Министерство образования и науки Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

Кафедра информационных систем и программирования

Институт Компьютерных систем и информационной безопасности

**ОТЧЕТ**

по лабораторной работе №3

«Управление ресурсами однопроцессорных систем оперативной обработки»

Дисциплина: Архитектура ЭВМ

Выполнил: студент 2 курса

группы 20-КБ-ПР2

Нажжар Каис А.М

Проверил: Павел М.У

Краснодар

2021

**1 Цель работы**

Моделирование вычислительной однопроцессорной системы оперативной обработки.

**2 Задание**

1. Смоделировать работу для простого случая (решение задачи без прерывания).

2. Смоделировать реализацию алгоритма SPT через алгоритм RR.

3. Построить характеристику I и II системы:

– среднее время пребывания короткой заявки в системе;

– степень загрузки процессора - вероятность занятого состояния.

Входными данными являются :

– вероятность прихода сложной заявки (R)(в процентах);

– максимальная длительность решения сложной задачи (L);

– длительность кванта времени, уделяемого одной задаче (LK).



**3 Код программы**

using System;

using System.Collections.Generic;

namespace lab3

{

class Program

{

static void Main(string[] args)

{

Console.Title = "Процессор: обработка задач";

const int N = 10; //количество заявок

Console.Write("Введите вероятность прихода большой заявки (R): ");

int R = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите максимальную длительность решения сложной задачи (L): ");

int L = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите длительность кванта времени, уделяемого одной задаче (LK): ");

int LK = Convert.ToInt32(Console.ReadLine());

Console.CursorVisible = false;

int C;

int CountTinyWork = 0;

int CountQuant = 0;

int LastSmallWork = 0;

Queue<int> A = new Queue<int>();

Queue<int> B = new Queue<int>();

Random r = new Random();

Console.WriteLine("--------------------");

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

{

if(r.Next(100)<R)

{

C = r.Next(LK + 1, L);

A.Enqueue(C);

B.Enqueue(C);

Console.WriteLine("Большая заявка: "+C);

}

else

{

C = r.Next(1, LK);

A.Enqueue(C);

B.Enqueue(C);

CountTinyWork++;

Console.WriteLine("Маленькая заявка: " + C);

}

}

Console.WriteLine("--------------------");

Console.WriteLine("-Модель для простого случая (без прерываний)-");

while (A.Count!=0)

{

C = A.Dequeue();

if (C > LK)

{

Console.WriteLine("Выполняется обработка... Большая задача: "+C);

do

{

C -= LK;

CountQuant++;

if (C > 0)

{

Console.WriteLine("Выполняется обработка... Остаток большой задачи: " + C);

}

else

{

Console.WriteLine("Большая задача успешно исчерпана.");

}

} while (C > 0);

}

else

{

Console.WriteLine("Выполняется обработка... Маленькая задача: "+ C);

CountQuant++;

LastSmallWork = CountQuant;

}

}

Console.WriteLine("======================================================================");

Console.Write("Среднее время пребывания короткой заявки в системе в квантах: ");

double middleTime = (double)LastSmallWork / CountTinyWork;

Console.WriteLine("{0}", middleTime);

Console.Write("Cтепень загрузки процессора - вероятность занятого состояния: ");

double chance = (double)(CountQuant - CountTinyWork) / CountQuant;

Console.WriteLine("{0:f2}", chance);

Console.WriteLine("======================================================================");

Console.WriteLine("-Модель SPT через RR-");

while (B.Count!=0)

{

C = B.Dequeue();

if (C > LK)

{

Console.WriteLine("Выполняется обработка... Большая задача: "+C);

C -= LK;

Console.WriteLine("Выполняется обработка... Остаток большой задачи: " + C);

Console.WriteLine("Перемещение остатка в конец очереди.");

B.Enqueue(C);

}

else

{

Console.WriteLine("Выполняется обработка... Маленькая задача: "+C);

}

CountQuant++;

}

Console.WriteLine("======================================================================");

Console.Write("Среднее время пребывания короткой заявки в системе в квантах: ");

middleTime = (double)LastSmallWork / CountTinyWork;

Console.WriteLine("{0}", middleTime);

Console.Write("Cтепень загрузки процессора - вероятность занятого состояния: ");

chance = (double)(CountQuant - CountTinyWork) / CountQuant;

Console.WriteLine("{0:f2}", chance);

Console.WriteLine("======================================================================");

Console.ReadKey();

}

}

}

**4 Результат выполнения программы**



