

INSTYTUT TELEINFORMATYKI I AUTOMATYKI

Wydział Cybernetyki WAT

Przedmiot: SYSTEMY OPERACYJNE

SPRAWOZDANIE Z ĆWICZENIA LABORATORYJNEGO Nr 10

Temat ćwiczenia: SYNCHRONIZACJA WĄTKÓW/MUTEX

Wykonał:

Piotr Matyjek
Grupa: **I6Y3S1**

Ćwiczenie wykonane dnia
19.12.2017

Prowadzący ćwiczenie
mgr. inż. Stanisław Matusiak

Ocena:

.....

Opis rozwiązania

Został mi przydzielony nr zadania 1. Tworzona jest zmienna globalna „wynik”, do której będą zapisywane wyniki działań przeprowadzonych w każdym wątku. Następnie inicjalizowane są dwa mutex-y. Pierwszy, który będzie dawał dostęp wątkowi do zapisu do zmiennej globalnej, a drugi, który będzie umożliwiał jej odczyt. Następnie definiowana jest struktura, która przechowuje wartość „x” i wartość „i”, która jest różna dla każdego wątku. W funkcji, którą będzie wykonywał wątek najpierw blokowany jest mutex zapisu, aby inne wątki nie mogły zapisać swoich wartości do „wynik”. Będą one czekać, aż ten mutex będzie odblokowany. Następnie wykonywane są odpowiednie obliczenia i zapisywana jest wartość do „wynik” i odblokowywany jest mutex odczytu. W funkcji głównej programu, najpierw są tworzone odpowiednie zmienne i tablice, po czym przy użyciu „scanf” konwertowany jest pierwszy argument wywołania tego programu na typ double i zapisywany jest do zmiennej x. Następnie blokowany jest mutex zapisu. Tworzone są wątki i są odłączane od programu. Następnie mutex zapisu jest odblokowywany, proces jest usypiany na 10 milisekund i w kolejnej pętli for wypisywane są identyfikatory wątków. Następnie blokowany jest mutex odczytu, czytany jest wynik i jest sumowany i odblokowywany jest mutex zapisu. Po zakończeniu pętli wyświetlana jest suma i wynik funkcji $e(x)$, dla porównania.

Kod źródłowy programu

```
#include <stdio.h>
#include <pthread.h>
#include <math.h>
#include <stdlib.h>
#include <unistd.h>

double wynik;

pthread_mutex_t mojmutex=PTHREAD_MUTEX_INITIALIZER;           //zapis
pthread_mutex_t mojmutex2=PTHREAD_MUTEX_INITIALIZER;          //odczyt

typedef struct ele
{
    double x;
    int i;
}elementy;

void *wyraz (void *arg)
{
    //pthread_mutex_lock(&mojmutex2);
    pthread_mutex_lock(&mojmutex);
    |
    elementy *point=(elementy*)arg;

    int i, f,g;
    double iks_kw, minus_jeden, x, silnia=1;

    i=point->i;
    x=point->x;

    f=i;

    iks_kw=pow(x,(double)(i));

    for(g=1;g<=f;g++)
    {
        silnia=silnia*g;
    }

    wynik=(iks_kw)/silnia;
    pthread_mutex_unlock(&mojmutex2);

    return 0;
```

```

int main(int argc, char *argv[])
{
    int N=10, i;
    double suma=0, x;
    elementy T[10];
    pthread_t L[10];

    sscanf(argv[1], "%lf", &x); //konwersja pierwszego argumentu na double

    pthread_mutex_lock(&mojmutex);

    for(i=0; i<N; i++)
    {
        T[i].x=x;
        T[i].i=i;

        pthread_create(&L[i], NULL, wyraz, &T[i]);
        pthread_detach(L[i]);
    }

    pthread_mutex_unlock(&mojmutex);
    usleep(10000);

    for(i=0; i<N; i++)
    {

        printf("%ld\t", (long)L[i]);

        pthread_mutex_lock(&mojmutex2);
        printf("%lf\n", wynik);
        suma=suma+(wynik);
        pthread_mutex_unlock(&mojmutex);
    }

    printf("suma: %f\n", suma);

    printf("exp(x)= %f\n", exp(x)); //wypisanie wartosci funkcji e(x) dla sprawdzenia poprawności

    return 0;
}

```

Wyniki uruchomienia

```
piotr@Piotr-Debian:~/Dokumenty$ gcc matyjek_mutex.c -o matyjek.o -lm -lpthread
piotr@Piotr-Debian:~/Dokumenty$ ./matyjek.o 0.71
-1219097792      1.000000
-1227490496      0.252050
-1235883200      0.010588
-1244275904      0.059652
-1252668608      0.001504
-1261061312      0.710000
-1269454016      0.000178
-1277846720      0.000018
-1286239424      0.000002
-1294632128      0.000000
suma: 2.033991
exp(x)= 2.033991
piotr@Piotr-Debian:~/Dokumenty$ ./matyjek.o 1.2
-1219368128      1.200000
-1227760832      0.086400
-1236153536      0.720000
-1244546240      0.004147
-1252938944      0.000711
-1261331648      0.288000
-1269724352      0.020736
-1278117056      0.000107
-1286509760      1.000000
-1294902464      0.000014
suma: 3.320115
exp(x)= 3.320117
piotr@Piotr-Debian:~/Dokumenty$ █
```

PODSUMOWANIE

Program został napisany w języku C i uruchomiony w środowisku systemu Linux Debian ver. 9. Program wykonuje się poprawnie, jak widać na zamieszczonych powyżej przykładach wywołania programów. Program został oddany do sprawdzenia na zajęciach laboratoryjnych i wtedy też został przesłany kod źródłowy programu na wskazany e-mail.