**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

Король В.Н.

Принял: преподаватель

Карась О.В.

Гомель 2022

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

**Задача:** В соответствии с вариантом (выдаётся 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.

|  |  |
| --- | --- |
| 3. | Алгоритм FCFS. Не вытесняющий |

Результат выполнения первого задания указан на рисунке 1.

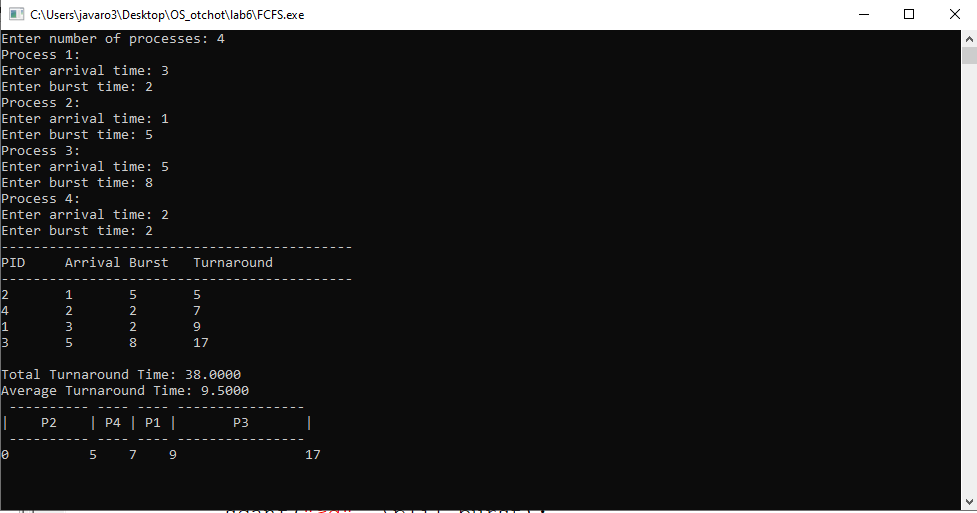


Рисунок 1 – пример реализации алгоритма FCFS. Не вытесняющий

**Вывод:** В результате выполнения лабораторной работы более углублённо были усвоены алгоритмы планирования и был разработан планировщик на языке программирования *C*.

**Листинг программы**

#include<stdio.h>

#include<string.h>

#include<conio.h>

struct PCB

{

int pid, arrival, burst, turnaround;

};

int main()

{

int i, num, j;

float avg = 0, sum = 0;

struct PCB p[10], temp;

printf("Enter number of processes: ");

scanf("%d", &num);

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

{

printf("Process %d:\n", i+1);

printf("Enter arrival time: ");

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

printf("Enter burst time: ");

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

p[i].pid = i+1;

}

for(i = 0; i < num-1; i++)

{

for(j = 0; j < num-1; j++)

{

if(p[j].arrival > p[j+1].arrival)

{

temp = p[j];

p[j] = p[j+1];

p[j+1] = temp;

}

}

}

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

{

sum = sum + p[i].burst;

p[i].turnaround = sum;

}

sum = 0;

printf("--------------------------------------------");

printf("\n");

printf("PID\tArrival\tBurst\tTurnaround\n");

printf("--------------------------------------------");

printf("\n");

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

{

printf("%d\t%d\t%d\t%d\n",p[i].pid,p[i].arrival,p[i].burst,p[i].turnaround);

sum += p[i].turnaround;

}

avg = sum/(float)num;

printf("\nTotal Turnaround Time: %.4f", sum);

printf("\nAverage Turnaround Time: %.4f", avg);

printf("\n ");

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

{

for(j = 0; j < p[i].burst; j++)

printf("--");

printf(" ");

}

printf("\n|");

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

{

for(j = 0; j < p[i].burst - 1; j++)

printf(" ");

printf("P%d", p[i].pid);

for(j = 0; j < p[i].burst - 1; j++)

printf(" ");

printf("|");

}

printf("\n ");

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

{

for(j = 0; j < p[i].burst; j++)

printf("--");

printf(" ");

}

printf("\n");

printf("0");

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

{

for(j = 0; j < p[i].burst; j++)

printf(" ");

printf("%d", p[i].turnaround);

}

fflush(stdin);

getchar();

return 0;

}