Santa Clara University

Department of Computer Engineering
Advanced Operating Systems (COEN 383)

Project-2 Preview (6 pts)
Instructor: Ahmed Ezzat

Process Scheduling Algorithms

We will build simulation written in Java or C programming language that experiment with different runs using different process scheduling algorithms:

- First-come first-served (FCFS) // non-preemptive
- Shortest job first (SJF) // non-preemptive
- Shortest remaining time (SRT) // preemptive
- Round robin (RR) // preemptive
- Highest priority first (HPF) // both non-preemptive and preemptive

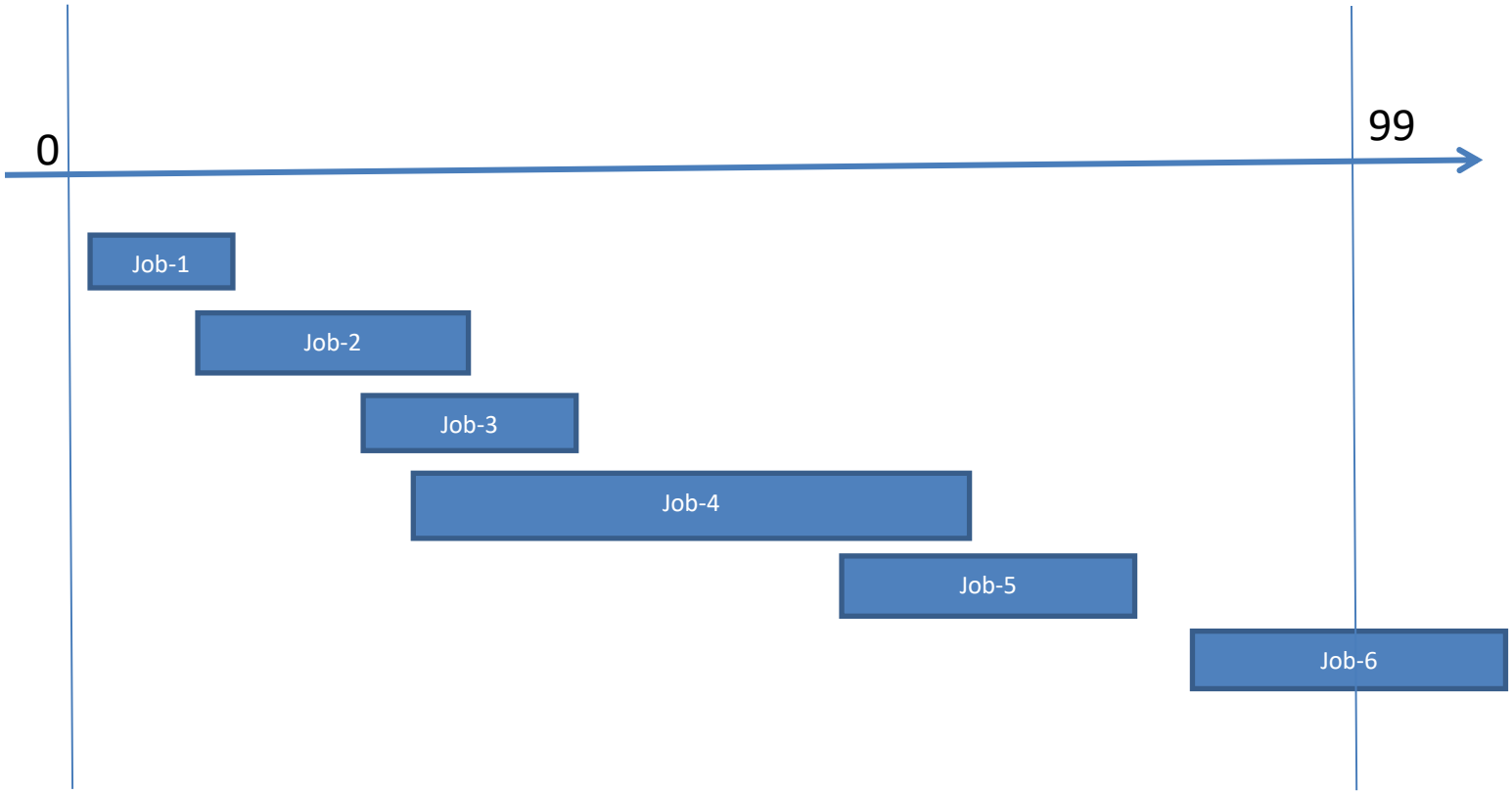
The total simulation time is 100 quantum/time-units.

First generate your workload. A process is represented by
<arrival time, runtime, priority>

- Use specific seed value for your random number generator.
- Job can arrive anytime up to 99, and job service time is anytime up to 10 quantum. Job is assigned priority 1..4.
- Unix **rand**() function returns random number between 0 and RAND_MAX (32767).
- ```
  #include   <stdlib.h>
  int   main()
  {
        int    seed   = time(NULL);
        srand(seed);                          // guarantee consistency when debugging
        int   arrival_time = rand()   % 100;      // will return num between 0 and 99
        int   service_time = (rand() % 10) + 1; // will return num between 1 and 10
  ```
  ```

```
        int     priority = (rand() % 4) + 1;            // priority between 1 .. 4
}
```

- Generate ~10 jobs, sort them based on arrival time. Run and verify that CPU is never idle more than 2 quanta waiting for work to do. Otherwise increase number of jobs.



- No process is allowed if start time > 99, but a job can complete after time = 100 quantum.
- CPU is scheduled at quanta boundary, i.e., if processes completed before end of quanta then CPU will be idle the remaining of this quanta
- Generate 5 sets of workloads. Each algorithm is run 5 times and get average per algorithm.

Definitions:

- Turnarund time (TAT): Time required for a particular process to complete, from submission time to completion. It is equal to the sum total of *Waiting time* and *Execution time*.
- Response time (RPT): The time taken in a program from the issuance of a command to the commence/beginning of a response to that command.(i.e., the time-interval between submission of a request, and the first response to that request).
- Wait time = TAT – Service time