

NJIT

A white curved line that starts under the 'N' and sweeps upwards and to the right, ending under the 'T'.

New Jersey's Science &  
Technology University

*THE EDGE IN KNOWLEDGE*

Alex. Gerbessiotis

CS Department



**The yellowish/brownish notes are copyright the textbook author.**

**The bluish notes are copyright the instructor and are add-ons and provide more explanation or additional details**



# Module 1

*Operating  
Systems:  
Internals  
and  
Design  
Principles*

# Chapter 9 Uniprocessor Scheduling

Ninth Edition  
By William Stallings

# Processor Scheduling

- Aim is to assign processes to be executed by the processor in a way that meets system objectives, such as response time, throughput, and processor efficiency
- Broken down into three separate functions:



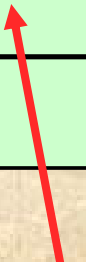
# Table 9.1

Previously:  
Job Scheduler or Scheduler



## Types of Scheduling

|                        |                                                                                                    |
|------------------------|----------------------------------------------------------------------------------------------------|
| Long-term scheduling   | The decision to add to the pool of processes to be executed                                        |
| Medium-term scheduling | The decision to add to the number of processes that are partially or fully in main memory          |
| Short-term scheduling  | The decision as to which available process will be executed by the processor                       |
| I/O scheduling         | The decision as to which process's pending I/O request shall be handled by an available I/O device |



Previously:  
Dispatcher, or CPU Dispatcher, or CPU Scheduler

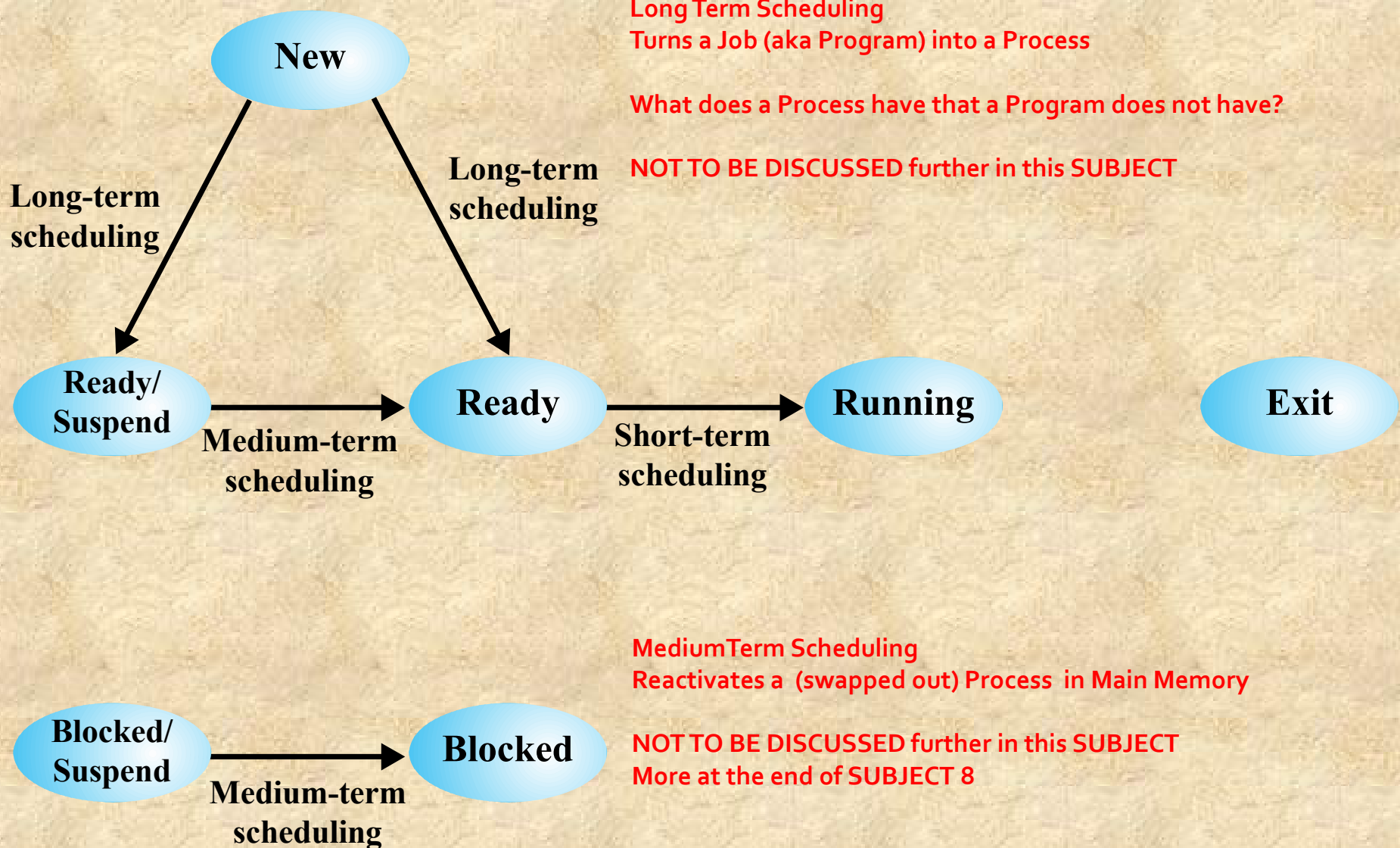
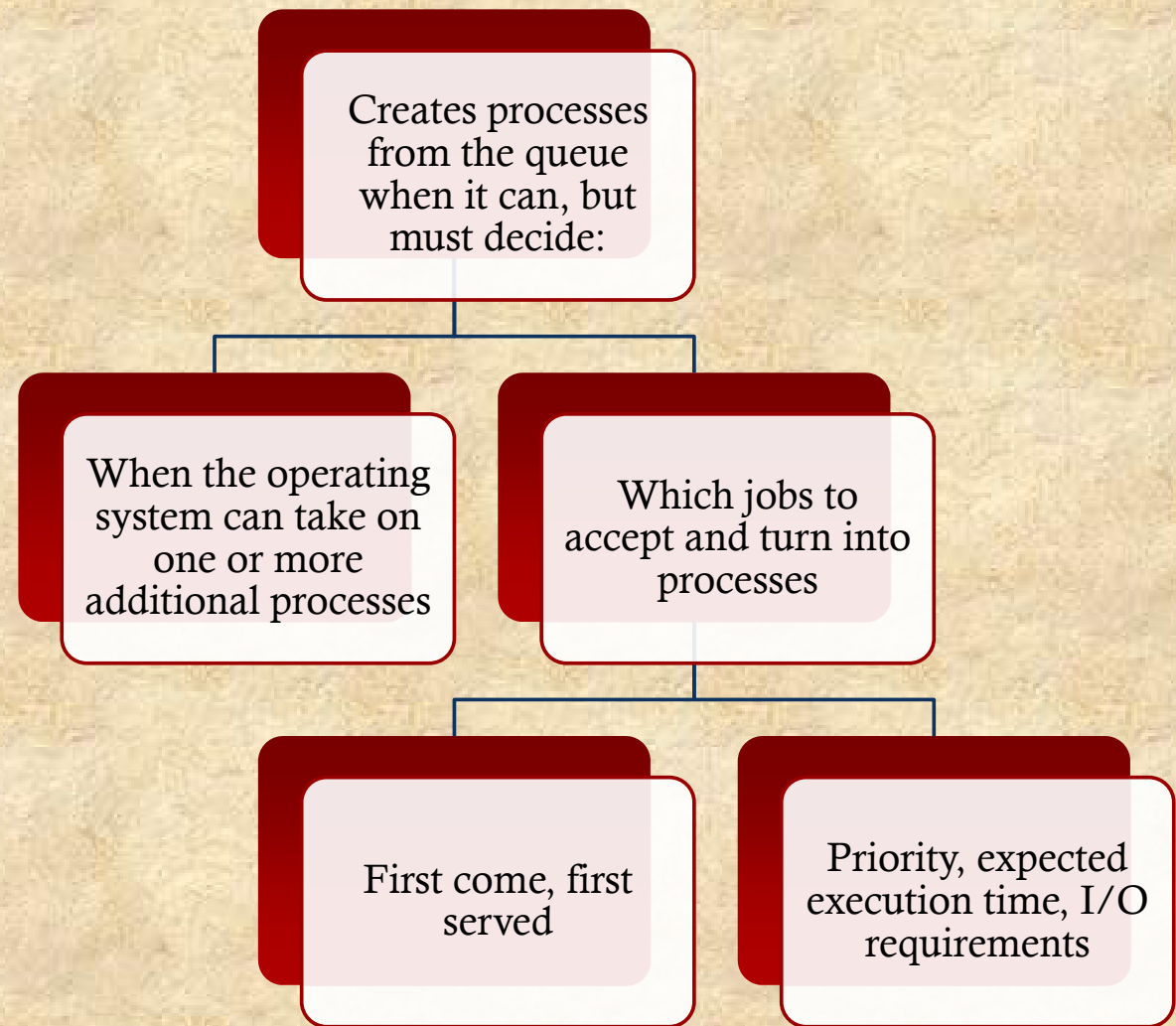


**Figure 9.1 Scheduling and Process State Transitions**

# Long-Term Scheduler

- Determines which programs are admitted to the system for processing
- Controls the **degree of multiprogramming**
  - The more processes that are created, the smaller the percentage of time that each process can be executed
  - May limit to provide satisfactory service to the current set of processes



# Medium-Term Scheduling

- Part of the swapping function
- Swapping-in decisions are based on the need to manage the degree of multiprogramming
  - Considers the memory requirements of the swapped-out processes

# Short-Term Scheduling

- Known as the **dispatcher**
- Executes most frequently
- Makes the fine-grained decision of which process to execute next
- Invoked when an event occurs that may lead to the blocking of the current process or that may provide an opportunity to preempt a currently running process in favor of another

## Examples:

- Clock interrupts
- I/O interrupts
- Operating system calls
- Signals (e.g., semaphores)



## Decision by scheduler

### Long Term:

A new program is to become a process (entry into PL, 4ptrs, initialization of kernel data structures)  
[Degree of multi-programming is determined by LT scheduler]  
**A process gets its entry in the process list (PID)**

### Medium Term:

Process from Main Memory (MM) to disk or the other way around  
**Process image copy on disk to MM or other way around**

### Short Term:

Process from MM still in MM but state Running or Ready or Blocked or Exit  
**Process Image in MM, preemption**



CPU Dispatcher aka Dispatcher aka Short term scheduler

## Qualitative criteria

Too difficult to describe or quantify

## Quantitative criteria

User oriented

**Response time**

**Turnaround time**

System oriented

**CPU (Processor) utilization**

**Throughput**

Scheduler deals with processes and their states i.e. queues associated with those states

# Short Term Scheduling Criteria

- Main objective is to allocate processor time to optimize certain aspects of system behavior
- A set of criteria is needed to evaluate the scheduling policy

## User-oriented criteria

- Relate to the behavior of the system as perceived by the individual user or process (such as response time in an interactive system)
- Important on virtually all systems

## System-oriented criteria

- Focus is on effective and efficient utilization of the processor (rate at which processes are completed)
- Generally of minor importance on single-user systems

# Short-Term Scheduling Criteria: Performance

## Examples:

- Response time and throughput

## Example:

- Predictability

Criteria can  
be classified  
into:

Performance-related

Non-performance  
related

Quantitative

Easily  
measured

Qualitative

Hard to  
measure

#### User Oriented, Performance Related

**Turnaround time** This is the interval of time between the submission of a process and its completion. Includes actual execution time plus time spent waiting for resources, including the processor. This is an appropriate measure for a batch job.

**Response time** For an interactive process, this is the time from the submission of a request until the response begins to be received. Often a process can begin producing some output to the user while continuing to process the request. Thus, this is a better measure than turnaround time from the user's point of view. The scheduling discipline should attempt to achieve low response time and to maximize the number of interactive users receiving acceptable response time.

**Deadlines** When process completion deadlines can be specified, the scheduling discipline should subordinate other goals to that of maximizing the percentage of deadlines met.

#### User Oriented, Other

**Predictability** A given job should run in about the same amount of time and at about the same cost regardless of the load on the system. A wide variation in response time or turnaround time is distracting to users. It may signal a wide swing in system workloads or the need for system tuning to cure instabilities.

#### System Oriented, Performance Related

**Throughput** The scheduling policy should attempt to maximize the number of processes completed per unit of time. This is a measure of how much work is being performed. This clearly depends on the average length of a process but is also influenced by the scheduling policy, which may affect utilization.

**Processor utilization** This is the percentage of time that the processor is busy. For an expensive shared system, this is a significant criterion. In single-user systems and in some other systems, such as real-time systems, this criterion is less important than some of the others.

#### System Oriented, Other

**Fairness** In the absence of guidance from the user or other system-supplied guidance, processes should be treated the same, and no process should suffer starvation.

**Enforcing priorities** When processes are assigned priorities, the scheduling policy should favor higher-priority processes.

**Balancing resources** The scheduling policy should keep the resources of the system busy. Processes that will underutilize stressed resources should be favored. This criterion also involves medium-term and long-term scheduling.

## Table 9.2

# Scheduling Criteria

waiting time:  
if defined, it means time  
process is in ready state,  
not blocked

Chapter 2 / Subject 2  
defined

(Table can be found on page 403 in textbook)

# Practice 1



- 1.1. A process got preempted before its ..... expired. Is the missing word a timeslice or quantum?
- 1.2. Which scheduler assigns a process an entry in the Process List?
- 1.3. What do we call the index of the Process List associated with a given process. (Use prevalent name in Linux.)
- 1.4. What does a process have that a program does not have?
- 1.5. In a time sharing system four jobs are waiting for execution in this specific order (left-most job executing first): A,B,C,D. They arrived in the queue at the same time  $t=0$  though. The timeslice is 1s. What is the response time for job D? Each job has execution time 2s. What is the wait time for jobs A, B, C, D respectively? What is the average wait time for the system?
- 1.6. It is 10pm; since 9pm (past hour) 10 new processes were created. No process exited. The number of processes in the system is 1000. The quantum is 1ms. How many was the Job Scheduler invoked since 9pm? How many time was the Dispatcher was invoked? How many context switched? Be as precise as possible!



# Module 2



**Arrival time**

**Completion (Finish) time**

**Turnaround time ( $T_r$ )**

**Response time ( $R$ )**

**Service time ( $s$  or  $T_s$ )**

**Execution time ( $e$  or  $T_e$ )**

**Remaining exec time**

**Wait time**

**Quantum ( $q$ )**

**Timeslice ( $t_s$ )**

**Throughput**

**Starvation**

**CPU (processor) utilization**

time a process arrives in ready Q

time a process is done with running/ready Q

= Completion - Arrival

time from arrival CPU execs first instruction  
of the arrived process

time a process expects to use CPU

time a process has already used CPU ( $e \leq s$ )

$s - e$

time a process is waiting/in the ready queue  
(I/O does not count!)

**Round-robin in multitasking**

(A quantum's stay in CPU can be preempted)

**Timesharing related. Not to be used.**

**#processes completed in unit of time**

Process is starved from using the CPU

See Subject 2. Percent of time CPU is busy  
with processes



**FCFS** First-Come-First-Served  $\max[w]$

**SPN** Shortest-Process-Next  $\min[s]$

**SRN\*** Shortest-RemainingT-Next  $\min[s-e]$   
(from now on **SRT\*** instead of SRN)

[preemptive at arrival of a process]

**RR** Round Robin (quantum)  $\text{What if } q = \infty?$   
 $\infty = \text{infinity}$

[preemptive]

\* SRN and SRT will be used interchangeably. N=next , T=time

# Selection Function

- Determines which process, among ready processes, is selected next for execution
- May be based on priority, resource requirements, or the execution characteristics of the process
- If based on execution characteristics, then important quantities are:
  - $w$  = time spent in system so far, waiting
  - $e$  = time spent in execution so far
  - $s$  = total service time required by the process, including  $e$ ; generally, this quantity must be estimated or supplied by the user

# Decision Mode

- Specifies the instants in time at which the selection function is exercised
- Two categories:
  - Nonpreemptive
  - Preemptive

# Nonpreemptive vs Preemptive

## Nonpreemptive

- Once a process is in the running state, it will continue until it terminates or blocks itself for I/O

## Preemptive

- Currently running process may be interrupted and moved to ready state by the OS
- Decision to preempt may be performed when a new process arrives, when an interrupt occurs that places a blocked process in the Ready state, or periodically, based on a clock interrupt

|                            | <b>FCFS</b>                                                                     | <b>Round robin</b>                              | <b>SPN</b>                                      | <b>SRT</b>                  |
|----------------------------|---------------------------------------------------------------------------------|-------------------------------------------------|-------------------------------------------------|-----------------------------|
| <b>Selection function</b>  | $\max[w]$                                                                       | constant                                        | $\min[s]$                                       | $\min[s - e]$               |
| <b>Decision mode</b>       | Non-preemptive                                                                  | Preemptive (at time quantum)                    | Non-preemptive                                  | Preemptive (at arrival)     |
| <b>Through-Put</b>         | Not emphasized                                                                  | May be low if quantum is too small              | High                                            | High                        |
| <b>Response time</b>       | May be high, especially if there is a large variance in process execution times | Provides good response time for short processes | Provides good response time for short processes | Provides good response time |
| <b>Overhead</b>            | Minimum                                                                         | Minimum                                         | Can be high                                     | Can be high                 |
| <b>Effect on processes</b> | Penalizes short processes; penalizes I/O bound processes                        | Fair treatment                                  | Penalizes long processes                        | Penalizes long processes    |
| <b>Starvation</b>          | No                                                                              | No                                              | Possible                                        | Possible                    |

**Table 9.3**

Characteristics of Various Scheduling Policies

(Table can be found on page 405 in textbook)

# First-Come-First-Served (FCFS)

- Simplest scheduling policy
- Also known as first-in-first-out (FIFO) or a strict queuing scheme
- As each process becomes ready, it joins the ready queue
- When the currently running process ceases to execute, the process that has been in the ready queue the longest is selected for running
- Performs much better for long processes than short ones
- Tends to favor processor-bound processes over I/O-bound processes

# Round Robin

- Uses preemption based on a clock
- Also known as **time slicing** because each process is given a slice of time before being preempted
- Principal design issue is the length of the time quantum, or slice, to be used
- Particularly effective in a general-purpose time-sharing system or transaction processing system
- One drawback is its relative treatment of processor-bound and I/O-bound processes

# Shortest Process Next (SPN)

- Nonpreemptive policy in which the process with the shortest expected processing time is selected next
- A short process will jump to the head of the queue
- Possibility of starvation for longer processes
- One difficulty is the need to know, or at least estimate, the required processing time of each process
- If the programmer's estimate is substantially under the actual running time, the system may abort the job

# Shortest Remaining Time (SRT)

- Preemptive version of SPN
- Scheduler always chooses the process that has the shortest expected remaining processing time
- Risk of starvation of longer processes
- Should give superior turnaround time performance to SPN because a short job is given immediate preference to a running longer job

Arrival of a new process in the ready queue can only preempt a running process (with the new process replacing it) ; according to the textbook a ready process cannot do so. Some variations exist.

| Process | Arrival Time | Service Time |
|---------|--------------|--------------|
| A       | 0            | 3            |
| B       | 2            | 6            |
| C       | 4            | 4            |
| D       | 6            | 5            |
| E       | 8            | 2            |

assume unit implied is s...

**Table 9.4**  
**Process Scheduling Example**

# FCFS (nonp)



| Process | Arrival | Serv(Ts) | Slot  | Finish | Turnaround(Tr) | Tr/Ts |
|---------|---------|----------|-------|--------|----------------|-------|
| A       | 0       | 3        | 0-3   | 3      | 3              | 1     |
| B       | 2       | 6        | 3-9   | 9      | 7              | 1.17  |
| C       | 4       | 4        | 9-13  | 13     | 9              | 2.25  |
| D       | 6       | 5        | 13-18 | 18     | 12             | 2.40  |
| E       | 8       | 2        | 18-20 | 20     | 12             | 6.00  |
| Average |         |          |       |        | 8.60           | 2.56  |

## FCFS

| Process  | A | B | C | D | E |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |
|----------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| Arrival  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Time     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |
| ServiceT | 3 |   | 6 |   | 4 |   | 5 |   | 2 |   |    |    |    |    |    |    |    |    |    |    |    |
| from     | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 0  |
| to       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 0  | 1  |
| CPU      | A | A | A | B | B | B | B | B | B | C | C  | C  | C  | D  | D  | D  | D  | D  | E  | E  |    |

Slot number indicates the start of a period. Thus a 6 indicates the period from t=6 to t=7

## What is the wait time for W? Wait(E)?

# Round Robin (q=1)



| Process | Arrival | Serv(Ts) | Slot | Finish | Turnaround(Tr) | Tr/Ts |
|---------|---------|----------|------|--------|----------------|-------|
| A       | 0       | 3        |      | 4      | 4              |       |
| B       | 2       | 6        |      | 18     | 16             |       |
| C       | 4       | 4        |      | 17     | 13             |       |
| D       | 6       | 5        |      | 20     | 14             |       |
| E       | 8       | 2        |      | 15     | 7              |       |
| Average |         |          |      |        | 10.80          | 2.71  |

## RR q=1

|        |      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--------|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| from   |      | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| to     |      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| CPU    |      | A | A | B | A | B | C | B | D | C | B | E | D | C | B | E | D | C | B | D | D |
| ReadyQ | Head |   |   | A | B | C | B | D | C | B | E | D | C | B | E | D | C | B | D |   |   |
|        |      |   |   |   | C |   |   | C | B | E | D | C | B | E | D | C | B | D |   |   |   |
|        |      |   |   |   |   |   |   |   | D | C | B | E | D | C | B |   |   |   |   |   |   |
| ReadyQ | Tail |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Assumption: New before Preempted

# Round Robin (q=4)



| Process | Arrival | Serv(Ts) | Slot | Finish | Turnaround(Tr) | Tr/Ts |
|---------|---------|----------|------|--------|----------------|-------|
| A       | 0       | 3        |      | 3      | 3              |       |
| B       | 2       | 6        |      | 17     | 15             |       |
| C       | 4       | 4        |      | 11     | 7              |       |
| D       | 6       | 5        |      | 20     | 14             |       |
| E       | 8       | 2        |      | 19     | 11             |       |
| Average |         |          |      |        | 10.00          | 2.71  |

## RR q=4

|              |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| from         | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| to           | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| CPU          | A | A | A | B | B | B | B | C | C | C | C | D | D | D | D | B | B | E | E | D |
| ReadQ Head   |   |   | B | C | C | C | C | D | D | D | D | B | B | B | B | E | E | D | D |   |
|              |   |   |   |   |   | D | D | B | B | B | B | E | E | E | E | D | D |   |   |   |
| Ready Q Tail |   |   |   |   |   |   |   | E | E | E | E |   |   |   |   |   |   |   |   |   |

# SPN (Shortest P next) nonp

| Process | Arrival | Serv(Ts) | Slot | Finish | Turnaround(Tr) | Tr/Ts |
|---------|---------|----------|------|--------|----------------|-------|
| A       | 0       | 3        |      | 3      | 3              |       |
| B       | 2       | 6        |      | 9      | 7              |       |
| C       | 4       | 4        |      | 15     | 11             |       |
| D       | 6       | 5        |      | 20     | 14             |       |
| E       | 8       | 2        |      | 11     | 3              |       |
| Average |         |          |      |        | 7.60           | 1.84  |

| SPN  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |
|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|
| from | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |  |
| to   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |  |
| CPU  | A | A | A | B | B | B | B | B | B | E | E | C | C | C | C | D | D | D | D | D |   |  |



# SRT (Shortest R time)

| Process | Arrival | Serv(Ts) | Slot | Finish | Turnaround(Tr) | Tr/Ts |
|---------|---------|----------|------|--------|----------------|-------|
| A       | 0       | 3        |      | 3      | 3              |       |
| B       | 2       | 6        |      | 15     | 13             |       |
| C       | 4       | 4        |      | 8      | 4              |       |
| D       | 6       | 5        |      | 20     | 14             |       |
| E       | 8       | 2        |      | 10     | 2              |       |
| Average |         |          |      |        | 7.20           | 1.59  |

## SRT

|      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| from | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| to   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| CPU  | A | A | A | B | C | C | C | C | E | E | B | B | B | B | D | D | D | D | D | D |

Every time a new process arrives algorithm SRT get recalculated as policy is preemptive

# Another Example



| Process | Arrival | Serv(Ts) | MoreInfo                                       |
|---------|---------|----------|------------------------------------------------|
| A       | 0s      | 5s       | After 3s of service; 1s I/O; then remaining 2s |
| B       | 3s      | 6s       | After 3s of service; 5s I/O; then remaining 1s |

## RR with $q=1s$

Arr A B

012345678901234567890 <<<Be careful how you read these numbers...

CPU AAABABAB B

I/O A BBBB

RQ?

What is the turn around time of A?  $Tr(A)=7s$

What is the service time of A?  $s(A)=5s$

What is the wait time of A?  $w(A)=1s$

(A waited in the ready queue between  $t=5s$  and  $t=6s$  while B was using it..

The blocked time of A is NOT wait time.)

# Traditional UNIX Scheduling

- Used in both SVR3 and 4.3 BSD UNIX
  - These systems are primarily targeted at the time-sharing interactive environment
- Designed to provide good response time for interactive users while ensuring that low-priority background jobs do not starve
- Employs multilevel feedback using round robin within each of the priority queues
- Makes use of one-second preemption
- Priority is based on process type and execution history

# Summary

- Types of processor scheduling
  - Long-term scheduling
  - Medium-term scheduling
  - Short-term scheduling
- Traditional UNIX scheduling
- Scheduling algorithms
  - Short-term scheduling criteria
  - The use of priorities
  - Alternative scheduling policies
  - Performance comparison

# NJIT

A white curved line that starts under the 'N', arches over the 'J' and 'I', and ends under the 'T'.

New Jersey's Science &  
Technology University

*THE EDGE IN KNOWLEDGE*

**Alex. Gerbessiotis**

**CS Department**