

# OS LAB-5:

## CPU SCHEDULING

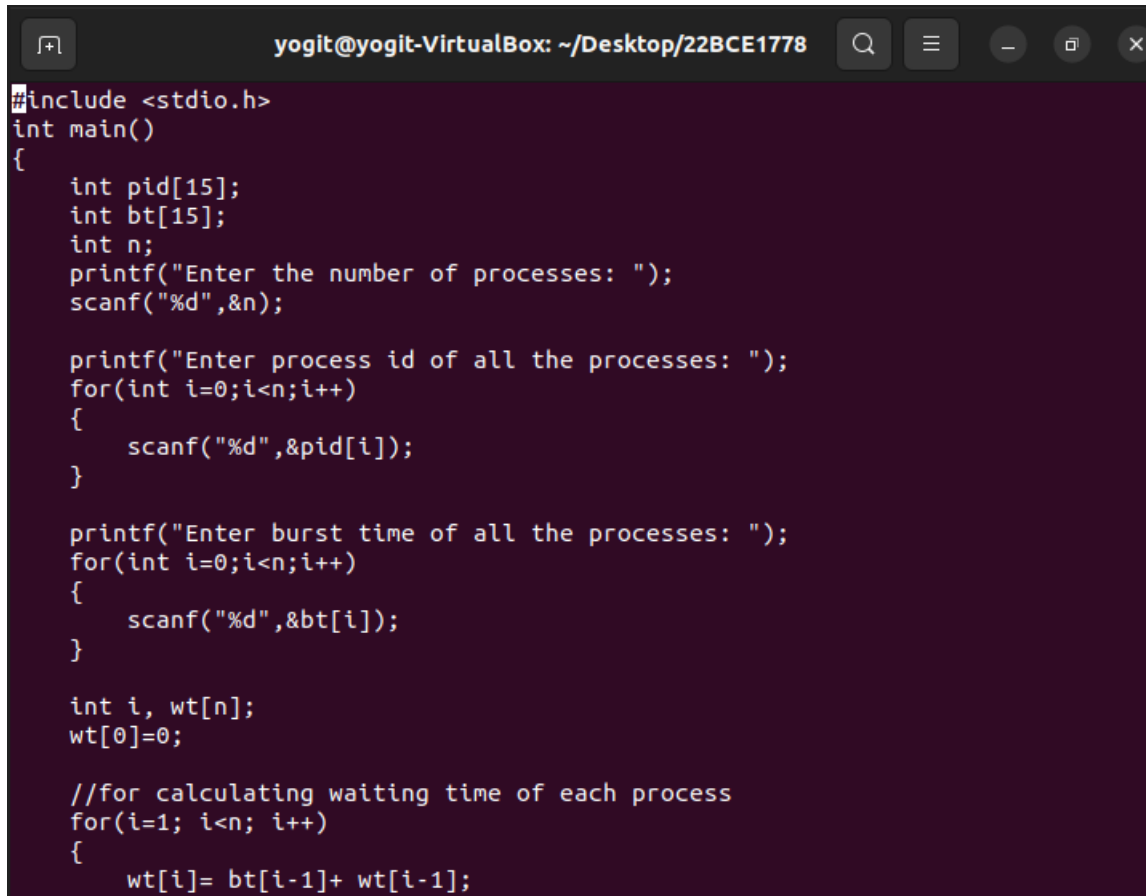
## ALGORITHMS:

J.Akash Siva Teja

22BCE1778

FIRST COME FIRST SERVE(FCFS) CPU  
SCHEDULING ALGORITHM:

CODE:

A screenshot of a code editor window titled 'yogit@yogit-VirtualBox: ~/Desktop/22BCE1778'. The window shows a C program for the First Come First Serve (FCFS) CPU scheduling algorithm. The code includes standard headers, declares arrays for process IDs (pid) and burst times (bt), and prompts the user to input the number of processes, their IDs, and their burst times. It then calculates the waiting time (wt) for each process based on the FCFS algorithm.

```
#include <stdio.h>
int main()
{
    int pid[15];
    int bt[15];
    int n;
    printf("Enter the number of processes: ");
    scanf("%d",&n);

    printf("Enter process id of all the processes: ");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&pid[i]);
    }

    printf("Enter burst time of all the processes: ");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&bt[i]);
    }

    int i, wt[n];
    wt[0]=0;

    //for calculating waiting time of each process
    for(i=1; i<n; i++)
    {
        wt[i]= bt[i-1]+ wt[i-1];
    }
}
```

```
yogit@yogit-VirtualBox: ~/Desktop/22BCE1778
//for calculating waiting time of each process
for(i=1; i<n; i++)
{
    wt[i]= bt[i-1]+ wt[i-1];
}

printf("Process      Burst Time      Waiting Time      TurnAround Time\n");
float twt=0.0;
float tat= 0.0;
for(i=0; i<n; i++)
{
    printf("%d\t\t", pid[i]);
    printf("%d\t\t", bt[i]);
    printf("%d\t\t", wt[i]);

    //calculating and printing turnaround time of each process
    printf("%d\t\t", bt[i]+wt[i]);
    printf("\n");

    //for calculating total waiting time
    twt += wt[i];

    //for calculating total turnaround time
    tat += (wt[i]+bt[i]);
}
}
```

## OUTPUT:

```
yogit@yogit-VirtualBox:~/Desktop/22BCE1778$ cc os.c
yogit@yogit-VirtualBox:~/Desktop/22BCE1778$ ./a.out
Enter the number of processes: 3
Enter process id of all the processes: 1
2
3
Enter burst time of all the processes: 24
3
3
Process      Burst Time      Waiting Time      TurnAround Time
1              24              0              24
2              3              24              27
3              3              27              30
yogit@yogit-VirtualBox:~/Desktop/22BCE1778$
```

## SHORTEST JOB FIRST(SJF) ALGORITHM:

### CODE:

```

#include <stdio.h>
int main()
{
    int A[100][4];
    int i, j, n, total = 0, index, temp;
    float avg_wt, avg_tat;
    printf("Enter number of process: ");
    scanf("%d", &n);
    printf("Enter Burst Time:\n");
    for (i = 0; i < n; i++) {
        printf("P%d: ", i + 1);
        scanf("%d", &A[i][1]);
        A[i][0] = i + 1;
    }
    for (i = 0; i < n; i++) {
        index = i;
        for (j = i + 1; j < n; j++)
            if (A[j][1] < A[index][1])
                index = j;
        temp = A[i][1];
        A[i][1] = A[index][1];
        A[index][1] = temp;

        temp = A[i][0];
        A[i][0] = A[index][0];
        A[index][0] = temp;
    }
    A[0][2] = 0;
    for (i = 1; i < n; i++) {
        A[i][2] = 0;
        for (j = 0; j < i; j++)
            A[i][2] += A[j][1];
        total += A[i][2];
    }
    total = 0;
    printf("P\t\t\t\tBT\t\t\t\tWT\t\t\t\tTAT\n");
    for (i = 0; i < n; i++) {
        A[i][3] = A[i][1] + A[i][2];
        total += A[i][3];
        printf("P%d\t\t\t\t%d\t\t\t\t%d\t\t\t\t%d\n", A[i][0],
            A[i][1], A[i][2], A[i][3]);
    }
}

```

## OUTPUT:

```
ex5@hostssh:~/22BCE1778$ cc osf.c
ex5@hostssh:~/22BCE1778$ ./a.out
Enter number of process: 4
Enter Burst Time:
P1: 6
P2: 8
P3: 7
P4: 3
P      BT      WT      TAT
P4      3      0      3
P1      6      3      9
P3      7      9     16
P2      8     16     24
ex5@hostssh:~/22BCE1778$ vi os1.c
ex5@hostssh:~/22BCE1778$
```