DATE: __/__/____
PAGE _____

Experiment - 4

Aim :- WAP to implement priority scheduling algorithm.

Theory :- Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems.

Each process is assigned a priority. Process with highest priority is executed first and so on. Processes with same priority are executed on first come first serve basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Steps for Priority Scheduling :-

1. Make the user enter the no. of processes, burst time and arrival time.
2. Make an initial calculation of waiting time and turn-around time and display the list.
3. Ask the user to choose a factor like burst time, arrival time or waiting time to assign priority.
4. Assign the priority as number starting from 1 to the chosen factor either in ascending or descending order and display an updated list.
5. If in case 2 processes have same burst time or arrival time or waiting time, then priority would be assign on the next available factor.

GOOD WRITE

Experiment – 4

Aim :- WAP to implement the Priority Scheduling.

```
#include<iostream>
using namespace std;

void display(int bt[], int ati[], int wt[], int tt[], int n){

   cout<<endl<<"Displaying the Process Schedule :-\n";
   cout<<endl<<"PS"<<"\t"<<"BT"<<"\t"<<"AT"<<"\t"<<"WT"<<"\t"<<"TT";
   int w = 0;
   for(int i = 0; i < n; i++){
      wt[i] = w;
      w += bt[i];
      tt[i] = bt[i] + wt[i];
      cout<<endl<<i+1<<"\t"<<bt[i]<<"\t"<<ati[i]<<"\t"<<wt[i]<<"\t"<<tt[i];
   }
   cout<<endl;
}

void set_priority(int keys[], int bt[], int ati[], int n){

   for(int i = 0; i < n - 1; i++){
      for(int j = 0; j < n - i - 1; j++){
         if(keys[j] < keys[j+1]){
            swap(keys[j], keys[j+1]);
            swap(bt[j], bt[j+1]);
            swap(ati[j], ati[j+1]);
         }
      }
   }
}

int main(){

   int n;
   cout<<"Enter the No. of Processes : ";
   cin>>n;
   int bt[10], ati[10], wt[10], tt[10];

   cout<<"\nEnter the Burst Time of Processes : ";
   for(int i = 0; i < n; i++){
      cin>>bt[i];
   }
```

```
    cout<<"\nEnter the Arrival Time of Processes : ";
    for(int i = 0; i < n; i++){
        cin>>ati[i];
    }

    display(bt, ati, wt, tt, n);

    cout<<"\nEnter the Factor of Priority :-";
    cout<<"\n1 for Burst Time";
    cout<<"\n2 for Arrival Time";
    cout<<"\n3 for Waiting Time";

    int opt;
    cout<<"\n\nEnter your choice : ";
    cin>>opt;
    int keys[10];

    if(opt == 1){
        for(int i = 0; i < n; i++){
            keys[i] = (100 * bt[i]) + (10 * ati[i]) + wt[i];
        }
    }
    else if(opt == 2){
        for(int i = 0; i < n; i++){
            keys[i] = (100 * ati[i]) + (10 * bt[i]) + wt[i];
        }
    }
    else if(opt == 3){
        for(int i = 0; i < n; i++){
            keys[i] = (100 * wt[i]) + (10 * bt[i]) + ati[i];
        }
    }
    else{
        cout<<"\nInvalid Input";
        return 0;
    }

    set_priority(keys, bt, ati, n);
    display(bt, ati, wt, tt, n);
    return 0;
}
```

OUTPUT :-

Enter the No. of Processes : 5

Enter the Burst Time of Processes : 5 3 2 5 4

Enter the Arrival Time of Processes : 1 2 3 4 5

Displaying the Process Schedule :-

PS	BT	AT	WT	TT
1	5	1	0	5
2	3	2	5	8
3	2	3	8	10
4	5	4	10	15
5	4	5	15	19

Enter the Factor of Priority :-
1 for Burst Time
2 for Arrival Time
3 for Waiting Time

Enter your choice : 1

Displaying the Process Schedule :-

PS	BT	AT	WT	TT
1	5	4	0	5
2	5	1	5	10
3	4	5	10	14
4	3	2	14	17
5	2	3	17	19