**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

по дисциплине «Операционные системы»

на тему: **«Программирование планировщиков процессов»**

Выполнил студент

группы ИТИ-11

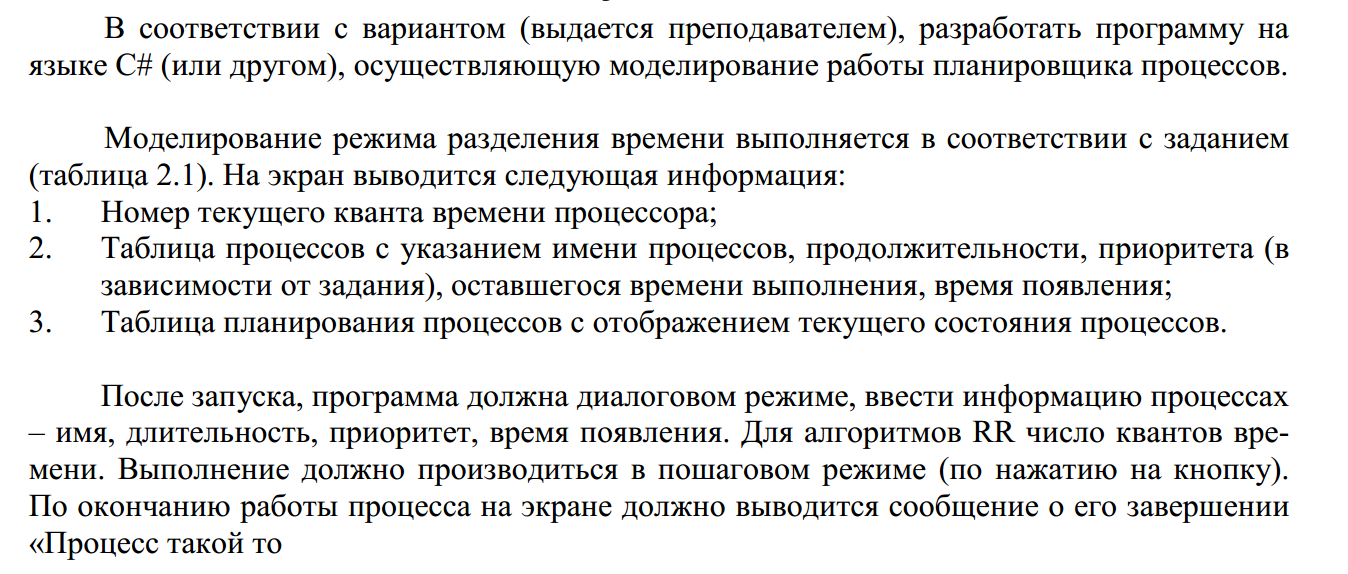
*Гаращук И. В.*

Проверил преподаватель

*Карась О.В.*

Гомель 2022

**Цель работы:** разработать программу, осуществляющую моделирование режима работы с разделение времени.

**Задание.** 

|  |  |
| --- | --- |
| **Вариант 5** | SJF невытесняющий |

При невытесняющем SJF-планировании процессор предоставляется избранному процессу на все необходимое ему время, независимо от событий, происходящих в вычислительной системе.

Результат работы программы представлен на рисунке 1:

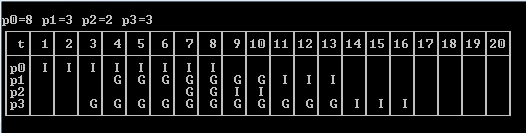


Рисунок 1 – Результат выполнения программы

**Вывод**: в ходе выполнения лабораторной работы были изучены такие способ синхронизации процессов как: переменная-замок, строгое чередование, алгоритм булочной (Bakery algorithm).

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

**Листинг задания:**

//Lab\_6

#include<stdio.h>

#include<conio.h>

#include<windows.h>

#include<math.h>

#include<iostream>

#include <string>

char bufRus[256];

char\* Rus(const char\*text)

{

CharToOem(text, bufRus);

return bufRus;

}

struct process

{

int longProcess;

int timeBirth;

char theProcess[20];

}p[4];

int main()

{

int i,j,t,k,n=0,minBirth,runProc;

p[0]={8,1};

p[1]={3,4};

p[2]={2,7};

p[3]={3,3};

puts(p[0].theProcess);

minBirth=0;

for(j=0;j<4;j++)

if(p[minBirth].timeBirth>p[j].timeBirth)

minBirth=j;

else if(p[minBirth].timeBirth==p[j].timeBirth && p[minBirth].longProcess>p[j].longProcess)

minBirth=j;

runProc=minBirth;

t=0;

printf("p0=%d p1=%d p2=%d p3=%d \n",p[0].longProcess,p[1].longProcess,p[2].longProcess,p[3].longProcess);

while(t<15){

for(i=0;i<4;i++)

if(p[runProc].longProcess>p[i].longProcess && p[i].timeBirth<=t&&p[i].longProcess>0)

runProc=i;

else if(p[runProc].longProcess==p[i].longProcess && i<runProc)

runProc=i;

n=p[runProc].longProcess+t;

for(i=t;i<n;i++){

for(j=0;j<4;j++){

p[j].theProcess[t]=' ';

if(j!=runProc&&p[j].timeBirth<=t+1 && p[j].longProcess!=0)

p[j].theProcess[t]='G';

}

p[runProc].theProcess[t]='I';

p[runProc].longProcess--;

t++;

}

//printf("p0=%d p1=%d p2=%d p3=%d run=%d t=%d\n",p[0].longProcess,p[1].longProcess,p[2].longProcess,p[3].longProcess,runProc,t);

for(i=0;i<4;i++)

if(p[i].longProcess!=0)

runProc=i;

}

printf("\xDA\xC4\xC4\xC2");

for(i=0;i<19;i++)

printf("\xC4\xC4\xC2");

printf("\xC4\xC4\xBF\n\xB3 t\xB3");

for(i=0;i<19;i++)

printf("%2d\xB3",i+1);

printf("20\xB3\n\xC3\xC4\xC4\xC5");

for(i=0;i<19;i++)

printf("\xC4\xC4\xC5");

printf("\xC4\xC4\xB4\n");

for(i=0;i<4;i++){

printf("\xB3p%d\xB3",i);

for(j=0;j<20;j++)

printf("%2c\xB3",p[i].theProcess[j]);

printf("\n");

}

printf("\xC0\xC4\xC4\xC1");

for(i=0;i<19;i++)

printf("\xC4\xC4\xC1");

printf("\xC4\xC4\xD9\n");

return 0;

}