WEEK 3

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

CODE:

```
#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, spawt = 0;
float upatat = 0, upawt = 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 += sptat[x];
     spawt += spwt[x];
  } else {
     op += uppt[x];
     uptat[x] = op - upat[x];
     uppt[x] = 0;
     upwt[x] = uptat[x] - p2[x];
     upatat += uptat[x];
     upawt += upwt[x];
  }
}
int main() {
  printf("Enter the number of System Processes: ");
  scanf("%d", &n1);
```

```
printf("Enter the number of User Processes: ");
scanf("%d", &n2);
printf("Enter the arrival times for System Processes:\n");
for (i = 0; i < n1; i++)
  scanf("%d", &spat[i]);
printf("Enter the process times for System Processes:\n");
for (i = 0; i < n1; i++)
  scanf("%d", &sppt[i]);
printf("Enter the arrival times for User Processes:\n");
for (i = 0; i < n2; i++)
  scanf("%d", &upat[i]);
printf("Enter the process times for User Processes:\n");
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++)
  p1[i] = sppt[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 >= spat[i] && sppt[i] != 0) {
```

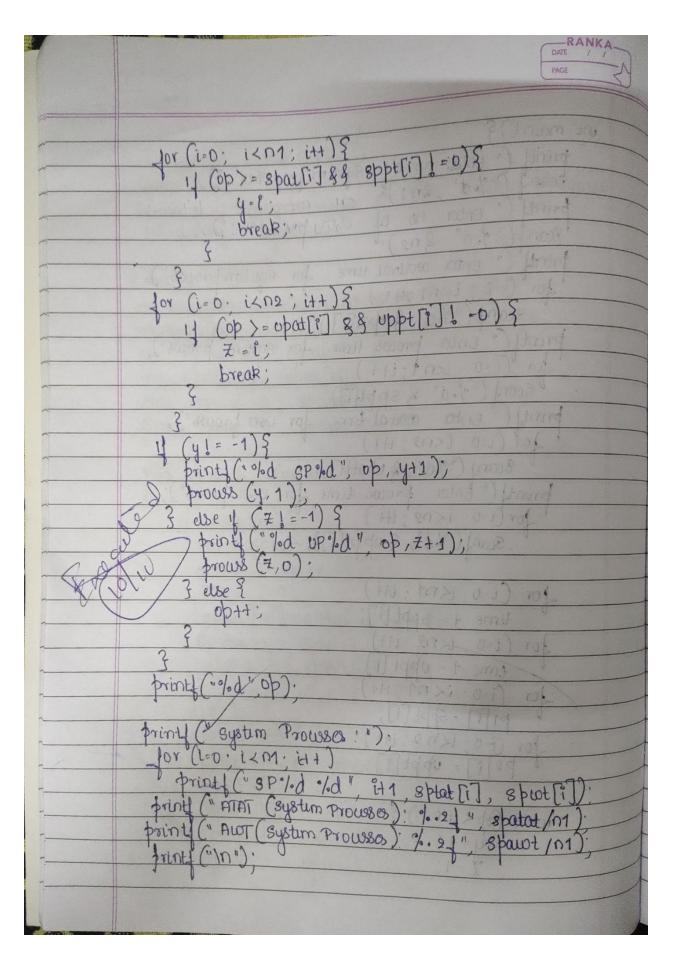
```
y = i;
        break;
     }
  }
  for (i = 0; i < n2; i++) {
     if (op >= upat[i] && uppt[i] != 0) {
        z = i;
        break;
     }
  }
  if (y != -1) {
     printf("%d SP%d ", op, y + 1);
     process(y, 1);
  } else if (z != -1) {
     printf("%d UP%d ", op, z + 1);
     process(z, 0);
  } 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, sptat[i],spwt[i]);
printf("ATAT(System Processes): %.2f\n", spatat / n1);
printf("AWT(System Processes): %.2f\n", spawt/n1);
printf("User Processes:\n");
for (i = 0; i < n2; i++)
  printf("UP%d %d %d\n", i + 1, uptat[i], upwt[i]);
printf("ATAT(User Processes): %.2f\n", upatat / n2);
printf("AWT(User Processes): %.2f\n", upawt / n2);
return 0;
```

}

OBSERVATION:

```
12/7/23
          Write a cprogram to semulate multi-livel quie
                       algorithm.
          schiduling
          # indude < stdio.h>
          int spat [10], upat [10], i, n1, n2, p1 [10], P2 [10];
          int splot[10], upt[10], time=0, op=0, y, Z, pt;
           int 8 ptat [10], uptat[10];
           int sput [io], uput[io];
           float spotat = 0; spouot = 0;
           foat upatat = 0, uparot = 0;
            Void prows (intx, ent issystem)
               11 (issystem) }
                  opt = sppt[2];
                  sptat[x] = op - spat [x];
                  8ppt [x] =0
                   8 ptot [x] = 8 ptot[x] - 1/x
                  spatal + = sptat[x]
                  spout += spoutfi
                 else &
                   opt = uppt[a];
                   uptat[x] = op - upatfa
                   up pt [al] = 0',
                   up wt [2] = up tat [2] - |22 [2].
                   upatat + = uptat [2]
                   upant + = upwtri
```

```
int main () ?
  scan ("%d', 8n1)
   scan ("%d", gn2).
  printly (" Enter overlyal time for Gystum process");
for (i=0; i<n1; i+1)
    scanf ("0/0d", & spat[i]);
   print (" Ento proces time for system proces");
     for (i=0 . Kn1; i++
    "scant ("o/.d", & sppt[i]);
   print! (" Enter avoiral time for usor procus");
    for (i=0; i<n2; i+1)
    sant ("%d", & upat[i]); " 1
    print (" Enter prows time for user prous");
      for (i=0; i<n2; i+1)
        scanf ("%d", & up pt[i]);
    for (i=0; KM; itt.
       time += sppt[i]
     or (i=0 ikn2 itt)
      time + = uppt [1]
     for (i=0 · i< n1; itt
      p1[i] = 8ppl[i];
     for Ci=o; ixn2; i+t)
       perij = uppt[i];
    while (op < time) ?
```



```
print (" usor prouses: ");
for (i.o; 1<n2; i++) d
print ("ATAT (usor Prouses): "1.21", upatat/n2);
print ("Awt (usor Prouses): 1/021", upart/no);
  retwin 0;
                        A count (same n)
 OUT PUT: -
 Enter the number of bystem proces: 3
Enter the number of User proces: 1
Enter the arrival times for system processes:
Enter the process times for system processes:
4 3 5
Enter the openival times
                         or user prouses!
0
Entor the process times
                          for user processes:
 0 SP1 4 SP2 7 UP1 15 SP3 20
 Bystim prouse!
 581 4 0
SP2 7 4 3 ( +0 0 )
8P3 10 5 1000 1000
       (system process): 7.00
ATAT
 A WT ( 8ystem prowsu): 3.00
        prouses:
 USOR
        15 7
 UP1
        (USUT processes): 15.00
                                       al hai .
 ATAT
        (user processes): 7
 ALOT
```

OUTPUT:

```
■ "C<\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\OS\oslab\lab3\bin\Debug\lab3.exe"

Enter the number of System Processes: 3
Enter the number of User Processes: 1
Enter the arrival times for System Processes:
0 8 10
Enter the process times for System Processes:
1 8 Enter the process times for User Processes:
0 Enter the process times for User Processes:
0 Enter the process times for User Processes:
8 0 SP1 4 SP2 7 UP1 15 SP3 20
System Processes:
SP1 4 8 0
SP2 7 4
SP3 10 5
ATAT(System Processes): 7.00
AWT(System Processes): 3.00
User Processes:
UP1 15 7
ATAT(User Processes): 15.00
AWT(User Processes): 7.00
Process returned 0 (0x0) execution time : 51.340 s
Press any key to continue.
```