



**DHA Suffa University**  
**CS 206 – Operating Systems – Lab**  
**Fall 2017**  
**Lab 13 – FCFS and SJF Scheduling**



**Objective(s):**

- Implementing
  - First Come First Serve Scheduling Algorithm
  - Shortest Job First Scheduling Algorithm

Before getting started, a few things that would help you in implementing these algorithms:

- Array in Shell Scripting is used same as an array in C/C++ with some obvious exceptions:

$arr[i] = x$  in C/C++ is equivalent to  $arr[i]=x$   
 $arr[i] = arr[i] + x$  in C/C++ is equivalent to  $arr[i]=`expr ${arr[i]} + $x`$

- Current time can be recorded in Shell script using the system variable of SECONDS.

i.e. *currentTime=SECONDS*

**First Come First Serve Scheduling Algorithm:**

- Aim:  
Write a Shell script to Input processes from user and run them on FCFS basis.

**Algorithm:**

- Step 1:** Start the process
- Step 2:** Accept the number of processes in the ready Queue
- Step 3:** For each process in the ready Q, assign the process id and accept the CPU burst time
- Step 4:** Set the waiting time (WT) of the first process as '0' and its burst time as its turn-around time
- Step 5:** For each process in the Ready Q calculate  
a) WT for process(n) = WT of process(n-1) + Burst time(BT) of process(n-1)  
b) Turn-around time (TaT) of Process(n) = WT of Process(n) + BT of process(n)
- Step 6:** Calculate  
a) Average waiting time = Total waiting Time / Number of processes  
b) Average Turnaround time = Total Turnaround Time / Number of processes
- Step 7:** Stop the process

### Shortest Job First Scheduling Algorithm:

#### Aim:

Write a Shell script to Input processes from user and run them on SJF basis.

#### Algorithm:

- Step 1:** Start the process
- Step 2:** Accept the number of processes in the ready Queue
- Step 3:** For each process in the ready Q, assign the process id and accept the CPU burst time
- Step 4:** Start the Ready Q according the shortest Burst time by sorting according to lowest to highest burst time.
- Step 5:** Set the waiting time of the first process as '0' and its turnaround time as its burst time.
- Step 6:** For each process in the ready queue, calculate
  - a) Waiting time (WT) for process(n) = WT of process (n-1) + Burst time (BT) of process(n-1)
  - b) Turn-around time for Process(n)= WT of Process(n)+ BT of process(n)
- Step 7:** Calculate
  - a) Average waiting time = Total waiting Time / Number of process
  - b) Average Turnaround time = Total Turnaround Time / Number of process
- Step 8:** Stop the process

### Assignment Instructions:

1. These implementations will be considered as your Lab Assignment for this week.
2. Do not attempt to talk or work in groups.
3. At the end of this lab, you will submit these implementations as a compressed file named as your roll number.
4. The compressed file should contain at least two shell scripts implementing the algorithms.
5. You can access your lab manuals from *netstorage.dsu.edu.pk*.
6. There will be only one submission attempt allowed on LMS so be sure before submitting your solution.
7. As a backup, you are required to submit the compressed file on [oslabatdsu@gmail.com](mailto:oslabatdsu@gmail.com) with subject as Lab13-cs151111-4A.
8. If you are sitting in a section you don't have access to LMS of; you will send an email with the subject as Lab13-cs151111-4Awith4C.
9. Students having submission only on email (except those mentioned in para 8) will bear the penalty of 50% marks deduction.
10. **Beware!!!** In order to avoid plagiarism conviction (which will lead to **-100%** marks in this assignment); Delete your solutions after submitting them so that other students would not be able to access them.