School of Engineering and Applied Science, Ahmedabad University

CSE 332: Operating Systems

Section 2

End-Sem: Practical

AU1940051 Shubh Dhebar

Implementation Code

Q2A:

Round Robin implementation:

```
//AU1940051 Shubh Dhebar
#include<stdio.h>


int calc_times(int proc[], int n, int arrival_time[], int burst_time[], int tq, int wait_time[], int turnaroundtime[]) {
  int temp[n];
  for(int z=0;z<n;z++){
        temp[z]=burst_time[z];
  }
  int y = n;
  int count = 0;
  int i = 0;
  int sum = 0;
  while(y>0){
        if(temp[i] <= tq && temp[i] > 0)
        {
           sum=sum + temp[i];
           temp[i] = 0;
           count=1;
        }
        else if(temp[i] > 0)
```

```
        {
                temp[i] = temp[i] - tq;
                sum = sum + tq;
        }
        if(temp[i]==0 && count==1)
        {
                y--;
                wait_time[i] = sum - arrival_time[i] - burst_time[i];
                turnaroundtime[i] = sum - arrival_time[i];
                count =0;
        }
        if(i==n-1)
        {
         i=0;
        }
        else if(arrival_time[i+1]<=sum)
        {
        i++;
        }
        else
        {
        i=0;
        }
   }

}


int avgtime(int proc[], int n, int arrival_time[], int burst_time[], int tq) {
   int wait_time[n], tat[n], total_wt = 0, total_tat = 0;
```

```
  int i;

  calc_times(proc, n, arrival_time, burst_time, tq, wait_time, tat);

  printf("Processes  Burst   WaitingTime   TurnAroundTime \n");

  for ( i=0; i<n; i++) {
    total_wt = total_wt + wait_time[i];
    total_tat = total_tat + tat[i];            // Calculate total waiting time and total turn around time
    printf(" %d\t  %d\t\t %d \t\t%d\n", proc[i], burst_time[i], wait_time[i], tat[i]);
  }
  printf("Average waiting time = %f\n", (float)total_wt / (float)n);
  printf("Average turn around time = %f\n", (float)total_tat / (float)n);
  return 0;
}
int main(){
        int proc[] = {1, 2, 3, 4, 5, 6};   //process ids
        int arrival_time[] = {0, 1, 2, 3, 4, 5}; //arrival times
        int n = sizeof proc / sizeof proc[0];
        int burst_time[] = {7, 5, 12, 8, 9, 5};   //Burst time of all processes
        int tq=6; //time quantum

        avgtime(proc, n, arrival_time, burst_time, tq);
        return 0;
}
```

Modified Round robin implementation:

```
//AU1940051 Shubh Dhebar
#include<stdio.h>
#include<math.h>
```

```
int calc_times(int proc[], int n, int arrival_time[], int burst_time[], int tq, int wait_time[], int turnaroundtime[]) {
  int temp[n];
  for(int z=0;z<n;z++){
        temp[z]=burst_time[z];
  }
  int y = n;
  int count = 0;
  int i = 0;
  int sum = 0;
  while(y>0){
        if(temp[i] <= tq && temp[i] > 0)
        {
           sum=sum + temp[i];
           temp[i] = 0;
           count=1;
        }
        else if(temp[i] > 0)
        {
                temp[i] = temp[i] - tq;
                sum = sum + tq;
        }
        if(temp[i]==0 && count==1)
        {
                y--;
                wait_time[i] = sum - arrival_time[i] - burst_time[i];
                turnaroundtime[i] = sum - arrival_time[i];
                count =0;
        }
        if(i==n-1)
```

```
        {
         i=0;
        }
        else if(arrival_time[i+1]<=sum)
        {
        i++;
        }
        else
        {
        i=0;
        }
   }

}


int avgtime(int proc[], int n, int arrival_time[], int burst_time[], int tq) {
   int wait_time[n], tat[n], total_wt = 0, total_tat = 0;
   int i;

   calc_times(proc, n, arrival_time, burst_time, tq, wait_time, tat);

   printf("Processes  Burst   WaitingTime   TurnAroundTime \n");

   for ( i=0; i<n; i++) {
      total_wt = total_wt + wait_time[i];
      total_tat = total_tat + tat[i];              // Calculate total waiting time and total turn around
time
      printf(" %d\t  %d\t\t %d \t\t%d\n", proc[i], burst_time[i], wait_time[i], tat[i]);
   }
   printf("Average waiting time = %f\n", (float)total_wt / (float)n);
```

```
   printf("Average turn around time = %f\n", (float)total_tat / (float)n);
   return 0;
}
int time_quantum(int burst_time[], int n){
        int tq=0;
        int arr[n];
        int mean=0; int sum=0;
        for(int j=0;j<n;j++){
                arr[j]=burst_time[j];
                sum=sum+burst_time[j];
        }
        mean=sum/n;
        for(int i=0;i<n;i++)//Selection sort
        {
                int pos=i;
                for(int j=i+1;j<n;j++)
                {
                   if(arr[j]<arr[pos])
                       pos=j;
                }

                int temp=arr[i];
                arr[i]=arr[pos];
                arr[pos]=temp;
        }
        int median=burst_time[(n+1)/2 - 1];
        if(mean>median){
                tq = (int)(pow((double)(mean*burst_time[n-1]),0.5));
        }
        else{
```

```
            tq = (int)(pow((double)(median*burst_time[n-1]),0.5));
        }
        return tq;

}
int main(){
        int proc[] = {1, 2, 3, 4, 5, 6};   //process ids
        int arrival_time[] = {0, 1, 2, 3, 4, 5}; //arrival times
        int n = sizeof proc / sizeof proc[0];
        int burst_time[] = {7, 5, 12, 8, 9, 5};   //Burst time of all processes

        int tq=time_quantum(burst_time, n); //time quantum
        printf("Modufied time quantum = %d\n",tq);
        avgtime(proc, n, arrival_time, burst_time, tq);
        return 0;

}
```

OUTPUT:

```
shubhdhebar@ububtu-hp:~/Documents/os/endsem_practical$ gcc rr_scheduling_2A.c
shubhdhebar@ububtu-hp:~/Documents/os/endsem_practical$ ./a.out
Processes  Burst   WaitingTime   TurnAroundTime
 1         7              28             35
 2         5              5              10
 3         12             27             39
 4         8              32             40
 5         9              33             42
 6         5              24             29
Average waiting time = 24.833334
Average turn around time = 32.500000
shubhdhebar@ububtu-hp:~/Documents/os/endsem_practical$ gcc mod_rr_scheduling_2A.c -lm
shubhdhebar@ububtu-hp:~/Documents/os/endsem_practical$ ./a.out
Modufied time quantum = 7
Processes  Burst   WaitingTime   TurnAroundTime
 1         7              0              7
 2         5              6              11
 3         12             29             41
 4         8              33             41
 5         9              33             42
 6         5              28             33
Average waiting time = 21.500000
Average turn around time = 29.166666
```