.5 lab | Programowanie Równoległe  T: Monitory i zmienne warunku | 02.12.2023.r. |

Cel ćwiczenia:

Zapoznanie się z synchronizacją przy pomocy zmiennych warunku oraz asynchronicznym dostępem do zasobu.

Przebieg ćwiczenia:

Po pobraniu plików oraz przygotowaniu środowiska zgodnie z poleceniami prowadzącego w katalogu lab\_8\_bariera zaimplementowano algorytm realizujący funkcję bariery w utworzonym pliku bariera.c. Dodano między innymi zmienne statyczne ilosc\_watkow i ilosc\_watkow\_w\_funkcji, które związane są z ilością wątków biorących udział w przechodzeniu przez barierę oraz zmienną odpowiadającą za warunek i muteksy. W funkcji bariera() zastosowano pthread\_cond\_signal czyli zmienną rozgłaszającą sygnał, w tym przypadku, budzono wątki, po tym jak już ostatni wątek ze wszystkich dotarł do bariery. Wątki, które nie dotarły do bariery jako ostatnie były usypiane za pomocą pthread\_cond\_wait i muteksu.

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie bariera.c main.c

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Main.c

Wynik programu:

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Jak widać za każdym razem, gdy wszystkie zadane 4 wątki docierały do bariery, bariera została zwalniana, po czym wątki docierały do kolejnej bariery, do czasu aż pokonały wszystkie 4 bariery, które zadano.

Następnie po wypakowaniu kolejnych plików od prowadzącego utworzono katalog lab\_8\_pthreads. W pliku czytelnia.h zdefiniowano zmienne odpowiadające za czytelników i pisarza.

Obraz zawierający tekst, Czcionka, zrzut ekranu, design

Opis wygenerowany automatycznie

Następnie zaimplementowano śledzenie liczby czytelników i pisarzy w czytelni:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Następnie umieszczono w procedurach pisania i czytania sprawdzenia warunków poprawnych wartości aktualnych liczb pisarzy i czytelników, sprawdzamy tu również czy liczba czytelników bądź pisarzy nie jest przypadkiem ujemna. Zgodnie z problemem czytelników i pisarzy, w czytelni w jednym momencie może się znajdować 1 pisarz lub wielu czytelników, obie instrukcje umieszczono wewnątrz kompilacji warunkowej #ifdef MY\_DEBUG.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Fragment Makefile, przekazanie opcji -DMY\_DEBUG do kompilatora:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Przetestowanie programu:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Jak widać doszło do przerwania programu, warunek nie został spełniony, z powodu wystąpienia jednocześnie pisarzy oraz czytelników w jednym momencie w czytelni.

Modyfikacja kodu między innymi na podstawie pseudokodu monitora Czytelnia z wykładu:

Reprezentacja czytelników oraz pisarzy czekających na wejście do czytelni:

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

Pilnowanie aby wątki między sobą prawidłowo modyfikowały dane w czytelni oraz zmienne globalne:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Przetestowanie programu po modyfikacji:

Obraz zawierający tekst, zrzut ekranu, Czcionka, menu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Program działa w nieskończonej pętli, nie dochodzi do konfliktów w dostępie do czytelni.

Wnioski:

Aby prawidłowo wykonać oba programy użyliśmy zmiennych warunku (pthread\_cond\_wait, pthread cond\_signal) oraz muteksy (pthread\_mutex). Dzięki użyciu polecenia wait mogliśmy, zapewnić, że wątki zostaną zawieszane i będą oczekiwać na wykonanie programu przez pozostałe wątki, czy też używając signal takowe wątki wybudzać. Zmienna warunku może być dobrym zastosowaniem w kwestii ustalenia miejsca synchronizacji współpracujących wątków.