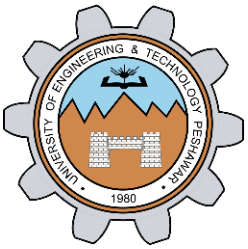
OPERATING SYSTEMS LAB 10

SCHEDULING ALGORITHM



Spring 2024

Submitted by: **Hassan Zaib**

Registration No: **22pwsce2144**

Class Section: **A**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Submitted to:

Engr. Abdullah Hamid

Month Day, Year (30 May, 2024)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Title: FIRST COME FIRST AND SHORTEST JOB FIRST ALGORITHM

Code:

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
void swap(int arr[][7],int x,int y){
int temp,i;
for (i=0;i<7-1;i++){
    temp=arr[x][i];
    arr[x][i]=arr[y][i];
    arr[y][i]=temp;
}
}
void read_job_entry(int array[][7],int num_jobs){
int i;
for (i=0;i<num_jobs;i++){
    printf("enter job no for process %d:",i+1);
    scanf("%d",&array[i][0]);
    printf("enter arrival time for process %d:",i+1);
    scanf("%d",&array[i][1]);
    printf("enter burst time for process %d:",i+1);
    scanf("%d",&array[i][2]);
}    }
     void print_sorted(int array[][7],int num_jobs){
     int i,j;
     printf("\n the sorted jobs:/n");
     printf("job no\tarrival time\tburst time\n");
```

```
printf("job no\tarrival time\tburst time\n");
for(i=0;i<num_jobs;i++){
   for(j=0;j<3;j++){
       printf("%d\t\t",array[i][j]);
}
    printf("\n");
 }
printf("\n");
}
void sort_by_burst_time(int array[][7],int num_jobs){
int i,j;
for (i=0;i<num_jobs-1;i++){
   for (j=0;j<num_jobs-i-1;j++){
       if(array[j][2]>array[j+1][2]){
           swap(array,j,j+1);
       }
   } }
}
void calculate(int array[][7],int num_jobs){
int i;
for(i=0;i<num_jobs;i++){
   if(i==0){
       array[i][3]=array[i][1];//arival is equal
   }
   else{
       array[i][3]=array[i-1][4];
   }
```

```
    }
    array[i][4]=array[i][3] + array[i][2];
    array[i][5]=array[i][3] - array[i][1];
    array[i][6]=array[i][4] - array[i][1];
}
}

void printing(int array[][7], int num_jobs){
int i;
printf("scheduling according to algo:\n");
printf("job no.\tarriaval time\tburst time\tstart time\tfinish time\twaiting time\tturnaround time\n");
for(i=0;i<num_jobs;i++){
   printf("%d\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n",array[i][0],array[i][1],array[i][2],array[i][3],array[i][4],array[i][5],

}
}
void FCFS(){
int num_jobs;
printf("enter total number of jobs:");
scanf("%d",&num_jobs);
int process[num_jobs][7];
read_job_entry(process,num_jobs);
print_sorted(process,num_jobs);
calculate(process,num_jobs);
printing(process,num_jobs);
}

void SJFS(){
int num_jobs;
printf("enter total number of jobs:");
scanf("%d",&num_jobs);
int process[num_jobs][7];
read_job_entry(process,num_jobs);
print_sorted(process,num_jobs);
calculate(process,num_jobs);
printing(process,num_jobs);
}
int main(){
printf("FCFS Scheduling \n\n");
FCFS();
printf("\n\n");
printf("SJFS Scheduling \n\n");
SJFS();
printf("\n\n");
return 0;
}
```

Output:

```
SJFS Scheduling

enter total number of jobs:2
enter job no for process 1:1
enter arrival time for process 1:2
enter burst time for process 1:3
enter job no for process 2:4
enter arrival time for process 2:5
enter burst time for process 2:6

 the sorted jobs:/njob no       arrival time    burst time
1               2               3
4               5               6

scheduling according to algo:
job no. arriaval time   burst time      start time      finish time     waiting time    turnaround time
1       2               3               2               5               0               3
4       5               6               5               11              0               6
```

```
FCFS Scheduling

enter total number of jobs:2
enter job no for process 1:1
enter arrival time for process 1:3
enter burst time for process 1:4
enter job no for process 2:2
enter arrival time for process 2:2
enter burst time for process 2:5

 the sorted jobs:/njob no       arrival time    burst time
1               3               4
2               2               5

scheduling according to algo:
job no. arriaval time   burst time      start time      finish time     waiting time    turnaround time
1       3               4               3               7               0               4
2       2               5               7               12              5               10
```

CSE 302L: Operating Systems Lab

LAB ASSESSMENT RUBRICS

Marking Criteria	Exceeds expectation (2.5)	Meets expectation (1.5)	Does not meet expectation (0)	Score
1. Correctness	Program compiles (no errors and no warnings). Program always works correctly and meets the specification(s). Completed between 81-100% of the requirements.	Program compiles (no errors and some warnings). Some details of the program specification are violated, program functions incorrectly for some inputs. Completed between 41-80% of the requirements.	Program fails to or compile with lots of warnings. Program only functions correctly in very limited cases or not at all. Completed less than 40% of the requirements.	
2. Delivery	Delivered on time, and in correct format (disk, email, hard copy etc.)	Not delivered on time, or slightly incorrect format.	Not delivered on time or not in correct format.	
3. Coding Standards	Proper indentation, whitespace, line length, wrapping, comments and references.	Missing some of whitespace, line length, wrapping, comments or references.	Poor use of whitespace, line length, wrapping, comments and references.	
4. Presentation of document	Includes name, date, and assignment title. Task titles, objectives, output screenshots included and good formatting and excellently organized.	Includes name, date, and assignment title. Task titles, objectives, output screenshots included and good formatting.	No name, date, or assignment title included. No task titles, no objectives, no output screenshots, poor formatting.	

Instructor:

Name: Engr. Abdullah Hamid Signature: _____