

Lab - 4

12/7/23

write a C program to simulate multi level queue scheduling algorithm considering the following scenario. All the process in the system are divided into two categories - system process and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.

```
#include <stdio.h>
```

```
int spat[10], upat[10], i, n1, n2, p1[10], p2[10];  
int sppt[10], uppt[10], time = 0, op = 0, y, z, pt;  
int sptat[10], uptat[10];  
int spwt[10], upwt[10];
```

```
float spatat = 0, spwt = 0;
```

```
float upatat = 0, upwt = 0;
```

```
void process (int x, int isSystem) {  
    if (isSystem) {
```

```
        op = sppt[x];
```

```
        sptat[x] = op - spat[x];
```

```
        sppt[x] = 0;
```

```
        spwt[x] = sptat[x] - p1[x];
```

```
        spatat += spat[x];
```

```
        spawt += spwt[x];
```

```
    } else {
```

```
        op = uppt[x];
```

```
        uptat[x] = op - upat[x];
```

```
        uppt[x] = 0;
```

```
        upwt[x] = uptat[x] - p2[x];
```

```
        upatat += upat[x];
```

```
        upawt += upwt[x];
```

```
    }
```

```

int main() {
    printf ("Enter the no of System Processes: ");
    scanf ("%d", &n1);

    printf ("Enter the no of User Processes: ");
    scanf ("%d", &n2);

    printf ("Enter the arrival times for System Process:");
    for (i = 0; i < n1; i++)
        scanf ("%d", &spat[i]);

    printf ("Enter the process times for System Process:");
    for (i = 0; i < n1; i++)
        scanf ("%d", &sppt[i]);

    printf ("Enter the arrival times for User Process:");
    for (i = 0; i < n2; i++)
        scanf ("%d", &upat[i]);

    printf ("Enter the process time for System Process:");
    for (i = 0; i < n2; i++)
        scanf ("%d", &uppt[i]);

    for (i = 0; i < n1; i++)
        time += sppt[i];

    for (i = 0; i < n2; i++)
        time += uppt[i];

    for (i = 0; i < n1; i++)
        pt[i] = spat[i];

    for (i = 0; i < n2; i++)
        p2[i] = uppt[i];

    printf ("\n");
}

```

```
while (op < time) {
```

```
    y = -1;
```

```
    z = -1;
```

```
    for (i = 0; i < n1; i++) {
```

```
        if (op >= spot[i] && spot[i] != 0) {
```

```
            y = i;
```

```
            break;
```

```
    }
```

```
    for (i = 0; i < n2; i++) {
```

```
        if (op >= upat[i] && upat[i] != 0) {
```

```
            z = i;
```

```
            break;
```

```
    }
```

```
    if (y != -1) {
```

```
        printf (" %.d SP %.d", op, y+1);
```

```
        process (y, 1);
```

```
    } else if (z != -1) {
```

```
        printf (" %.d UP %.d", op, z+1);
```

```
        process (z, 0);
```

```
    } else {
```

```
        op++
```

```
    }
```

```
    }
```

```
    printf (" %.d", op);
```

```
    printf ("\n");
```

```
    printf ("System Processes : \n");
```

```
    for (i = 0; i < n1; i++)
```

```
        printf ("SP %.d %.d %.d \n", i+1, spot[i], upat[i]);
```



```

printf ("ATAT(SP): %.2f \n", spata / n1);
printf ("AWT(SP): %.2f \n", spawt / n1);
printf ("\n");
printf ("User Process : \n");
for (i = 0; i < n2; i++)
    printf ("UP %d %d %d \n", i+1, upat[i], upwt[i]);
printf ("ATAT(U.P) : %.2f \n", upodat / n2);
printf ("AWT(U.P) : %.2f \n", upawt / n2);
return 0;
}

```

Output :

Enter the number of system process : 5

Enter the number of user process : 5

Enter ~~the~~ AT of SP :

0 1 1

Enter the PT of SP :

5 3 4

Enter the A.T of O.P :

1 3 4

Enter the P.T of UP :

5 3 2

```

int util[n];
float utilization = 0;
for (i = 0; i < n; i++) {
    util[i] = hyp / per + [i];
    util[i] = util[i];
    utilization = util[i];
    printf ("util %d %d\n", i+1, util[i]);
}
utilization = hyp;
printf ("utilization = %f\n", utilization);
return 0;
}

```

Output :

Enter no of process : 3

Enter period & processing time :

6 2

8 3

10 2

Hyper period = 120

util 1 : 40

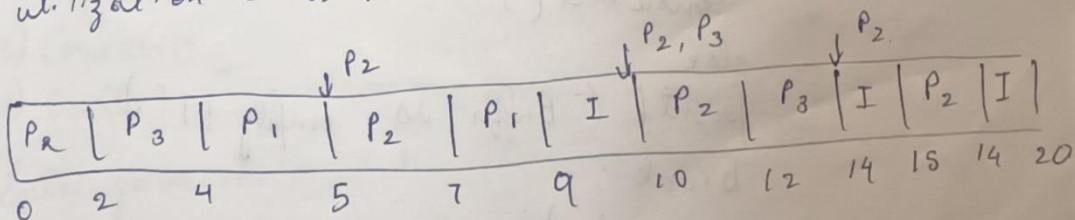
util 2 : 45

util 3 : 20

utilization = 0.908333

8/10

19/7/23



## OUTPUT:

```
F:\OS\multiQLab.exe
Enter the number of system processes: 3
Enter the Arrival time and the Burst time for system processes:
0 2
1 3
8 5
Enter the number of user processes: 3
Enter the Arrival time and the Burst time for user processes:
0 2
0 3
2 4
```

PROCESS	ARRIVAL TIME	BURST TIME	WAITING TIME	TURNAROUND TIME
S0	0	2	0	2
S1	1	3	1	4
S2	8	5	0	5
U0	0	2	5	7
U1	0	3	12	15
U2	2	4	13	17

```
Average Turnaround Time -- 8.333333
Average Waiting Time -- 5.166667
Process returned 33 (0x21)   execution time : 21.961 s
Press any key to continue.
```

Rate monotonic

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int gcd(int a, int b){  
    if (b == 0) return a;  
    else gcd(b, a % b);  
}
```

```
{  
    int lcm(int a, int b){  
        return (a * b) / gcd(a, b);  
    }
```

```
int hyperperiod(int period[], int n){  
    int k = period[0];  
    n--;
```

```
    while (n >= 1){
```

```
        k = lcm(k, period[n-1]);
```

```
    }  
    return k;
```

```
{  
    int main(){
```

```
        int period[10], cpul[10], n, i, lenb, hyp;
```

```
        printf("Enter no of process: \n");
```

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

```
        printf("Enter period & processing time: \n");
```

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

```
            scanf("%d %d", &period[i], &cpul[i]);
```

```
        }
```

```
        hyp = hyperperiod(period, n);
```

```
        printf("Hyper period = %d \n", hyp);
```



```

int util[n];
float utilization = 0;
for (i = 0; i < n; i++) {
    util[i] = hyp / per + ED;
    util[i] = util[i];
    utilization = util[i];
    printf ("util %d %d\n", i+1, util[i]);
}
utilization = hyp;
printf ("utilization = %f\n", utilization);
return 0;
}

```

Output :

Enter no of process : 3

Enter period & processing time :

6 2

8 3

10 2

Hyper period = 120

util 1 : 40

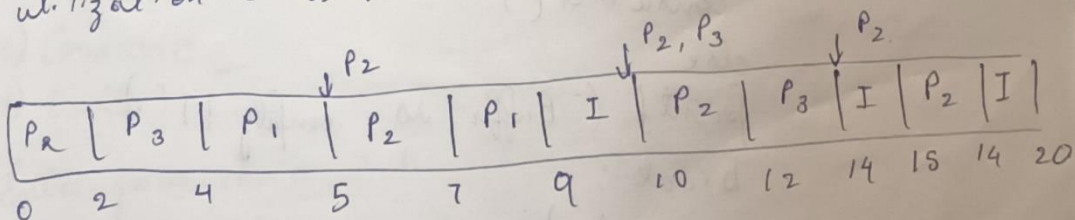
util 2 : 45

util 3 : 20

utilization = 0.908333

8/10

19/7/23





## OUTPUT:

```
F:\OS\rateLAB.exe
1.Rate monotonic
2.exit

1
Enter the number of tasks: 3
Task 1
Enter period: 20
Enter execution time: 3
Enter deadline: 20
Task 2
Enter period: 5
Enter execution time: 2
Enter deadline: 5
Task 3
Enter period: 10
Enter execution time: 2
Enter deadline: 10
CPU Utilization: 75.0000%
```













