**Отчет лабораторной работы №4**

**Тема « Условные (сигнальные) переменные»**

**Султанов Шакир КА**

Cтуденты должны знать:

1. Условные (сигнальные) переменные;
2. Назначение фунций: pthread\_cond\_wait(), pthread\_cond\_signal(), sem\_cond\_broadcast().

**Задание**

Используя механизм синхронизации параллельных потоков «барьер», написать параллельную версию следующей последовательной программы.

#include <stdio.h>

#include <stdlib.h>

int f(int\* x ){return \*x;}

int main()

{ int x1,x2,x3,y1,y2,y3;

while(1){x1=rand() % 50;

x2=rand() % 50;

x3=rand() % 50;

y1=f(&x1);

y2=f(&x2);

y3=f(&x3);

printf("Hello, World!\n");

if((y1+y2+y3)>130)break;

}

return 0;

}

**КОД**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

int f(int\* x ){return \*x;}

pthread\_cond\_t condBarrier;

pthread\_mutex\_t mutexBarrier;

int n = 0;

int nThreads = 4;

/\*struct xy {

int\* x;

int\* y;

};\*/

void barrier() {

n++;

if (n < nThreads) {

pthread\_mutex\_lock(&mutexBarrier);

pthread\_cond\_wait(&condBarrier, &mutexBarrier);

pthread\_mutex\_unlock(&mutexBarrier);

} else {

n = 0;

pthread\_cond\_broadcast(&condBarrier);

}

}

void\* thread\_func(void\* arg) {

//struct xy xy1 = \*((struct xy\*) arg);

int\* y = (int\*) arg;

int x;

while(1) {

x = rand() % 50;

\*y = f(&x);

printf("Thread #%d computed:%d\n", (int)pthread\_self(), \*y);

barrier();

barrier();

}

//return &y;

}

int main()

{

int /\*x1,x2,x3,\*/y1,y2,y3;

int res;

/\*struct xy xy1, xy2, xy3;

xy1.x = &x1; xy1.y = &y1;

xy2.x = &x2; xy2.y = &y2;

xy3.x = &x3; xy3.y = &y3;\*/

pthread\_t t1, t2, t3;

pthread\_create(&t1, NULL, thread\_func, &y1);

pthread\_create(&t2, NULL, thread\_func, &y2);

pthread\_create(&t3, NULL, thread\_func, &y3);

while(1) {

//x1=rand() % 50;

//x2=rand() % 50;

//x3=rand() % 50;

barrier();

res = y1+y2+y3;

printf("Hello, World! Res is: %d\n", res);

if(res > 130) {

break;

}

printf("Press 'Enter' to continue\n");

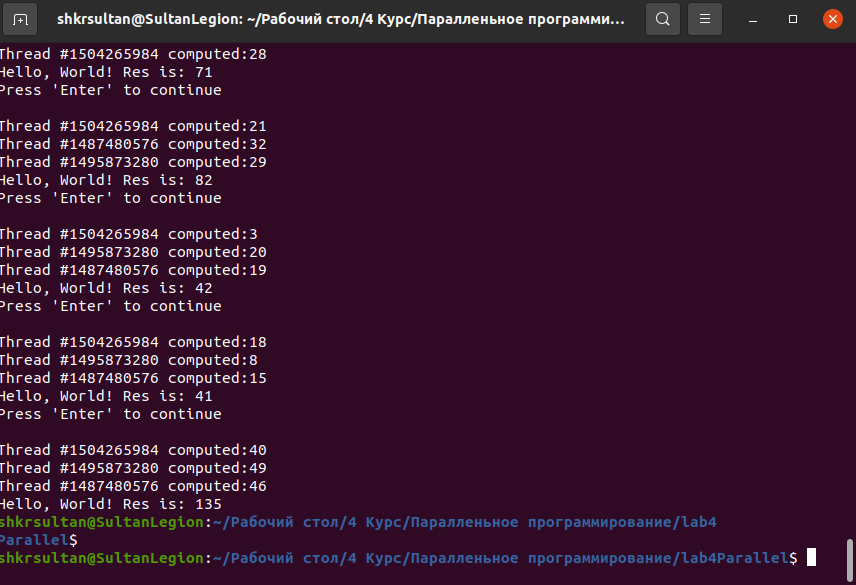
getchar();

barrier();

}

return 0;

}  
  
**Результат**

****