***CSE321 Lab Assignment 04***

Name: Mehnaz Ara Fazal

ID: 20301295

Section: 01

Email: [mehnaz.ara.fazal@g.bracu.ac.bd](mailto:mehnaz.ara.fazal@g.bracu.ac.bd)

**Task-1 SJF Preemptive Scheduling**

**#include <stdio.h>**

**int main(){**

**//WT = TA - BT**

**//TA = CT - AT**

**int n = 0;**

**printf("Number of Process\n");**

**scanf("%d",&n);**

**int noOfFinishedProcess = 0;**

**int bt[n];**

**int wt[n];**

**int ta[n];**

**int arr\_t[n];**

**int ct[n];**

**int remaining\_bt[n];**

**int time\_counter = 0;**

**float awt = 0;**

**float ata = 0;**

**printf("Provide all arrival time, one by one\n");**

**for(int b = 0;b<n;b++){**

**scanf("%d",&arr\_t[b]);**

**}**

**printf("Provide all burst time, one by one\n");**

**for(int k = 0;k<n;k++){**

**scanf("%d",&bt[k]);**

**remaining\_bt[k] = bt[k];**

**}**

**int total\_time = 0; //sum of all burst times**

**for (int z = 0;z<n;z++){**

**total\_time += bt[z];**

**}**

**int gantt\_chart[total\_time];**

**int \*ptr = &gantt\_chart[0];**

**while (noOfFinishedProcess != n){**

**int counter = 0;**

**for (int s = 0;s<n;s++){**

**if (arr\_t[s] <= time\_counter && remaining\_bt[s] != 0){**

**counter += 1;**

**}**

**}**

**int temp\_arr[counter]; //burst comparison array**

**int arr\_index = 0;**

**for (int m =0;m<n;m++){**

**if(arr\_t[m] <= time\_counter && remaining\_bt[m] != 0){**

**temp\_arr[arr\_index] = m;**

**arr\_index += 1;**

**}**

**}**

**int min\_index = -1;**

**if(counter > 1){**

**min\_index = temp\_arr[0];**

**for(int w = 1;w<counter;w++){**

**if(remaining\_bt[temp\_arr[w]] < remaining\_bt[min\_index]){**

**min\_index = temp\_arr[w];**

**}**

**}**

**}**

**else if(counter == 1){**

**min\_index = temp\_arr[0];**

**}**

**if (min\_index != -1){**

**remaining\_bt[min\_index] -= 1;**

**time\_counter += 1;**

**\*ptr = min\_index + 1;**

**ptr ++;**

**}**

**if(remaining\_bt[min\_index] == 0){**

**ct[min\_index] = time\_counter;**

**noOfFinishedProcess ++;**

**}**

**}**

**for (int r = 0;r<n;r++){**

**ta[r] = ct[r] - arr\_t[r];**

**wt[r] = ta[r] - bt[r];**

**}**

**printf("Gantt Chart:\t");**

**for(int v = 0;v<total\_time;v++){**

**if (v != 0){**

**if (gantt\_chart[v] != gantt\_chart[v-1]){**

**printf("P%d\t",gantt\_chart[v]);**

**}**

**}**

**else if (v == 0){**

**printf("P%d\t", gantt\_chart[v]);**

**}**

**}**

**printf("\n");**

**printf("Proc\tAT\tBT\tCT\tWT\tTAT\n");**

**for(int p = 0;p<n;p++){**

**printf("P%d\t%d\t%d\t%d\t%d\t%d\n",p+1,arr\_t[p],bt[p],ct[p],wt[p],ta[p]);**

**}**

**for(int a = 0;a<n;a++){**

**awt += wt[a];**

**ata += ta[a];**

**}**

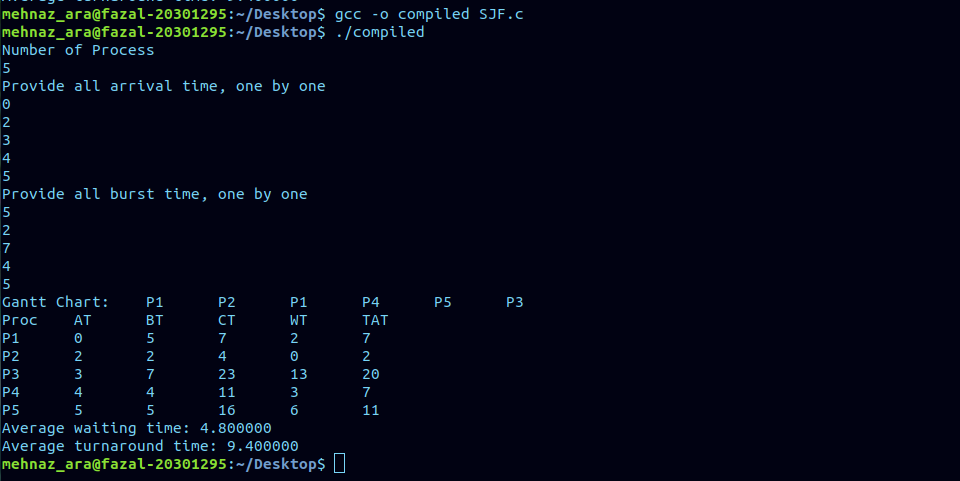
**awt = awt/n;**

**ata = ata/n;**

**printf("Average waiting time: %f\n",awt);**

**printf("Average turnaround time: %f\n",ata);**

**return 0;**

**}  
**

**Task-2 Round Robin Scheduling**

**#include <stdio.h>**

**int main(){**

**//WT = TA - BT**

**//TA = CT - AT**

**//Since arrival time not given in the question, assuming all arrival times to be 0ms and hence (Turn Around Time = Completion Time)**

**int n;**

**printf("Number of Process\n");**

**scanf("%d",&n);**

**int noOfFinishedProcess = 0;**

**int bt[n];**

**int wt[n];**

**int ta[n];**

**int remaining\_bt[n];**

**int time\_quantum = 0;**

**int time\_counter = 0;**

**int time\_counter1[n];**

**float awt = 0;**

**float ata = 0;**

**int gantt\_chart[n\*5];**

**int \*ptr = &gantt\_chart[0];**

**for (int l = 0;l<(n\*5);l++){**

**gantt\_chart[l] = 0;**

**}**

**printf("Provide time quantum value\n");**

**scanf("%d",&time\_quantum);**

**printf("Provide all burst time, one by one\n");**

**for(int k=0;k<n;k++){**

**scanf("%d",&bt[k]);**

**remaining\_bt[k] = bt[k];**

**}**

**while(noOfFinishedProcess != n){**

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

**if (remaining\_bt[i] != 0){**

**if (remaining\_bt[i] > time\_quantum){**

**\*ptr = (i+1);**

**ptr ++;**

**remaining\_bt[i] = remaining\_bt[i] - time\_quantum;**

**time\_counter += time\_quantum;**

**}**

**else if (remaining\_bt[i] < time\_quantum){**

**\*ptr = (i+1);**

**ptr ++;**

**time\_counter += remaining\_bt[i];**

**noOfFinishedProcess += 1;**

**time\_counter1[i] = time\_counter;**

**remaining\_bt[i] = 0;**

**ta[i] = time\_counter1[i];**

**}**

**else if (remaining\_bt[i] == time\_quantum){**

**\*ptr = (i+1);**

**ptr ++;**

**time\_counter += time\_quantum;**

**noOfFinishedProcess += 1;**

**time\_counter1[i] = time\_counter;**

**remaining\_bt[i] = 0;**

**ta[i] = time\_counter1[i];**

**}**

**}**

**}**

**}**

**for (int j = 0;j<n;j++){**

**wt[j] = ta[j] - bt[j];**

**}**

**printf("Gantt Chart:\t");**

**int temp = 0;**

**while(gantt\_chart[temp] != 0){**

**printf("P%d\t",gantt\_chart[temp]);**

**temp ++;**

**}**

**printf("\n");**

**printf("Proc\tAT\tBT\tCT\tWT\tTAT\n");**

**for(int m = 0;m<n;m++){**

**printf("P%d\t%d\t%d\t%d\t%d\t%d\n",m+1,0,bt[m],time\_counter1[m],wt[m],ta[m]);**

**}**

**for (int p = 0;p<n;p++){**

**awt += wt[p];**

**ata += ta[p];**

**}**

**awt = awt/n;**

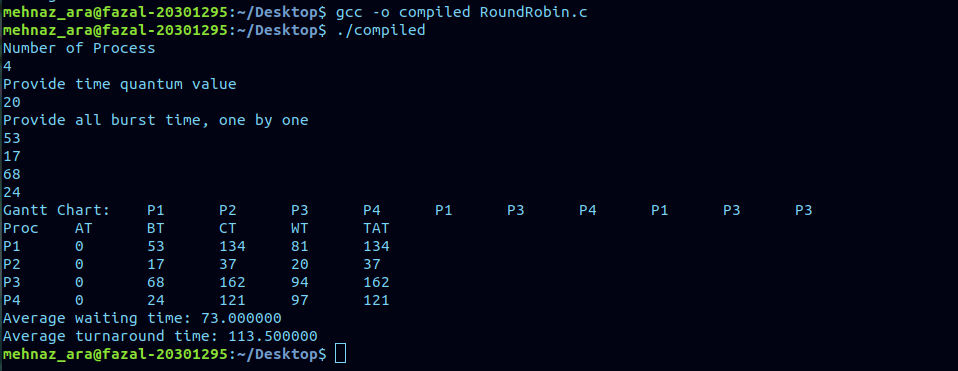
**ata = ata/n;**

**printf("Average waiting time: %f\n",awt);**

**printf("Average turnaround time: %f\n",ata);**

**return 0;**

**}**

****

**Task-3 Priority Scheduling  
#include <stdio.h>**

**int main(){**

**//WT = TA - BT**

**//TA = CT - AT**

**int n = 0;**

**printf("Number of Process\n");**

**scanf("%d",&n);**

**int noOfFinishedProcess = 0;**

**int bt[n];**

**int wt[n];**

**int ta[n];**

**int arr\_t[n];**

**int ct[n];**

**int priority[n];**

**int remaining\_bt[n];**

**int time\_counter = 0;**

**float awt = 0;**

**float ata = 0;**

**printf("Provide all arrival time, one by one\n");**

**for(int b = 0;b<n;b++){**

**scanf("%d",&arr\_t[b]);**

**}**

**printf("Provide all burst time, one by one\n");**

**for(int k = 0;k<n;k++){**

**scanf("%d",&bt[k]);**

**remaining\_bt[k] = bt[k];**

**}**

**printf("Provide all priorities, one by one\n");**

**for(int u = 0;u<n;u++){**

**scanf("%d",&priority[u]);**

**}**

**int total\_time = 0; //sum of all burst times**

**for (int z = 0;z<n;z++){**

**total\_time += bt[z];**

**}**

**int gantt\_chart[total\_time];**

**int \*ptr = &gantt\_chart[0];**

**while (noOfFinishedProcess != n){**

**int counter = 0;**

**for (int s = 0;s<n;s++){**

**if (arr\_t[s] <= time\_counter && remaining\_bt[s] != 0){**

**counter += 1;**

**}**

**}**

**int temp\_arr[counter]; //burst comparison array**

**int arr\_index = 0;**

**for (int m =0;m<n;m++){**

**if(arr\_t[m] <= time\_counter && remaining\_bt[m] != 0){**

**temp\_arr[arr\_index] = m;**

**arr\_index += 1;**

**}**

**}**

**int min\_index = -1;**

**if(counter > 1){**

**min\_index = temp\_arr[0];**

**for(int w = 1;w<counter;w++){**

**if(priority[temp\_arr[w]] < priority[min\_index]){**

**min\_index = temp\_arr[w];**

**}**

**}**

**}**

**else if(counter == 1){**

**min\_index = temp\_arr[0];**

**}**

**if (min\_index != -1){**

**remaining\_bt[min\_index] -= 1;**

**time\_counter += 1;**

**\*ptr = min\_index + 1;**

**ptr ++;**

**}**

**if(remaining\_bt[min\_index] == 0){**

**ct[min\_index] = time\_counter;**

**noOfFinishedProcess ++;**

**}**

**}**

**for (int r = 0;r<n;r++){**

**ta[r] = ct[r] - arr\_t[r];**

**wt[r] = ta[r] - bt[r];**

**}**

**printf("Gantt Chart:\t");**

**for(int v = 0;v<total\_time;v++){**

**if (v != 0){**

**if (gantt\_chart[v] != gantt\_chart[v-1]){**

**printf("P%d\t",gantt\_chart[v]);**

**}**

**}**

**else if (v == 0){**

**printf("P%d\t", gantt\_chart[v]);**

**}**

**}**

**printf("\n");**

**printf("Proc\tAT\tBT\tCT\tWT\tTAT\n");**

**for(int p = 0;p<n;p++){**

**printf("P%d\t%d\t%d\t%d\t%d\t%d\n",p+1,arr\_t[p],bt[p],ct[p],wt[p],ta[p]);**

**}**

**for(int a = 0;a<n;a++){**

**awt += wt[a];**

**ata += ta[a];**

**}**

**awt = awt/n;**

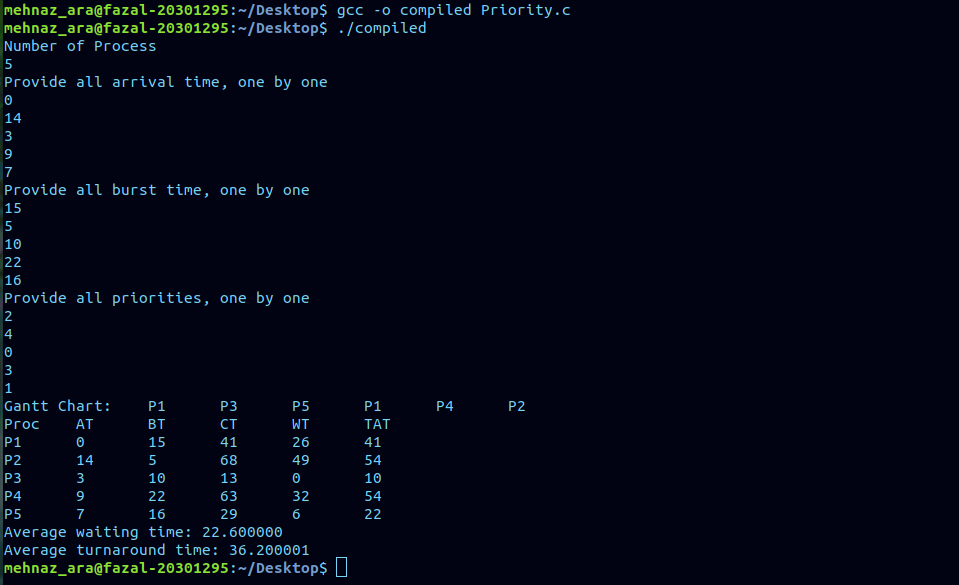
**ata = ata/n;**

**printf("Average waiting time: %f\n",awt);**

**printf("Average turnaround time: %f\n",ata);**

**return 0;**

**}**

****