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 № 9

по дисциплине «Операционные системы»

на тему: «Простейшие схемы управления памятью»

Выполнила студентка группы ИТИ-11

Житко А.С.

Проверила преподаватель-стажер

Карась О.В.

Гомель 2022

**Лабораторная работа №9**

«Простейшие схемы управления памятью»

**Цель работы:** изучение алгоритмов управления памятью, разработка программы менеджера памяти.

**Задача 1.**

Разработать программу, реализующую заданный алгоритм выделения памяти.

Менеджер памяти должен: По запросу процесса выделять память, согласно заданного алгоритма (таблица 1.1). На экран должна выводиться следующая информация о состоянии памяти: объем памяти, объём свободной памяти, размер наибольшего свободного блока, количество запросов на выделение памяти, количество удовлетворённых запросов (%).

Таблица 1.1

|  |  |
| --- | --- |
| Вариант | Задание |
| 10 | Схема с переменными разделами. Стратегия наиболее подходящего |

**Ход работы**

**Порядок выполнения задания 1.**

На рисунке 1.1 показан запуск программы, в котором необходимо ввести исходные данные.

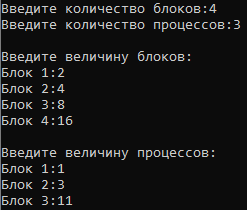


Рисунок 1.1 – Запуск программы

Рисунок 1.2 представляет собой окно с результатом выполнения программы.

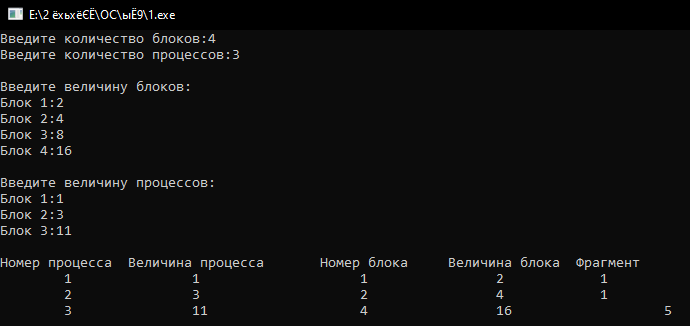


Рисунок 1.2 – Окно с результатом выполнения фрагмента программы

**Вывод:** в ходе выполнения лабораторной работы изучили алгоритмы управления памятью и получили навыки разработки программы менеджера памяти. В результате проделанной работы на языке программирования С++ была создана, протестирована и отлажена одна программа.

**ПРИЛОЖЕНИЕ А**

**Листинги программ**

**Листинг программы задания 1:**

#include <stdio.h>

#include <iostream>

int main()

{

setlocale(LC\_ALL,"Rus");

int fragment[20],b[20],p[20],i,j,nb,np,temp,lowest=9999;

int barray[20],parray[20];

printf("Введите количество блоков:");

scanf("%d",&nb);

printf("Введите количество процессов:");

scanf("%d",&np);

printf("\nВведите величину блоков:\n");

for(i=1;i<=nb;i++)

{

printf("Блок %d:",i);

scanf("%d",&b[i]);

}

printf("\nВведите величину процессов:\n");

for(i=1;i<=np;i++)

{

printf("Блок %d:",i);

scanf("%d",&p[i]);

}

for(i=1;i<=np;i++)

{

for(j=1;j<=nb;j++)

{

if(barray[j]!=1)

{

temp=b[j]-p[i];

if(temp>=0)

if(lowest>temp)

{

parray[i]=j;

lowest=temp;

}

}

}

fragment[i]=lowest;

barray[parray[i]]=1;

lowest=10000;

}

printf("\nНомер процесса\tВеличина процесса\tНомер блока\tВеличина блока\tФрагмент");

for(i=1;i<=np && parray[i]!=0;i++)

printf("\n\t%d\t\t%d\t\t %d\t\t %d\t\t %d",i,p[i],parray[i],b[parray[i]],fragment[i]);

}