4ITRC2 Operating System Lab

Lab assignment 5

Name:-pawan kasde

roll no.:-23I4051

Aim: To create C programs for the different scheduling algorithms. To perform: Create and execute C programs for following CPU Scheduling Algorithms:

1. First Come First Serve (FCFS)

```
#include <stdio.h>


int main() {
  int n, i;
  printf("Enter number of processes: ");
  scanf("%d", &n);
  int bt[n], wt[n], tat[n];


  printf("Enter burst time for each process:\n");
  for(i = 0; i < n; i++) {
    printf("P[%d]: ", i+1);
    scanf("%d", &bt[i]);
  }


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


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


  printf("\nProcess\tBT\tWT\tTAT\n");
  for(i = 0; i < n; i++) {
    printf("P[%d]\t%d\t%d\t%d\n", i+1, bt[i], wt[i], tat[i]);
  }
  return 0;
}
```

```
Enter number of processes: 3
P[1]: 5
P[2]: 8
P[3]: 12

Process BT  WT  TAT
P[1]    5   0   5
P[2]    8   5   13
P[3]   12   13  25
```

2. Shortest Job First (SJF)

```
#include <stdio.h>
```

```
void sort(int bt[], int p[], int n) {
    for(int i=0; i<n-1; i++) {
        for(int j=i+1; j<n; j++) {
            if(bt[i] > bt[j]) {
                int temp = bt[i]; bt[i] = bt[j]; bt[j] = temp;
                temp = p[i]; p[i] = p[j]; p[j] = temp;
            }
        }
    }
}

int main() {
    int n;
    printf("Enter number of processes: ");
    scanf("%d", &n);
    int bt[n], p[n], wt[n], tat[n];

    for(int i=0; i<n; i++) {
        printf("Enter burst time for P[%d]: ", i+1);
        scanf("%d", &bt[i]);
        p[i] = i+1;
    }
    sort(bt, p, n);

    wt[0] = 0;

    for(int i=1; i<n; i++) {

        wt[i] = wt[i-1] + bt[i-1];

    }


    for(int i=0; i<n; i++) {

        tat[i] = wt[i] + bt[i];

    }


    printf("\nProcess\tBT\tWT\tTAT\n");

    for(int i=0; i<n; i++) {

        printf("P[%d]\t%d\t%d\t%d\n", p[i], bt[i], wt[i], tat[i]);

    }

    return 0;

}
```

```
Enter number of processes: 3

P[1]: 7

P[2]: 4

P[3]: 2


Process BT  WT  TAT

P[3]    2   0   2

P[2]    4   2   6
```

```
P[1]    7   6   13
```

3. Round Robin Scheduling To Submit: C Codes for the above scheduling algorithms with their outputs.

```
#include <stdio.h>


int main() {
    int i, n, tq, sq = 0;
    printf("Enter number of processes: ");
    scanf("%d", &n);
    int bt[n], rt[n], wt[n], tat[n];


    for(i = 0; i < n; i++) {
        printf("Enter burst time for P[%d]: ", i+1);
        scanf("%d", &bt[i]);
        rt[i] = bt[i];
        wt[i] = 0;
    }


    printf("Enter time quantum: ");
    scanf("%d", &tq);


    while(1) {
        int done = 1;
        for(i = 0; i < n; i++) {
```

```
        if(rt[i] > 0) {
            done = 0;
            if(rt[i] > tq) {
                rt[i] -= tq;
                sq += tq;
            } else {
                sq += rt[i];
                wt[i] = sq - bt[i];
                rt[i] = 0;
            }
        }
    }
    if(done) break;
}

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

printf("\nProcess\tBT\tWT\tTAT\n");
for(i = 0; i < n; i++) {
    printf("P[%d]\t%d\t%d\t%d\n", i+1, bt[i], wt[i], tat[i]);
}
return 0;
```

```
}
```

```
Enter number of processes: 3

P[1]: 10

P[2]: 5

P[3]: 8

Enter time quantum: 2


Process BT  WT  TAT

P[1]    10  13  23

P[2]     5   6  11

P[3]     8  10  18
```