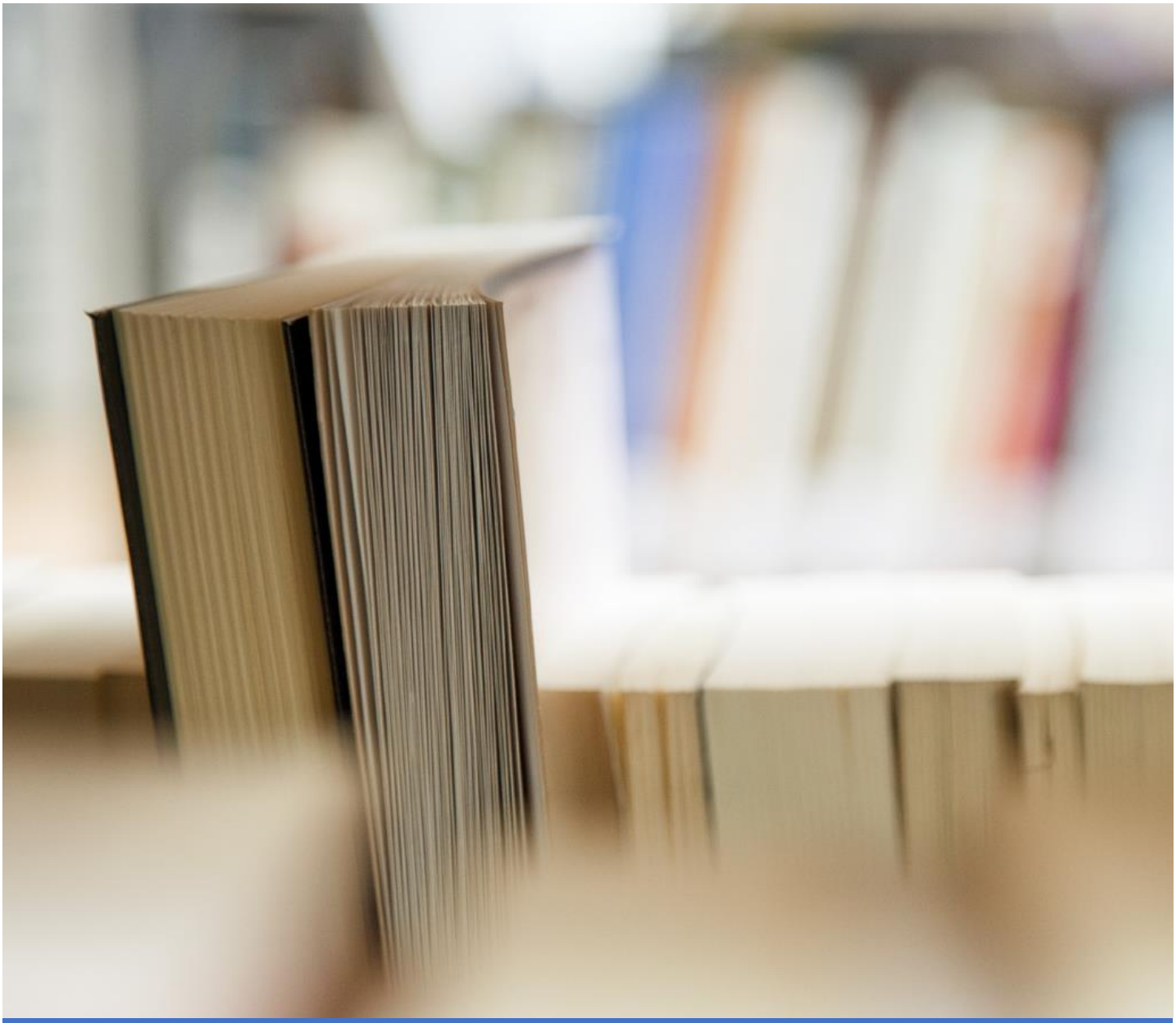


Operation System Lab

SCHEDULING

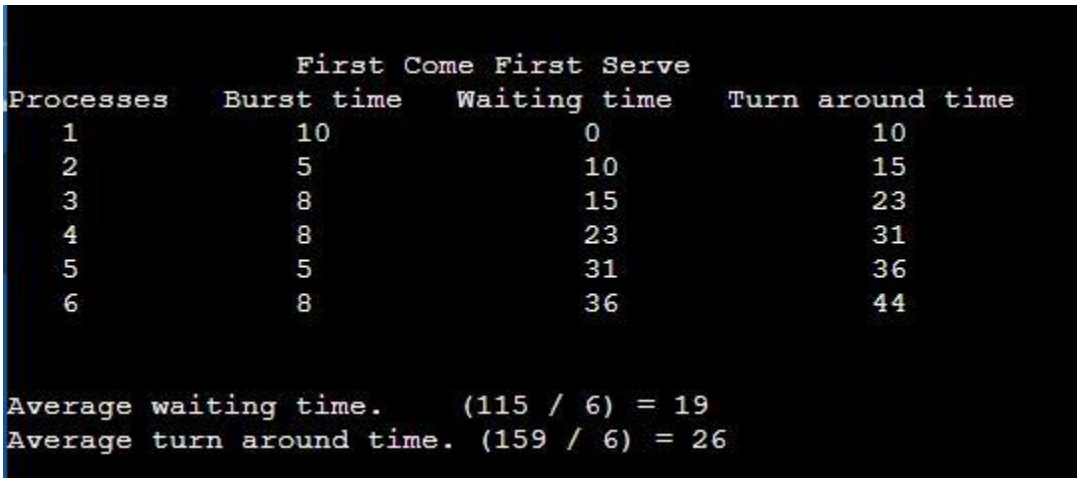
Muhammad Fahad
FA19-BSSE

First Come First Serve:

Code:

```
#include<stdio.h>
void findWaitingTime(int processes[], int n, bt[], int wt[]){
    wt[0] = 0;
    for (int  i = 1; i < n ; i++ )
        wt[i] =  bt[i-1] + wt[i-1] ;
}

void findTurnAroundTime( int processes[], int n, int bt[], int wt[], int tat[]){
    for (int  i = 0; i < n ; i++)
        tat[i] = bt[i] + wt[i];
}

void findavgTime( int processes[], int n, int bt[]){
    int wt[n], tat[n], total_wt = 0, total_tat = 0;

    findWaitingTime(processes, n, bt, wt);
    findTurnAroundTime(processes, n, bt, wt, tat);

    printf("Processes   Burst time   Waiting time   Turn around time\n");
    for (int  i=0; i<n; i++){
        total_wt = total_wt + wt[i];
        total_tat = total_tat + tat[i];

        printf("   %d\t\t%d\t\t%d\t\t%d\n",(i+1), bt[i] ,wt[i] ,tat[i] );
    }
    int s=(float)total_wt / (float)n;
    int t=(float)total_tat / (float)n;

    printf("\n\nAverage waiting time. \t (%d / %d) = %d",total_wt, n, s);
    printf("\nAverage turn around time. (%d / %d) = %d",total_tat, n,t);
}

int main(){
    int processes[] = { 1, 2, 3,8,5,8};
    int  burst_time[] = {10, 5, 8,8,5,8};

    int n = sizeof processes / sizeof processes[0];

    findavgTime(processes, n,  burst_time);
    return 0;
}
```

Output:



```
                First Come First Serve
Processes   Burst time   Waiting time   Turn around time
   1            10              0              10
   2            5               10             15
   3            8               15             23
   4            8               23             31
   5            5               31             36
   6            8               36             44


Average waiting time.    (115 / 6) = 19
Average turn around time. (159 / 6) = 26
```

Muhammad Fahad
FA19-BSSE

Shortest Job First:

Code:

```c
#include <stdio.h>

void main()
{


  printf("\n\n\t\tShortest Job First\n");

  int bt[] = {10, 5, 8, 7, 2, 1},
      p[] = {1, 2, 3, 4, 5, 6},
      n = 6,
      wt[n], tat[n], total = 0,
      // loop varaibles
      i, j, pos, temp;

  float avg_wt, avg_tat;

  n = 6;

  //sorting burst time in ascending order using selection sort
  for (i = 0; i < n; i++)
  {
    pos = i;
    for (j = i + 1; j < n; j++)
    {
      if (bt[j] < bt[pos])
        pos = j;
    }

    temp = bt[i];
    bt[i] = bt[pos];
    bt[pos] = temp;

    temp = p[i];
    p[i] = p[pos];
    p[pos] = temp;
  }

  wt[0] = 0;
```

```
  //calculate waiting time
  for (i = 1; i < n; i++)
  {
    wt[i] = 0;
    for (j = 0; j < i; j++)
      wt[i] += bt[j];

    total += wt[i];
  }

  avg_wt = (float)total / n;
  int total_wt = total;
  total = 0;

  printf("\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time");
  for (i = 0; i < n; i++)
  {
    tat[i] = bt[i] + wt[i];
    total += tat[i];
    printf("\np%d\t\t  %d\t\t    %d\t\t\t%d", p[i], bt[i], wt[i], tat[i]);
  }

  avg_tat = (float)total / n;

  printf("\n\nAverage waiting time.  \t  (%d / %d) = %f",total_wt, n, avg_wt);
  printf("\nAverage turn around time. (%d / %d) = %f",total, n,avg_tat);
}
```

Output:

Muhammad Fahad
FA19-BSSE

```
                Shortest Job First

Process     Burst Time          Waiting Time    Turnaround Time
p6                  1                   0                   1
p5                  2                   1                   3
p2                  5                   3                   8
p4                  7                   8                   15
p3                  8                   15                  23
p1                  10                  23                  33

Average waiting time.      (50 / 6) = 8.333333
Average turn around time.  (83 / 6) = 13.833333
```

Shortest Job Remaining:

Code:

```
#include <stdio.h>
struct process
{

  int WT, AT, BT, TAT;
};

struct process a[10] = {};

int main(){
  printf("\n\n\t\tShortest Job Remaining\n");

  int n = 6, temp[10];
  int count = 0, t = 0, short_P;
  float total_WT = 0, total_TAT = 0, Avg_WT, Avg_TAT;

  int bt[] = {10, 5, 8, 7, 2, 1},
      p[] = {0, 0, 3, 4, 0, 6};

  printf("\n");

  for (int i = 0; i < n; i++){
    a[i].BT = bt[i];
    a[i].AT = p[i];
    temp[i] = a[i].BT;
  }

  a[9].BT = 10000; // temp for max
  for (t = 0; count != n; t++){

    short_P = 9;

    for (int i = 0; i < n; i++){
      if (a[i].BT < a[short_P].BT && (a[i].AT <= t && a[i].BT > 0))
        short_P = i;
    }

    a[short_P].BT = a[short_P].BT - 1;
```

```
    // if any process is completed
    if (a[short_P].BT == 0){

      // one process complete
      count++;

      a[short_P].WT = t + 1 - a[short_P].AT - temp[short_P];

      a[short_P].TAT = t + 1 - a[short_P].AT;

      // total calculation
      total_WT = total_WT + a[short_P].WT;

      total_TAT = total_TAT + a[short_P].TAT;
    }
  }

  Avg_WT = total_WT / n;

  Avg_TAT = total_TAT / n;

  // printing of the answer
  printf("Id.\tAT.\tBT.\tWT.\tTAT.\n");

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

    printf(" %d\t%d\t%d\t%d\t%d\n", (i + 1), a[i].AT,
           (a[i].TAT - a[i].WT), a[i].WT, a[i].TAT);
  }
  printf("Avg waiting time of the process is %f\n", Avg_WT);

  printf("Avg turn around time of the process %f\n", Avg_TAT);
}
```

Output:

```
          Shortest Job Remaining

Id.     AT.     BT.     WT.     TAT.
 1      0       10      23      33
 2      0       5       2       7
 3      3       8       12      20
 4      4       7       4       11
 5      0       2       0       2
 6      6       1       1       2
Avg waiting time of the process is 7.000000
Avg turn around time of the process 12.500000
```

ROUND Robin [pre-emptive]:

Code:

```
      #include<stdio.h>
#include<conio.h>
#include <time.h>
#include <stdlib.h>

int getNumber(int max){
   return (rand() % (max - 1 + 1)) + 1;
}

int main(){
    srand(time(0));

    int i, NOP, sum=0,count=0, y, quant, wt=0, tat=0;

    float avg_wt, avg_tat;
    printf("Total number of process in the system: ");
    y = NOP = getNumber(8);
    printf("%d \n",y);

    int at[y], bt[y], temp[y];



    for(i=0; i<NOP; i++){
        at[i] = getNumber(3);
        bt[i] = getNumber(6);
        temp[i] = bt[i];
    }

    printf("Enter the Time Quantum for the process: \t");
    scanf("%d", &quant);

    printf("\n| Process No | Arrial Time | Burst Time |\t TAT \t|  Waiting Time |");

    for(sum=0, i = 0; y!=0; ){
        if(temp[i] <= quant && temp[i] > 0){
        sum = sum + temp[i];
        temp[i] = 0;
        count=1;
        }
        else if(temp[i] > 0){
            temp[i] = temp[i] - quant;
            sum = sum + quant;
        }
        if(temp[i]==0 && count==1){
```

```
            y--;
            printf("\n|      %d      |      %d      |      %d      |\t%d\t|\t%d\t|",
(i+1),at[i], bt[i], sum-at[i], (bt[i]-sum-at[i]));
            wt = wt+sum-at[i]-bt[i];
            tat = tat+sum-at[i];
            count =0;
        }
        if(i==NOP-1){
            i=0;
        }
        else if(at[i+1]<=sum){
            i++;
        }
        else{
            i=0;
        }
    }

    avg_wt = wt * 1.0/NOP;
    avg_tat = tat * 1.0/NOP;
    printf("\n\nAverage Turn Around Time: \t%f", avg_wt);
    printf("\nAverage Waiting Time: \t%f", avg_tat);
    getch();
}
```

Output:

Muhammad Fahad
FA19-BSSE

```
Enter your choice :  5
Enter Total Number of Process: 6


| Process No(Prority) | Burst Time |    TAT     |  Waiting Time |
|        3(1)         |      3     |    0       |       3       |
|        2(2)         |      2     |    3       |       5       |
|        1(2)         |      3     |    5       |       8       |
|        4(3)         |      2     |    8       |       10      |
|        5(3)         |      3     |    10      |       13      |
|        6(3)         |      2     |    13      |       15      |

Average Waiting Time=6
Average Turnaround Time=9
```

Muhammad Fahad
FA19-BSSE

Priority:

Code:

```
#include<stdio.h>
#include<conio.h>
#include <time.h>
#include <stdlib.h>


int Prority(){
    int i,j,n,total=0,pos,temp,avg_wt,avg_tat;

    n = getNumber(8);
    printf("Enter Total Number of Process: %d \n\n", n);

    int bt[n],p[n],wt[n],tat[n],pr[n];

    for(i=0;i<n;i++){
        bt[i] = getNumber(3);
        pr[i] = getNumber(6);
        p[i]=i+1;
    }

    for(i=0;i<n;i++){
        pos=i;
        for(j=i+1;j<n;j++){
            if(pr[j]<pr[pos])
                pos=j;
        }

        temp=pr[i];
        pr[i]=pr[pos];
        pr[pos]=temp;

        temp=bt[i];
        bt[i]=bt[pos];
        bt[pos]=temp;

        temp=p[i];
        p[i]=p[pos];
        p[pos]=temp;
    }

    wt[0]=0;

    for(i=1;i<n;i++){
        wt[i]=0;
        for(j=0;j<i;j++)
```

```
            wt[i]+=bt[j];

        total+=wt[i];
    }

    avg_wt=total/n;
    total=0;


    printf("\n| Process No(Prority) | Burst Time |\tTAT \t|  Waiting Time |");
    for(i=0;i<n;i++){
        tat[i]=bt[i]+wt[i];
        total+=tat[i];
        printf("\n|\t
%d(%d)          |       %d      |\t%d\t|\t%d\t|",p[i],pr[i],bt[i],wt[i],tat[i]);
    }

    avg_tat=total/n;     //average turnaround time
    printf("\n\nAverage Waiting Time=%d",avg_wt);
    printf("\nAverage Turnaround Time=%d\n",avg_tat);

return 0;
}
```

```
Enter your choice :  5
Enter Total Number of Process: 7


| Process No(Prority) | Burst Time |    TAT     |  Waiting Time |
|       4(1)          |      2     |    0       |       2       |
|       5(1)          |      3     |    2       |       5       |
|       1(4)          |      3     |    5       |       8       |
|       2(4)          |      1     |    8       |       9       |
|       7(5)          |      3     |    9       |       12      |
|       6(6)          |      2     |    12      |       14      |
|       3(6)          |      1     |    14      |       15      |

Average Waiting Time=7
Average Turnaround Time=9
```

Multi-Level:

Code:

```
#include<stdio.h>
#include<conio.h>
#include <time.h>
#include <stdlib.h>


int getNumberRandom(int max, int min){
   return (rand() % (max - min + 1)) + min;
}

int MM(){
  int i, k, n, temp;
  float wtavg, tatavg;
  char *temp2;

  n = getNumberRandom(10, 1);
  printf("Enter the number of processes: %d\n",n);

  int p[n],bt[n], su[n], wt[n],tat[n];

  for(i=0;i<n;i++){
    p[i] = i;
    bt[i] = getNumberRandom(10,1);
    su[i] = getNumberRandom(1, 0);
  }
  for(i=0;i<n;i++)
    for(k=i+1;k<n;k++)
      if(su[i] > su[k]){
      temp=p[i];
      p[i]=p[k];
      p[k]=temp;

            temp=bt[i];
      bt[i]=bt[k];
      bt[k]=temp;

      temp=su[i];
      su[i]=su[k];
      su[k]=temp;

      }

  wtavg = wt[0] = 0;
  tatavg = tat[0] = bt[0];
  for(i=1;i<n;i++){
```

```
    wt[i] = wt[i-1] + bt[i-1];
    tat[i] = tat[i-1] + bt[i];
    wtavg = wtavg + wt[i];
    tatavg = tatavg + tat[i];
  }

  printf("\n| PROCESS NO | SYSTEM/USER | BURST TIME |\tTAT\t|  Waiting Time |");

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

        if(su[i] == 0){
            temp2 = "  SYSTEM(0)  ";
        }else{
            temp2 = "   USER(1)   ";
        }
      printf("\n|      %d     |%s|      %d     |\t%d\t|\t%d\t|",p[i],temp2,bt[i],tat[i],wt[i]
);
    }


  printf("\nAverage Waiting Time is --- %f",wtavg/n);
  printf("\nAverage Turnaround Time is --- %f",tatavg/n);
  return 0;
}
```

```
Enter your choice :  6
Enter the number of processes: 6

| PROCESS NO | SYSTEM/USER | BURST TIME |      TAT     |  Waiting Time |
|      0     |  SYSTEM(0)  |      4     |      4       |       0       |
|      2     |  SYSTEM(0)  |      10    |      14      |       4       |
|      3     |  SYSTEM(0)  |      10    |      24      |       14      |
|      1     |   USER(1)   |      10    |      34      |       24      |
|      4     |   USER(1)   |      9     |      43      |       34      |
|      5     |   USER(1)   |      1     |      44      |       43      |
Average Waiting Time is --- 19.833334
Average Turnaround Time is --- 27.166666
```