

PROJECT REPORT (CSF102)

ON

FCFS SCHEDULING ALGORITHM

A report submitted in partial fulfilment of the requirement for the course

DATA STRUCTURES

Part of the degree of
BACHELOR OF TECHNOLOGY

In

Computer Engineering



Submitted to:

Dr. Mitali Srivastava

Assistant Professor

Submitted by:

Name-Bhavish

SAP ID -1000015397

SCHOOL OF COMPUTING
DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

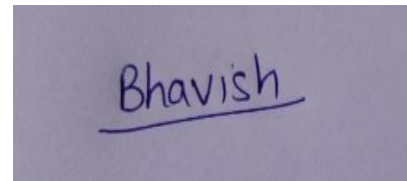
Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India

2021

CANDIDATES DECLARATION

I hereby certify that the work, which is being presented in the Report, entitled **FCFS Scheduling Algorithm**, in partial fulfilment of the requirement as part of the course Data Structures of the Degree of **Bachelor of Technology** and submitted to the DIT University is an authentic record of my work carried out during the period *2 may 2021* to *17 may 2021* under the guidance of **Dr. Mitali Srivastava** .

Date: 17 may 2021

A rectangular box containing a handwritten signature in blue ink. The signature reads "Bhavish" with a horizontal line underneath it.

BHAVISH-1000015397-K

ACKNOWLEDGEMENT

I take this opportunity with much pleasure to thank all the people who have helped me through the course of my journey towards producing this project. I sincerely thank my project guide, Dr. Mitali Srivastava, for her guidance, help and motivation. Apart from the subject of my research, I learnt a lot from her, which I am sure will be useful in different stages of my life I am especially grateful to my colleagues for their assistance, criticisms and useful insights. I am thankful to all the other B.Tech students of DIT UNIVERSITY with whom I share tons of fun memories. I would like to acknowledge the support and encouragement of my friends. My sincere gratitude also goes to all those who instructed and taught me through the years. Finally, this project would not have been possible without the confidence, endurance and support of my family. My family has always been a source of inspiration and encouragement. I wish to thank my parents, whose love, teachings and support have brought me this far.

Name- Bhavish

Roll No- 200102404

SAP ID- 1000015397

ABSTRACT

This is a mini project that helps users to find out the solution of problems based on FCFS Scheduling Algorithm. With this project, we can perform the basic problems that we do in our day-to-day life. The entire project is implemented using the C programming language. The source code of this project is around 85 lines and included in this report. The majority of the task is implemented using functions and files. Files are used for storing the user information and the tasks are implemented through functions.

TABLE OF CONTENT

S.NO	Chapter Name	Page No
1.	Introduction	1
2.	What is FCFS Scheduling	3
	2.1 Characteristics of FCFS method	3
	2.2 FCFS scheduling	3
	2.3 Example of Waiting Time	4
	2.4 Advantages of FCFS	5
	2.5 Disadvantages of FCFS	5
	2.6 About this project	6
	2.7 Function description	6
3.	Algorithms	7
4.	Code	8
5.	Implementation and screenshots	10
6.	Conclusion	11
7.	Reference	11

Introduction

Introduction to C Programming Language

C is a general-purpose programming language that is extremely popular, simple and flexible. It is machine-independent, structured programming language which is used extensively in various applications.

C was the basic language to write everything from operating systems (Windows and many others) to complex programs like the Oracle database, Git, Python interpreter and more.

It is said that 'C' is a god's programming language. One can say, C is a base for the programming. If you know 'C,' you can easily grasp the knowledge of the other programming languages that uses the concept of 'C'

It is essential to have a background in computer memory mechanisms because it is an important aspect when dealing with the C programming language.

History

The base or father of programming languages is 'ALGOL.' It was first introduced in 1960. 'ALGOL' was used on a large basis in European countries. 'ALGOL' introduced the concept of structured programming to the developer community. In 1967, a new computer programming language was announced called as 'BCPL' which stands for Basic Combined Programming Language. BCPL was designed and developed by Martin Richards, especially for writing system software. This was the era of programming languages. Just after three years, in 1970 a new programming language called 'B' was introduced by Ken Thompson that contained multiple features of 'BCPL.' This programming language was created using UNIX operating system at AT&T and Bell Laboratories. Both the 'BCPL' and 'B' were system programming languages. Student Name SAP ID Branch Languages such as C++/Java are developed from 'C'. These languages are widely used in various technologies. Thus, 'C' forms a base for many other languages that are currently in use.

Uses of C Programming Language

1. C' language is widely used in embedded systems.
2. It is used for developing system applications.
3. It is widely used for developing desktop applications.
4. Most of the applications by Adobe are developed using 'C' programming language.
5. It is used for developing browsers and their extensions. Google's Chromium is built using 'C' programming language.
6. It is used to develop databases. MySQL is the most popular database software which is built using 'C'.
7. It is used in developing an operating system. Operating systems such as Apple's OS X, Microsoft's Windows, and Symbian are developed using 'C' language. It is used for developing desktop as well as mobile phone's operating system.
8. It is used for compiler production.
9. It is widely used in IOT applications.

How C program works?

C is a compiled language. A compiler is a special tool that compiles the program and converts it into the object file which is machine readable. After the compilation process, the linker will combine different object files and creates a single executable file to run the program. The following diagram shows the execution of a 'C' program Nowadays, various compilers are available online, and you can use any of those compilers. The functionality will never differ and most of the compilers will provide the features required to execute both 'C' and 'C++' programs.

Following is the list of popular compilers available online:

- Clang compiler
- Student Name
- SAP ID
- Branch
- MinGW compiler (Minimalist GNU for Windows)
- Portable 'C' compiler
- Turbo C

What is FCFS Scheduling?

First Come First Serve (FCFS) is an operating system scheduling algorithm that automatically executes queued requests and processes in order of their arrival. It is the easiest and simplest CPU scheduling algorithm. In this type of algorithm, processes which requests the CPU first get the CPU allocation first. This is managed with a FIFO queue. The full form of FCFS is First Come First Serve.

As the process enters the ready queue, its PCB (Process Control Block) is linked with the tail of the queue and, when the CPU becomes free, it should be assigned to the process at the beginning of the queue.

Characteristics of FCFS method

- It supports non-preemptive and pre-emptive scheduling algorithm.
- Jobs are always executed on a first-come, first-serve basis.
- It is easy to implement and use.
- This method is poor in performance, and the general wait time is quite high.

FCFS scheduling

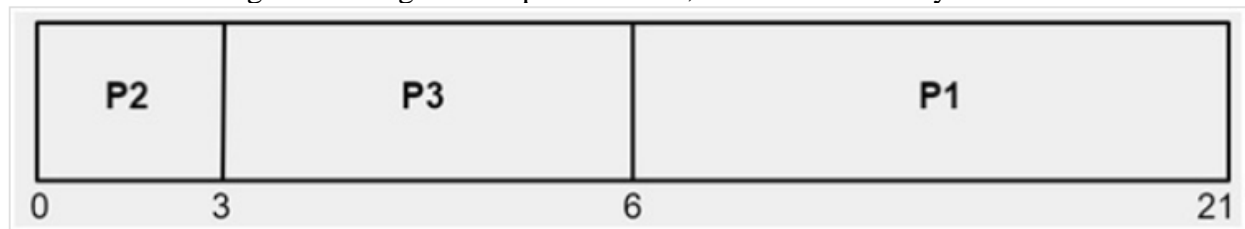
A real-life example of the FCFS method is buying a movie ticket on the ticket counter. In this scheduling algorithm, a person is served according to the queue manner. The person who arrives first in the queue first buys the ticket and then the next one. This will continue until the last person in the queue purchases the ticket. Using this algorithm, the CPU process works in a similar manner.

Example of Waiting Time

Let's say, there are four processes arriving in the sequence as P2, P3, P1 with their corresponding execution time as shown in the table below. Also, taking their arrival time to be 0.

Process	Order of arrival	Execution time in msec
P1	3	15
P2	1	3
P3	2	3

Gantt chart showing the waiting time of processes P1, P2 and P3 in the system



As shown above,

The waiting time of process P2 is 0

The waiting time of process P3 is 3

The waiting time of process P1 is 6

Average time = $(0 + 3 + 6) / 3 = 3$ msec.

As we have taken arrival time to be 0 therefore turn around time and completion time will be same.

Completion Time: Time taken for the execution to complete, starting from arrival time.

Turn Around Time: Time taken to complete after arrival. In simple words, it is the difference between the Completion time and the Arrival time.

Waiting Time: Total time the process has to wait before it's execution begins. It is the difference between the Turn Around time and the Burst time of the process.

Advantages of FCFS

Here, are pros/benefits of using FCFS scheduling algorithm:

- The simplest form of a CPU scheduling algorithm
- Easy to program
- First come first served

Disadvantages of FCFS

Here, are cons/ drawbacks of using FCFS scheduling algorithm:

- It is a Non-Preemptive CPU scheduling algorithm, so after the process has been allocated to the CPU, it will never release the CPU until it finishes executing.
- The Average Waiting Time is high.
- Short processes that are at the back of the queue have to wait for the long process at the front to finish.
- Not an ideal technique for time-sharing systems.
- Because of its simplicity, FCFS is not very efficient.

About this program

This is a mini project that helps users to find out the problems based on FCFS Scheduling algorithms. With this project, we can perform the basic problems that we do in our day-to-day life. The entire project is implemented using the C programming language. The source code of this project is around 85 lines and included in this report. The majority of the task is implemented using functions.

Function Description

The source code for user in FCFS Scheduling Project is relatively short and easy to understand. I have divided this mini project into many functions, most of which are related to different FCFS scheduling. Listed below are some of the more important functions which may help you understand the project better.

int (int, int):-Function declaration is required when you define a function in one source file and you call that function in another file. In such case, you should declare the function at the top of the file calling the function.

float:-Float is a datatype which is used to represent the floating point numbers. It is a 32-bit IEEE 754 single precision floating point number (1-bit for the sign, 8-bit for exponent, 23*-bit for the value. It has 6 decimal digits of precision.

for:- A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Algorithm

Start

Step 1-> In function int waitingtime(int proc[], int n, int burst_time[], int wait_time[])

Set wait_time[0] = 0

Loop For i = 1 and i < n and i++

Set wait_time[i] = burst_time[i-1] + wait_time[i-1]

End For

Step 2-> In function int turnaroundtime(int proc[], int n, int burst_time[], int wait_time[], int tat[])

Loop For i = 0 and i < n and i++

Set tat[i] = burst_time[i] + wait_time[i]

End For

Step 3-> In function int avgtime(int proc[], int n, int burst_time[])

Declare and initialize wait_time[n], tat[n], total_wt = 0, total_tat = 0;

Call waitingtime(proc, n, burst_time, wait_time)

Call turnaroundtime(proc, n, burst_time, wait_time, tat)

Loop For i=0 and i<n and i++

Set total_wt = total_wt + wait_time[i]

Set total_tat = total_tat + tat[i]

Print process number, bursttime wait time and turnaround time

End For

Print "Average waiting time =i.e. total_wt / n

Print "Average turn around time = i.e. total_tat / n

Step 4-> In int main()

Declare the input int proc[] = { 1, 2, 3}

Declare and initialize n = sizeof proc / sizeof proc[0]

Declare and initialize burst_time[] = {12, 6, 10}

Call avgtime(proc, n, burst_time)

Stop

CODE

```
#include<stdio.h>

int main()

{
    int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;
    printf("\nBHAVISH,1000015397\n");
    printf("Enter total number of processes: ");
    scanf("%d",&n);

    printf("\nEnter Process Burst Time\n");
    for(i=0;i<n;i++)
    {
        printf("  \nP[%d]:  \n",i+1);
        scanf("  %d",&bt[i]);
    }

    wt[0]=0;

    for(i=1;i<n;i++)
    {
        wt[i]=0;
        for(j=0;j<i;j++)
            wt[i]+=bt[j];
    }

    printf("\n  Process      Burst Time      Waiting Time      Turnaround Time\n");

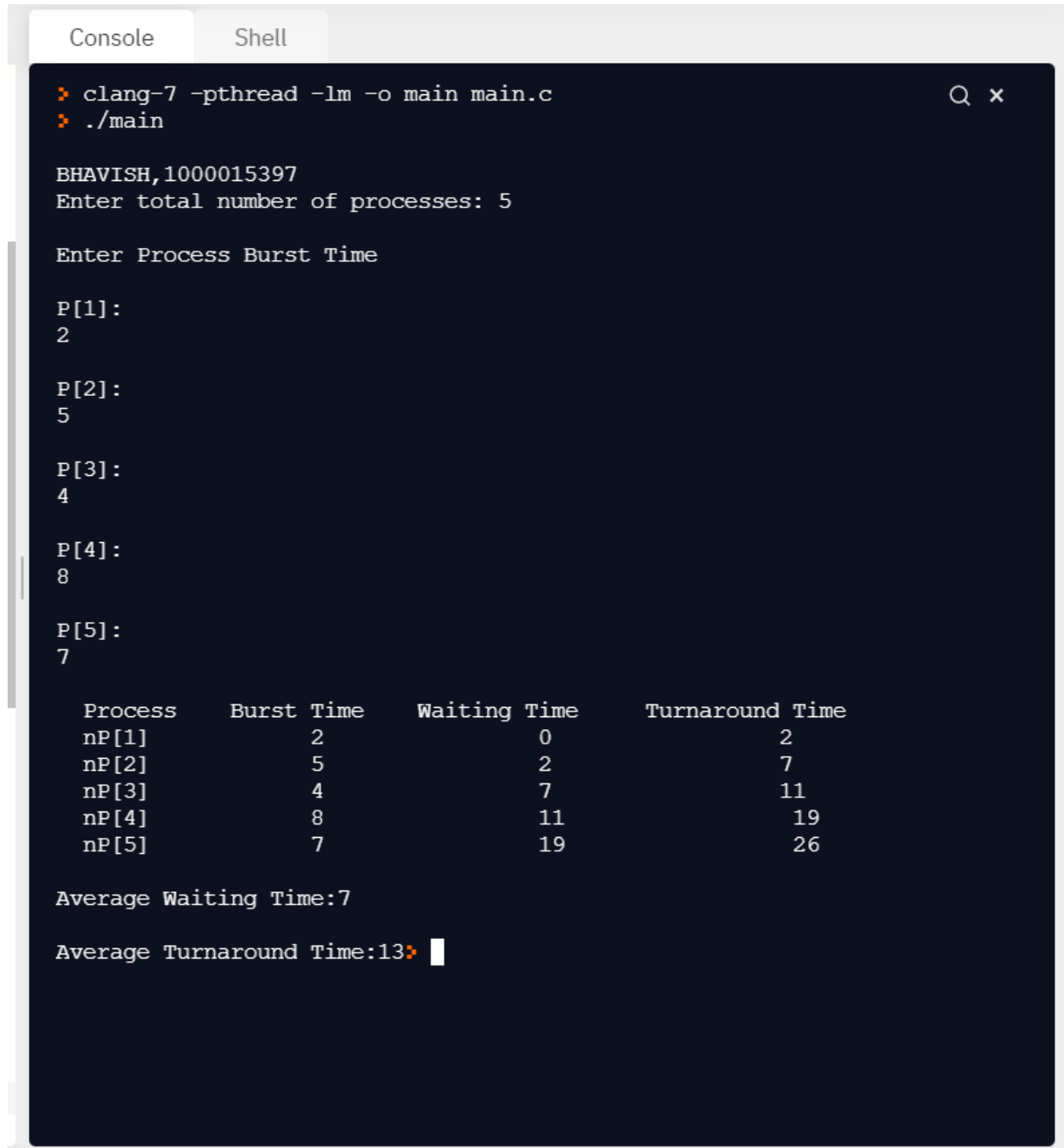
    for(i=0;i<n;i++)
    {
        tat[i]=bt[i]+wt[i];
        avwt+=wt[i];
        avtat+=tat[i];
        printf("  nP[%d]          %d          %d          %d\n",i+1,bt[i],wt[i],tat[i]);
    }

    avwt/=i;
    avtat/=i;
    printf("\nAverage Waiting Time:%d\n",avwt);
    printf("\nAverage Turnaround Time:%d",avtat);

    return 0;
}
```

}

Implementation and Screen Shots



```
clang-7 -pthread -lm -o main main.c
./main

BHAVISH,1000015397
Enter total number of processes: 5

Enter Process Burst Time

P[1]:
2

P[2]:
5

P[3]:
4

P[4]:
8

P[5]:
7

  Process    Burst Time    Waiting Time    Turnaround Time
nP[1]         2             0                2
nP[2]         5             2                7
nP[3]         4             7               11
nP[4]         8            11               19
nP[5]         7            19               26

Average Waiting Time:7

Average Turnaround Time:13
```

Conclusion

Summary:

- Definition: FCFS is an operating system scheduling algorithm that automatically executes queued requests and processes by order of their arrival
- It supports non-preemptive and pre-emptive scheduling algorithm.
- FCFS stands for First Come First Serve
- A real-life example of the FCFS method is buying a movie ticket on the ticket counter.
- It is the simplest form of a CPU scheduling algorithm
- It is a Non-Preemptive CPU scheduling algorithm, so after the process has been allocated to the CPU, it will never release the CPU until it finishes executing.

This FCFS scheduling is successful. Source code and algorithm of all questions are correct without having any error or warning that's why output values of all the inputs are correct.

References

Encyclopedia:

<https://www.encyclopedia.com/computing/dictionaries-thesauruses-pictures-and-press-releases/menu-driven-program>
[date-13/05/2021](#)

Google scholar:

<https://scholar.google.com/>
[date-15/05/2021](#)