**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 гр. ИТП-11:

Косиченко Д.А.

Принял: преподаватель-стажер:

Карась О.В.

Гомель 2022

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

**Задача:**

В соответствии с вариантом (выдается преподавателем), разработать программу на языке С# (или другом), осуществляющую моделирование работы планировщика процессов.

Моделирование режима разделения времени выполняется в соответствии с заданием (таблица 2.1). На экран выводится следующая информация:

1. Номер текущего кванта времени процессора;

2. Таблица процессов с указанием имени процессов, продолжительности, приоритета (в зависимости от задания), оставшегося времени выполнения, время появления;

3. Таблица планирования процессов с отображением текущего состояния процессов.

После запуска, программа должна диалоговом режиме, ввести информацию процессах – имя, длительность, приоритет, время появления. Для алгоритмов RR число квантов времени. Выполнение должно производиться в пошаговом режиме (по нажатию на кнопку). По окончанию работы процесса на экране должно выводится сообщение о его завершении «Процесс такой то

Примерный вариант предоставления информации о процессах.

|  |  |
| --- | --- |
| 2 | Алгоритм FCFS. Не вытесняющий, приоритетный |

**Листинг программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

List<Proces> list = new List<Proces>();

Proces pr = new Proces();

Input input = new Input();

bool f = true;

pr = new Proces("P0", 1, 10, 1, 2);

list.Add(pr);

pr = new Proces("P1", 2, 3, 4, 2);

list.Add(pr);

pr = new Proces("P2", 3, 3, 4, 1);

list.Add(pr);

pr = new Proces("P3", 4, 7, 7, 1);

list.Add(pr);

while (f)

{

Console.WriteLine("FCFS.Не вытесняющий,приоритетный");

Console.WriteLine("Выберите действие:");

Console.WriteLine("1 - Добавить процесс");

Console.WriteLine("2 - Вывести планировщик");

Console.WriteLine("3 - Выход");

int d = int.Parse(Console.ReadLine());

switch (d)

{

case 1:

{

list.Add(input.inp(list.Count));

}

break;

case 2:

{

int sum = 0;

for (int i = 0; i < list.Count; i++)

sum += list[i].dlit;

List<Proces> workList = new List<Proces>();

int start = 0;

{

int j = 0;

while (workList.Count == 0)

{

for (int i = 0; i < list.Count; i++)

if (list[i].timeStart == j) workList.Add(list[i]);

j++;

}

start = j--;

}

sum += start;

int[,] array = new int[list.Count, sum];

for (int k = 0; k < list.Count; k++)

{

int id = minValue\_id(workList);

int dlit = minValue\_dlit(workList);

for (int j = start; j < start + dlit && j < sum; j++)

array[id - 1, j] = 1;

for (int j = 0; j < workList.Count; j++)

if (workList[j].id == id)

{

workList.RemoveAt(j);

break;

}

for (int j = start + dlit; j < sum; j++)

array[id - 1, j] = 2;

for (int j = start + 1; j < start + dlit && j < sum; j++)

for (int i = 0; i < list.Count; i++)

if (list[i].timeStart == j) workList.Add(list[i]);

start += dlit;

}

for (int i = 0; i < list.Count; i++)

for (int j = 0; j < list[i].timeStart+1; j++)

array[i, j] = 2;

Console.WriteLine("Название Длительность Время появления Приоритет");

for (int i = 0; i < list.Count; i++)

Console.WriteLine("{0,5} {1,9} {2,14}{3,14}", list[i].name, list[i].dlit, list[i].timeStart, list[i].priority);

Console.WriteLine();

Console.WriteLine();

for (int i = 0; i < list.Count; i++)

{

for (int j = 0; j < sum; j++)

{

switch (array[i, j])

{

case 0:

{

Console.ForegroundColor = ConsoleColor.DarkGreen;

Console.Write("Г ");

Console.ForegroundColor = ConsoleColor.Gray;

}

break;

case 1:

{

Console.ForegroundColor = ConsoleColor.Blue;

Console.Write("И ");

Console.ForegroundColor = ConsoleColor.Gray;

}

break;

case 2:

Console.Write(" ");

break;

}

}

Console.WriteLine();

}

}

break;

case 3:

{

f = false;

}

break;

default:

Console.WriteLine("Выберите правильное действие");

break;

}

}

Console.ReadLine();

}

public static int minValue\_id(List<Proces> workList)

{

int min = workList[0].priority;

int id = workList[0].id;

for (int i = 0; i < workList.Count; i++)

{

if (min > workList[i].priority)

{

min = workList[i].priority;

id = workList[i].id;

}

}

return id;

}

public static int minValue\_dlit(List<Proces> workList)

{

int min = workList[0].priority;

int dlit = workList[0].dlit;

for (int i = 0; i < workList.Count; i++)

{

if (min > workList[i].priority)

{

min = workList[i].priority;

dlit = workList[i].dlit;

}

}

return dlit;

}

}

class Input

{

public Proces inp(int idProces)

{

string name;

int dlit, timeStart, priority;

idProces++;

Proces pr = new Proces();

Console.WriteLine("Введите имя");

name = Console.ReadLine();

Console.WriteLine("Введите длительность");

dlit = int.Parse(Console.ReadLine());

Console.WriteLine("Введите время появления");

timeStart = int.Parse(Console.ReadLine());

Console.WriteLine("Введите приоритет");

priority = int.Parse(Console.ReadLine());

pr = new Proces(name, idProces, dlit, timeStart, priority);

return pr;

}

}

}

В результате на экране отобразиться диалоговое окно, в котором будет отображено меню, с помощью которого можно добавить процесс или вывести на экран планировщик процессов (рисунок 1).



Рисунок 1 – Планировщик процессов

**Вывод:** в ходе лабораторной работы была разработана программа, осуществляющая моделирование режима работы с разделение времени.