OS INNOVATIVE ASSIGNMENT

Name and Roll No: 21BCE223 (VED PATEL) , 21BCE239(KUSH PUROHIT)

Course Code : 2CS403 OPERATING SYSTEM

MAIN FUNCTION CODE :

```c
    int main()
{
    printf("1:-For First come first serve\n2:-For Shortest job first\n3:-For
Round robin scheduling\n4:-For Priority scheduling\n5:-For Shortest Reamining
Time first\n6:-For Longest Remaining time first\n7:-To exit\n");
    FILE *f;
    f = fopen("inp.txt", "r");
    int n;
    fscanf(f, "%d", &n);
    int at[n], bt[n], temp[n], tat[n], wt[n];
    for (int i = 0; i < n; i++)
    {
        fscanf(f, "%d", &at[i]);
    }
    for (int i = 0; i < n; i++)
    {
        fscanf(f, "%d", &bt[i]);
        temp[i] = bt[i];
    }
    fclose(f);
    int x;
    while (1)
    {
        printf("Enter your choice : ");
        scanf("%d", &x);
        if (x == 1)
            fcfs(n, at, bt, temp, tat, wt);
        else if (x == 2)
            sjf(n, at, bt, temp, tat, wt);
        else if (x == 3)
            rr(n, at, bt, temp, tat, wt);
        else if (x == 4)
            priority(n, at, bt, temp, tat, wt);
        else if (x == 5)
```

```
            srtf(n, at, bt, temp, tat, wt);
        else if (x == 6)
            lrtf(n, at, bt, temp, tat, wt);
        else if (x == 7)
            break;
        else
            printf("Enter valid choice||\n");
    }
}
```

(1)   First Come First Serve (FCFS) :

```
(2) void fcfs(int n, int at[n], int bt[n], int temp[n], int tat[n], int
    wt[n])
(3) {
(4)     FILE *op;
(5)     op = fopen("outp.txt", "w");
(6)
(7)     int cnt = 0, t = 0, Total_wt = 0, Total_tat = 0;
(8)
(9)     fprintf(op, "%s", "Gantt Chart : ");
(10)
(11)         int id = -1;
(12)         int mi = 1000000000, ind = -1;
(13)         for (t; cnt < n; t++)
(14)         {
(15)             mi = 100000000;
(16)             ind = -1;
(17)
(18)             for (int j = 0; j < n; j++)
(19)             {
(20)
(21)                 if (bt[j] > 0 && at[j] <= t && at[j] < mi)
(22)                 {
(23)                     ind = j;
(24)                     mi = at[j];
(25)                 }
(26)             }
(27)             if (ind == -1 && id == -1)
(28)             {
(29)                 fprintf(op, "%s", "- ");
(30)
(31)                 continue;
(32)             }
(33)             if (ind != -1)
(34)                 id = ind;
```

```
(35)              bt[id] -= 1;
(36)              fprintf(op, "%d ", id+1);
(37)
(38)              if (bt[id] == 0)
(39)              {
(40)                  cnt++;
(41)
(42)                  tat[id] = t + 1 - at[id];
(43)                  wt[id] = tat[id] - temp[id];
(44)
(45)                  Total_tat += tat[id];
(46)                  Total_wt += wt[id];
(47)                  id = -1;
(48)              }
(49)          }
(50)
(51)          float avg_tat = (float)Total_tat / n;
(52)          float avg_wt = (float)Total_wt / n;
(53)          fprintf(op, "%s", "\n");
(54)          fprintf(op, "%s %f %s", "Average Turn-Around Time : ",
    avg_tat, "\n");
(55)          fprintf(op, "%s %f %s", "Average Waiting Time : ", avg_wt,
    "\n");
(56)
(57)          fclose(op);
(58)      }
```

OUTPUT :

```
1    Gantt Chart : 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 4 4 4 4
2    Average Turn-Around Time :   14.250000
3    Average Waiting Time :   8.750000
```

(2) Shortest Job First (SJF) :

```c
void sjf(int n, int at[n], int bt[n], int temp[n], int tat[n], int wt[n])
{
    FILE *op;
    op = fopen("outp.txt", "w");

    int cnt = 0, t = 0, Total_wt = 0, Total_tat = 0;

    fprintf(op, "%s", "Gantt Chart : ");

    int id = -1;
    int mx = 0, ind = -1;
    for (t; cnt < n; t++)
    {
        if (id != -1)
            mx = bt[id];
        else
            mx = 100000000;

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

            if (bt[j] > 0 && at[j] <= t && bt[j] < mx)
            {
                ind = j;
                mx = bt[j];
            }
        }
        if (ind == -1 && id == -1)
        {
            fprintf(op, "%s", "- ");

            continue;
        }
        if (ind != -1)
            id = ind;

        while (bt[id] > 0)
        {
            bt[id] -= 1;
            fprintf(op, "%d ", id+1);
            t++;
```

```
        }
        t--;

        if (bt[id] == 0)
        {

            cnt++;

            tat[id] = t + 1 - at[id];
            wt[id] = tat[id] - temp[id];

            Total_tat += tat[id];
            Total_wt += wt[id];
            id = -1;
        }
    }

    float avg_tat = (float)Total_tat / n;
    float avg_wt = (float)Total_wt / n;
    fprintf(op, "%s", "\n");
    fprintf(op, "%s %f %s", "Average Turn-Around Time : ", avg_tat, "\n");
    fprintf(op, "%s %f %s", "Average Waiting Time : ", avg_wt, "\n");

    fclose(op);
}
```

OUTPUT :

```
C: > Users > VED PATEL > Downloads >  ≡ inp.txt
    1    4
    2    0 1 3 5
    3    10 6 2 4
```

```
○ PS C:\C++ Conquer> cd "c:\Users\VED PATEL\Downloads\" ; if ($?) { gcc Ass
  ment } ; if ($?) { .\Assignment }
  1:-For First come first serve
  2:-For Shortest job first
  3:-For Round robin scheduling
  4:-For Priority scheduling
  5:-For Shortest Reamining Time first6:-For Longest Remaining time first
  7:-To exit
  Enter your choice : 2
  Enter your choice : ▯
```

```
C: > Users > VED PATEL > Downloads >  ≡ outp.txt
    1    Gantt Chart : 1 1 1 1 1 1 1 1 1 1 1 1 3 3 4 4 4 4 2 2 2 2 2 2
    2    Average Turn-Around Time :  12.750000
    3    Average Waiting Time :  7.250000
```

(3) Round Robin (RR) :

```c
void rr(int n, int at[n], int bt[n], int temp[n], int tat[n], int wt[n])
{

    int cnt = 0, t = 0, Total_wt = 0, Total_tat = 0, tc = 1;
    FILE *op, *fp;
    op = fopen("outp.txt", "w");

    fp = fopen("inp.txt", "r");

    fscanf(fp, "%d", &n);

    for (int i = 0; i < n; i++)
    {
        fscanf(fp, "%d", &at[i]);
    }
    for (int i = 0; i < n; i++)
    {
        fscanf(fp, "%d", &bt[i]);
        temp[i] = bt[i];
    }
    fscanf(fp, "%d", &tc);
    int f = -1, r = -1, ind = 0;
    int qu[n], ms[n];
    for (int i = 0; i < n; i++)
        qu[i] = i;
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (at[qu[j]] < at[qu[i]])
            {
                int tmp = qu[i];
                qu[i] = qu[j];
                qu[j] = tmp;
            }
        }
    }

    fprintf(op, "%s", "Gantt Chart : ");
    int id = -1;
    int mi = 1000000000;
```

```c
for(int i=0;i<n;i++)printf("%d\t",at[i]);
for (t; cnt < n; t++)
{
    while (ind < n && t >= at[qu[ind]])
    {

        if (f == -1)
            f = r = 0;
        else
            r = (r + 1) % n;
        ms[r] = qu[ind];
        ind++;

    }
    if (f == -1)
    {
        fprintf(op, "%s", "- ");

        continue;
    }
    id = ms[f];
    if (f == r)
        f = r = -1;
    else
        f = (f + 1) % n;
    int x = 0;
    if (tc > bt[id])
        x = bt[id];
    else
        x = tc;
    while (x--)
    {

        bt[id] -= 1;
        fprintf(op, "%d ", id+1);
        if ( ind<n && t >= at[qu[ind]])
        {
            if (f == -1)
                f = r = 0;
            else
                r = (r + 1) % n;
            ms[r] = qu[ind];
            ind++;

        }
        t++;

    }
    t--;

    if (bt[id] == 0)
```

```c
        {
            cnt++;

            tat[id] = t + 1 - at[id];
            wt[id] = tat[id] - temp[id];

            Total_tat += tat[id];
            Total_wt += wt[id];
            id = -1;
        }
        else
        {
            r = (r + 1) % n;
            ms[r] = id;
            if (f == -1)
                f = 0;
        }
    }

    float avg_tat = (float)Total_tat / n;
    float avg_wt = (float)Total_wt / n;
    fprintf(op, "%s", "\n");
    fprintf(op, "%s %f %s", "Average Turn-Around Time : ", avg_tat, "\n");
    fprintf(op, "%s %f %s", "Average Waiting Time : ", avg_wt, "\n");
    fclose(op);
    fclose(fp);
}
```

OUTPUT :

```
C: > Users > VED PATEL > Downloads >  ≡ inp.txt

  1     4
  2     0 1 3 5
  3     10 6 2 4
  4     4
```

```
o PS C:\C++ Conquer> cd "c:\Users\VED PAT
    } ; if ($?) { .\Assignment }
  1:-For First come first serve
  2:-For Shortest job first
  3:-For Round robin scheduling
  4:-For Priority scheduling
  5:-For Shortest Reamining Time first
  6:-For Longest Remaining time first
  7:-To exit
  Enter your choice : 3
  Enter your choice : []
```

```
1    Gantt Chart : 1 1 1 1 2 2 2 2 3 3 1 1 1 1 4 4 4 4 2 2 1 1
2    Average Turn-Around Time :  15.250000
3    Average Waiting Time :  9.750000
4
```

(4) Priority

```c
void priority(int n, int at[n], int bt[n], int temp[n], int tat[n], int wt[n])
{
    FILE *op, *fp;
    op = fopen("outp.txt", "w");

    fp = fopen("inp.txt", "r");
    int p[n];

    fscanf(fp, "%d", &n);

    for (int i = 0; i < n; i++)
    {
        fscanf(fp, "%d", &at[i]);
    }
    for (int i = 0; i < n; i++)
    {
        fscanf(fp, "%d", &bt[i]);
        temp[i] = bt[i];
    }
    for (int i = 0; i < n; i++)
    {
        fscanf(fp, "%d", &p[i]);
    }

    int cnt = 0, t = 0, Total_wt = 0, Total_tat = 0;

    fprintf(op, "%s", "Gantt Chart : ");

    int id = -1;
    int mi = 1000000000, ind = -1;
    for (t; cnt < n; t++)
    {
        if (id != -1)
            mi = p[id];
```

```c
        else
            mi = 100000000;

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

            if (bt[j] > 0 && at[j] <= t && p[j] < mi)
            {
                ind = j;
                mi = p[j];
            }
        }
        if (ind == -1 && id == -1)
        {
            fprintf(op, "%s", "- ");

            continue;
        }
        if (ind != -1)
            id = ind;

        bt[id] -= 1;
        fprintf(op, "%d ", id+1);

        if (bt[id] == 0)
        {
            cnt++;

            tat[id] = t + 1 - at[id];
            wt[id] = tat[id] - temp[id];

            Total_tat += tat[id];
            Total_wt += wt[id];
            id = -1;
        }
    }

    float avg_tat = (float)Total_tat / n;
    float avg_wt = (float)Total_wt / n;
    fprintf(op, "%s", "\n");
    fprintf(op, "%s %f %s", "Average Turn-Around Time : ", avg_tat, "\n");
    fprintf(op, "%s %f %s", "Average Waiting Time : ", avg_wt, "\n");
    fclose(op);
    fclose(fp);
}
```

OUTPUT :

```
1    4
2    0 1 3 5
3    10 6 2 4
4    2 4 5 1
```

```
ent } ; if ($?) { .\Assignment }
1:-For First come first serve
2:-For Shortest job first
3:-For Round robin scheduling
4:-For Priority scheduling
5:-For Shortest Reamining Time first
6:-For Longest Remaining time first
7:-To exit
Enter your choice : 4
Enter your choice : ▯
```

```
1    Gantt Chart : 1 1 1 1 1 1 4 4 4 4 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3
2    Average Turn-Around Time :  14.000000
3    Average Waiting Time :  8.500000
4
```

(5) Shortest Job Remaining First

```c
void srtf(int n, int at[n], int bt[n], int temp[n], int tat[n], int wt[n])
{
    FILE *op;
    op = fopen("outp.txt", "w");

    int cnt = 0, t = 0, Total_wt = 0, Total_tat = 0;

    fprintf(op, "%s", "Gantt Chart : ");

    int id = -1;
    int mi = 1000000000, ind = -1;
    for (t; cnt < n; t++)
    {
        if (id != -1)
            mi = bt[id];
        else
            mi = 100000000;

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

            if (bt[j] > 0 && at[j] <= t && bt[j] < mi)
            {
```

```c
                ind = j;
                mi = bt[j];
            }
        }
        if (ind == -1 && id == -1)
        {
            fprintf(op, "%s", "- ");

            continue;
        }
        if (ind != -1)
            id = ind;

        bt[id] -= 1;
        fprintf(op, "%d ", id+1);

        if (bt[id] == 0)
        {
            cnt++;

            tat[id] = t + 1 - at[id];
            wt[id] = tat[id] - temp[id];

            Total_tat += tat[id];
            Total_wt += wt[id];
            id = -1;
        }
    }

    float avg_tat = (float)Total_tat / n;
    float avg_wt = (float)Total_wt / n;
    fprintf(op, "%s", "\n");
    fprintf(op, "%s %f %s", "Average Turn-Around Time : ", avg_tat, "\n");
    fprintf(op, "%s %f %s", "Average Waiting Time : ", avg_wt, "\n");

    fclose(op);
}
```

OUTPUT :

```
1    4
2    0 1 3 5
3    10 6 2 4
```

```
1    Gantt Chart : 1 2 2 3 3 2 2 2 2 4 4 4 4 1 1 1 1 1 1 1 1 1 1
2    Average Turn-Around Time :  10.000000
3    Average Waiting Time :  4.500000
4
```

(6) Longest Remaining Time First (LRTF) :

```c
void lrtf(int n, int at[n], int bt[n], int temp[n], int tat[n], int wt[n])
{
    FILE *op;
    op = fopen("outp.txt", "w");

    int cnt = 0, t = 0, Total_wt = 0, Total_tat = 0;

    fprintf(op, "%s", "Gantt Chart : ");

    int id = -1;
    int mx = 0, ind = -1;
    for (t; cnt < n; t++)
    {
        if (id != -1)
            mx = bt[id];
        else
            mx = 0;

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

            if (bt[j] > 0 && at[j] <= t && bt[j] > mx)
            {
                ind = j;
                mx = bt[j];
            }
        }
        if (ind == -1 && id == -1)
        {
```

```c
            fprintf(op, "%s", "- ");

            continue;
        }
        if (ind != -1)
            id = ind;

        bt[id] -= 1;
        fprintf(op, "%d ", id+1);

        if (bt[id] == 0)
        {
            cnt++;

            tat[id] = t + 1 - at[id];
            wt[id] = tat[id] - temp[id];

            Total_tat += tat[id];
            Total_wt += wt[id];
            id = -1;
        }
    }

    float avg_tat = (float)Total_tat / n;
    float avg_wt = (float)Total_wt / n;
    fprintf(op, "%s", "\n");
    fprintf(op, "%s %f %s", "Average Turn-Around Time : ", avg_tat, "\n");
    fprintf(op, "%s %f %s", "Average Waiting Time : ", avg_wt, "\n");

    fclose(op);
}
```

OUTPUT :

```
1    Gantt Chart : 1 1 1 1 1 2 2 1 1 2 4 4 1 2 2 1 3 4 4 1 2 3
2    Average Turn-Around Time :  18.250000
3    Average Waiting Time :  12.750000
4
```