



CPU SCHEDULING ALGORITHMS

PROGRAMMING ASSIGNMENT I

NOORUL ZUMANA SHAJAHAN 5916887



FIRST COME FIRST SERVED

```
int n,AT[],BT[],WT[];
```

```
AT = new int [n];
```

```
BT = new int[n];
```

```
WT = new int[n];
```

Arrays to store data

```
System.out.println("Enter Burst time: ");
```

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

```
    System.out.println("Enter BT for process: " + (i + 1));
```

```
    BT[i] = Integer.parseInt(is.readLine());
```

```
System.out.println("*****");
```

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

```
    System.out.println("Enter AT for process: " + (i + 1));
```

```
    AT[i] = Integer.parseInt(is.readLine());
```

```
System.out.println("*****");
```

Stores burst time and arrival time

```
WT[0] = 0;
for (int i = 1; i < n; i++){
    WT[i] = WT[i - 1] + BT[i - 1];
    WT[i] = WT[i] - AT[i];}
for (int i = 0; i < n; i++){
    AWT = AWT + WT[i] + i;}
```


Calculates waiting time

```
AWT = AWT / n;
```

Calculates Average waiting time


PREEMPTIVE SJF

Create array to store the data



```
int proc[][] = new int[n + 1][4];
```

Stores data in the array



```
for(int i = 1; i <= n; i++)  
{  
    System.out.println("Please enter the Arrival Time for Process " + i + ": ");  
    proc[i][0] = Integer.parseInt(br.readLine());  
    System.out.println("Please enter the Burst Time for Process " + i + ": ");  
    proc[i][1] = Integer.parseInt(br.readLine());  
}
```

Calculate the total amount of time

```
int total_time = 0;
for(int i = 1; i <= n; i++)
{
    total_time += proc[i][1];
}
int time_chart[] = new int[total_time];
```

```
for (int i = 0; i < total_time; i++) {
```

```
    int sel_proc = 0;
    int min = 99999;
    for (int j = 1; j <= n; j++) {
        if (proc[j][0] <= i) {
            if (proc[j][1] < min && proc[j][1] != 0) {
                min = proc[j][1];
                sel_proc = j;
            }
        }
    }
}
```

Selects the shortest process that arrived

Condition to check if process arrived

```
for (int j = 1; j <= n; j++) {  
    if (proc[j][0] <= i) {  
        if (proc[j][1] != 0) {  
            proc[j][3]++;  
            if (j != sel_proc) →  
                proc[j][2]++;  
        } else if (j == sel_proc) →  
            proc[j][3]++;  
    }  
}
```

If the process is not assigned and arrived its VWT will increase by 1

The process arrived and is completed

```
float WT = 0;  
for (int i = 1; i <= n; i++) {  
    WT += proc[i][2];  
}  
WT /= n;  
System.out.println("The Average WT is: " + WT + "ms");
```

→ Calculates the Average waiting time

PRIORITY

```
int x, n, p[], pp[], bt[], w[], t[], awt, i;  
p = new int[10];  
pp = new int[10];  
bt = new int[10];  
w = new int[10];  
t = new int[10];
```

—————→ Creates arrays

```
for (i = 0; i < n; i++) {  
    System.out.print("\nProcess[" + (i + 1) + "]:");  
    bt[i] = s.nextInt();  
    pp[i] = s.nextInt();  
    nfil = i + 1;
```

—————→ Stores data

```
for (i = 0; i < n - 1; i++) {
    for (int j = i + 1; j < n; j++) {
        if (pp[i] < pp[j]) {
            x = pp[i];
            pp[i] = pp[j];
            pp[j] = x;
            x = bt[i];
            bt[i] = bt[j];
            bt[j] = x;
            x = p[i];
            p[i] = p[j];
            p[j] = x;
        }
    }
}
```


→ Calculates total time

```
w[0] = 0;
awt = 0;
t[0] = bt[0];
for (i = 1; i < n; i++) {
    w[i] = t[i - 1];
    awt += w[i];
    t[i] = w[i] + bt[i];
}
```

→ Calculates Average
waiting time


ROUND ROBIN

Create array to store the data



```
int proc[][] = new int[n + 1][4];
```

Calculating the total time



```
int total_time = 0;
for (int i = 1; i <= n; i++) {
    total_time += proc[i][1];
}
int time_chart[] = new int[total_time];
int sel_proc = 1;
int current_q = 0;
```

```
for (int j = 1; j <= n; j++) {  
    if (proc[j][1] != 0) {  
        proc[j][3]++;  
        if (j != sel_proc)  
            proc[j][2]++;  
    } else if (j == sel_proc)  
        proc[j][3]++;  
}
```

→ Waiting time calculation

```
float WT = 0;  
for (int i = 1; i <= n; i++) {  
    WT += proc[i][2];  
}  
WT /= n;  
System.out.println("The Average WT is: " + WT + "ms");
```

→ Prints Average waiting time