



SCHEDULING ALGORITHM



**SHORTEST
JOB FIRST**



SCHEDULING ALGORITHM

ACTIVITY 2 **SJF**

SJF ALGORITHM

- In the Shortest Job First scheduling algorithm, the processes are scheduled in ascending order of their CPU burst times, i.e. the CPU is allocated to the process with the shortest execution time.

SJF non-preemptive scheduling

- In the non-preemptive version, once a process is assigned to the CPU, it runs into completion. Here, the short term scheduler is invoked when a process completes its execution or when a new process(es) arrives in an empty ready queue.

SJF preemptive scheduling

- This is the preemptive version of SJF scheduling and is also referred as Shortest Remaining Time First (SRTF) scheduling algorithm. Here, if a short process enters the ready queue while a longer process is executing, process switch occurs by which the executing process is swapped out to the ready queue while the newly arrived shorter process starts to execute. Thus the short term scheduler is invoked either when a new process arrives in the system or an existing process completes its

SJF ALGORITHM

Features of SJF Algorithm

- SJF allocates CPU to the process with shortest execution time.
- In cases where two or more processes have the same burst time, arbitration is done among these processes on first come first serve basis.
- There are both preemptive and non-preemptive versions.
- It minimizes the average waiting time of the processes.
- It may cause starvation of long processes if short processes continue to come in the system.

SJF ALGORITHM

Solution If the CPU scheduling policy is SJF non-preemptive, calculate the average waiting time and average turnaround time.

Gantt Chart

	P4		P1		P3		P5		P2	
0		6		7		9		12		16

SJF ALGORITHM

Consider the set of 5 processes whose arrival time and burst time are given below.

Process	AT	BT	ET	TAT	WT
P1	3	1	7	4	3
P2	1	4	16	15	11
P3	4	2	9	5	3
P4	0	6	6	6	0
P5	2	3	12	10	7

SJF ALGORITHM

FORMULA

ET = Exit Time

TAT = Turn Around Time

AT = Arrival Time

ET = Gantt chart

$TAT = ET - AT$

$WT = TAT - BT$



ACTIVITY 2

SJF

ACTIVITY 2:

SJF ALGORITHM


INSTRUCTIONS

1. Make a java program that will compute for the following.
 - a. Exit time
 - b. Turn around time
 - c. Waiting time
 - d. Average turn around time
 - e. Average waiting time
2. The user will have the choice to input 3 to 5 process, including Arrival Time and Burst time.
3. If the CPU scheduling policy is SJF pre-emptive, calculate the average waiting time and average turnaround time.
4. The program will ask the user if he/she wants to try again?
5. Submit the java file using the filename format: SNFN2.java
6. Provide a sample output of the program.

Criteria: 50 points

1. Complete and running program no errors. **50 points.**
2. Incomplete but running program no errors. **35 points.**
3. Running but no correct output **15 points.** (for the effort)
4. 2nd checking must be complete and working no errors **20 points.**
5. Programs to be checked should be error free. No checking of program if error persists.
6. Once I checked the program it is recorded. If you request for 2nd checking you be in the last priority.

PROGRAM OUTPUT



ACTIVITY 2: SJF ALGORITHM

ACTIVITY 2: SJF ALGORITHM

Title: SJF SCHEDULING ALGORITHM

Enter no. of process: (IN)

Process	AT	BT	ET	TAT	WT
P1	(IN)	(IN)	(0)	(0)	(0)
P2	(IN)	(IN)	(0)	(0)	(0)
P3	(IN)	(IN)	(0)	(0)	(0)
P4	(IN)	(IN)	(0)	(0)	(0)
P5	(IN)	(IN)	(0)	(0)	(0)

Average TAT: (0)

Average WT: (0)

Try Again: [Y/N]

Gannt Chart:

(Example)

| P1 | P2 | P3 | P4 | P5 |
0 4 7 8 10 14



EOA



SAMPLE OUTPUT FOR CHECKING
PURPOSE ONLY

Activity 2: SJF (SRTF) Algorithm

Process	Arrival	Burst	Completion	Turnaround	Waiting
P1	0	8	19	19	11
P2	1	4	7	6	2
P3	2	9	28	26	17
P4	3	5	12	9	4
P5	4	2	6	2	0

Average Turnaround Time = 12.4 ms

Average Waiting Time = 6.8 ms

Gantt Chart:
(Example)

