

Topic 10

Processor Scheduling

Objectives:

- Understand the purpose of long term scheduling, medium term scheduling and short term scheduling.
- Understand the following scheduling policies: First-Come-First-Serve, Round-Robin, Shortest Process Next, Shortest Remaining Time, Highest Response Ratio Next, and Feedback. <https://powcoder.com> Add WeChat powcoder
- Understand the basic scheduling policy of the traditional UNIX systems.

Readings:

- Stallings: Chapter 9.

Aim of Scheduling

- Assign processes to be executed by the processor(s)
- Response time
- Throughput
- Processor efficiency

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Table 9.1 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

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

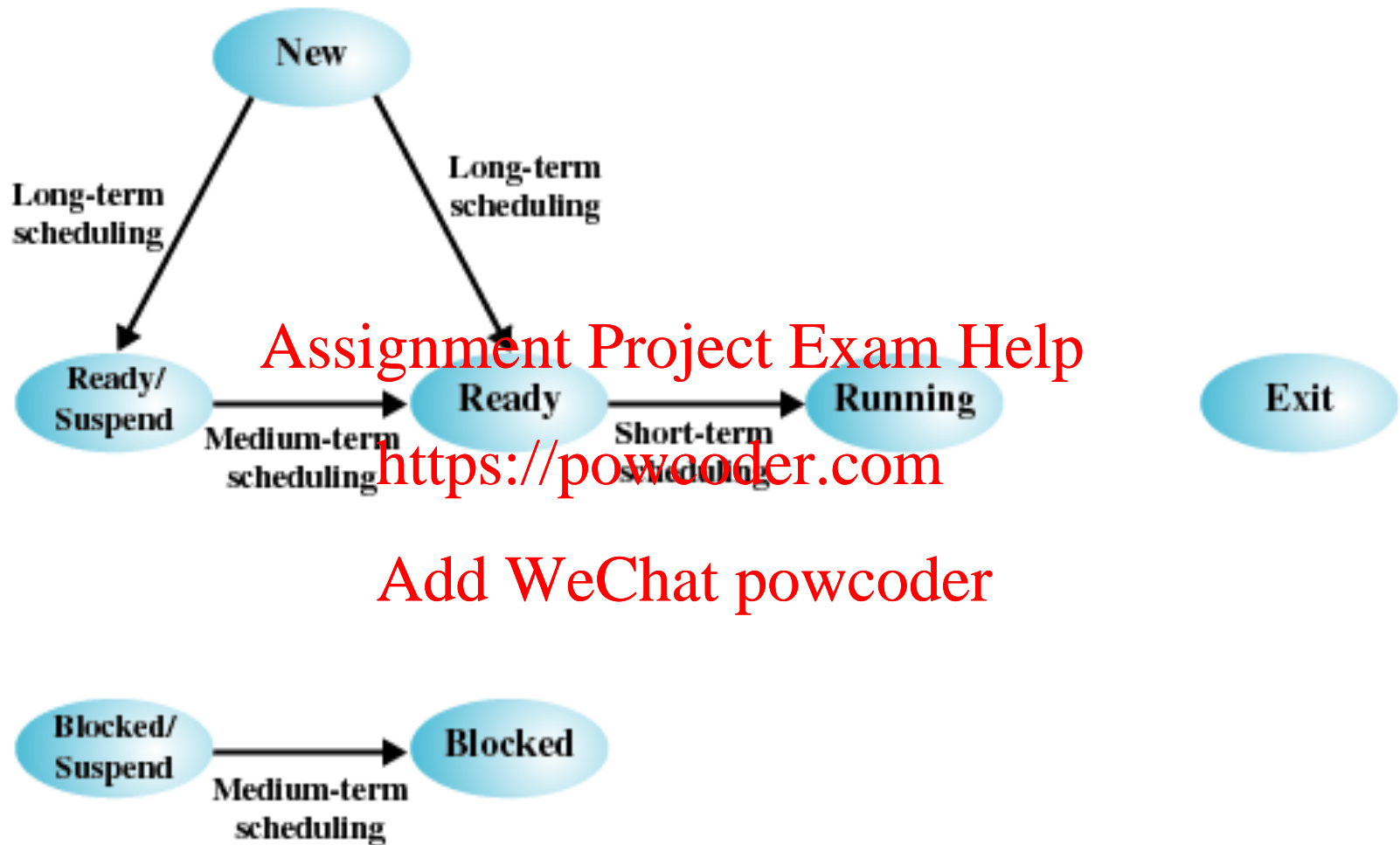


Figure 9.1 Scheduling and Process State Transitions

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

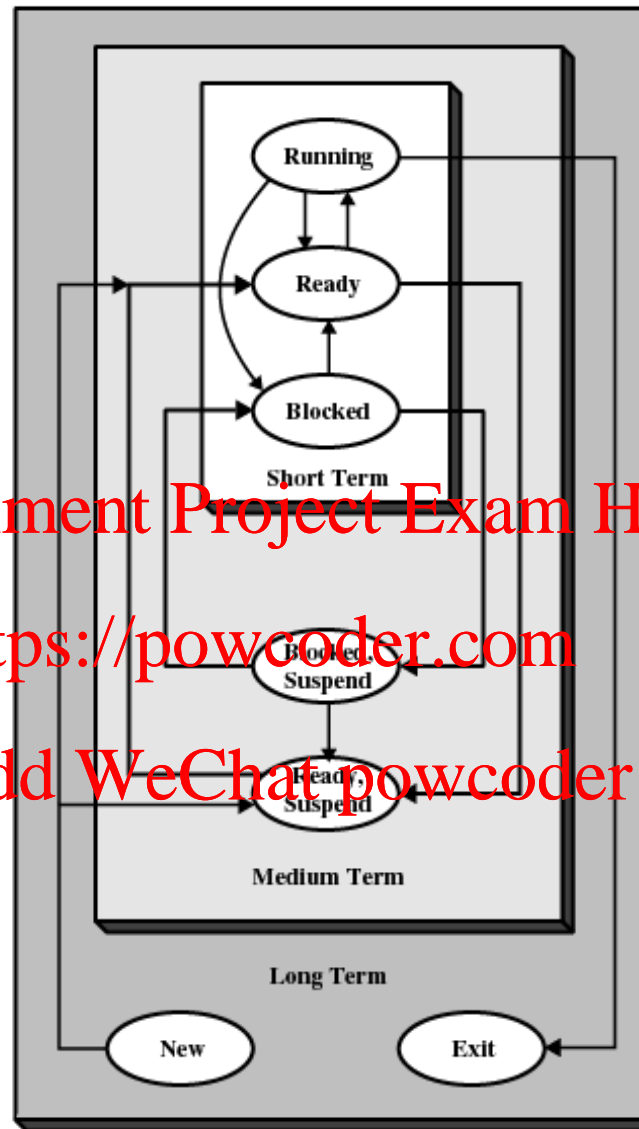


Figure 9.2 Levels of Scheduling

Long-Term Scheduling

- Determines which programs are admitted to the system for processing
- Controls the degree of multiprogramming
- More processes, smaller percentage of time each process is executed

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Medium-Term Scheduling

- Part of the swapping function
- Based on the need to manage the degree of multiprogramming

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Short-Term Scheduling

- Known as the dispatcher
- Executes most frequently
- Invoked when an event occurs
 - Clock interrupts
 - I/O interrupts
 - Operating system calls
 - Signals

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Short-Term Scheduling Criteria

- User-oriented
 - Response Time
 - Elapsed time between the submission of a request until there is output.
- System-oriented
 - Effective and efficient utilization of the processor
- Performance-related
 - Quantitative
 - Measurable such as response time and throughput

Table 9.2 Scheduling Criteria

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.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

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.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

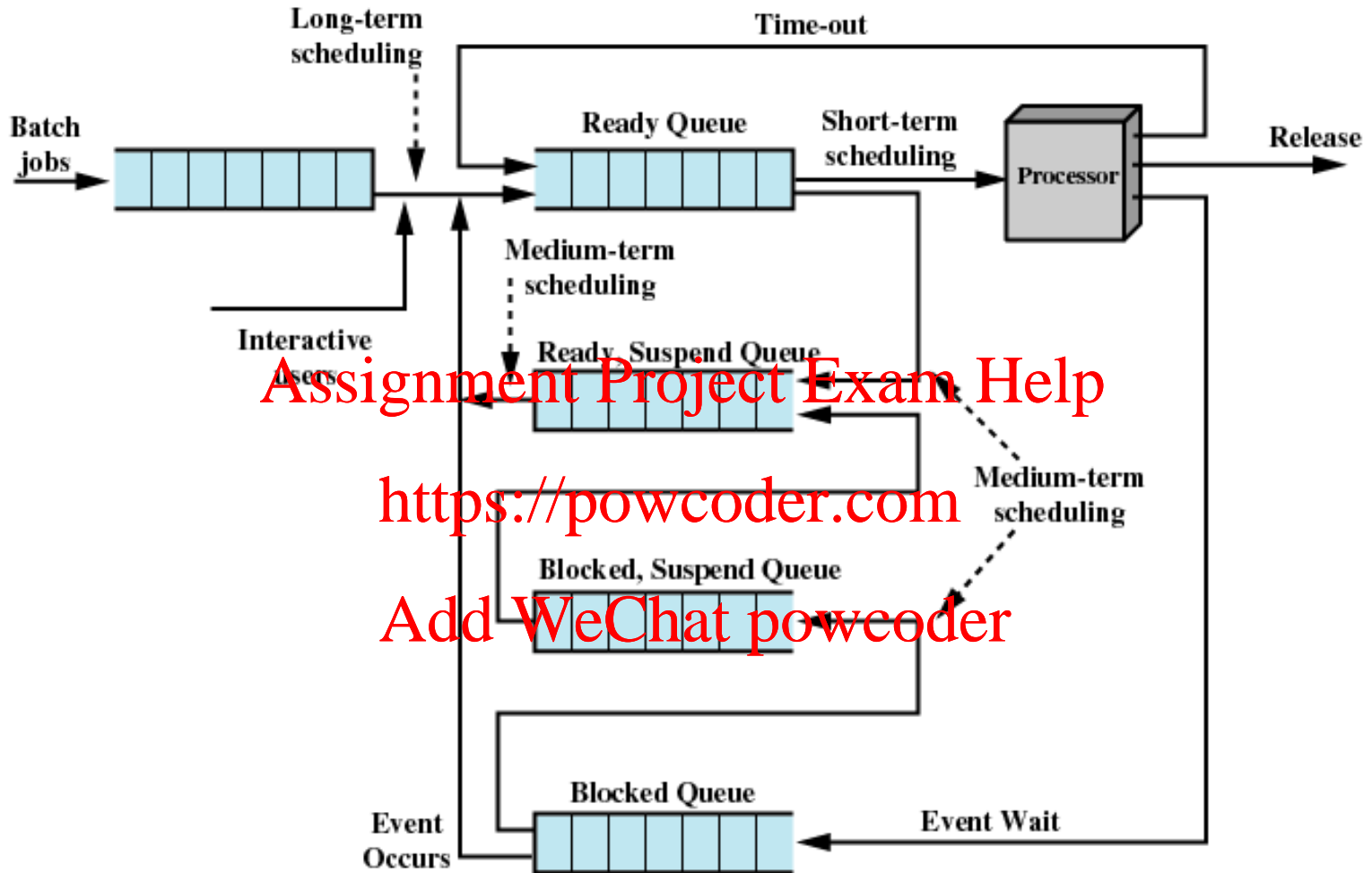


Figure 9.3 Queuing Diagram for Scheduling

Priorities

- Scheduler will always choose a process of higher priority over one of lower priority
- Have multiple ready queues to represent each level of priority
- Lower-priority may suffer starvation
 - Allow a process to change its priority based on its age or execution history

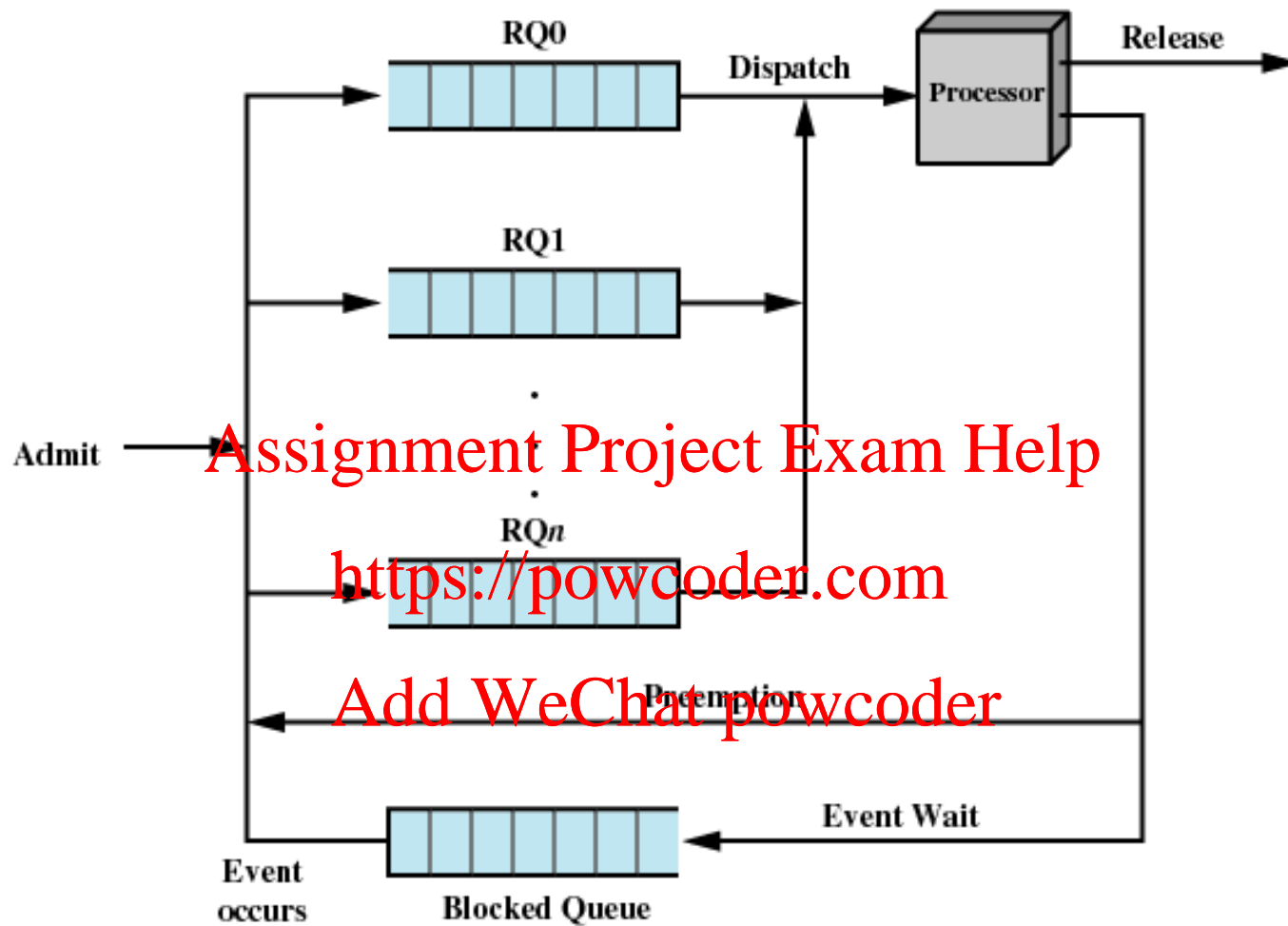


Figure 9.4 Priority Queuing

Decision Mode

- 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 the Ready state by the operating system
 - Allows for better service since any one process cannot monopolize the processor for very long

Assignment Project Exam Help

<https://powcoder.com>

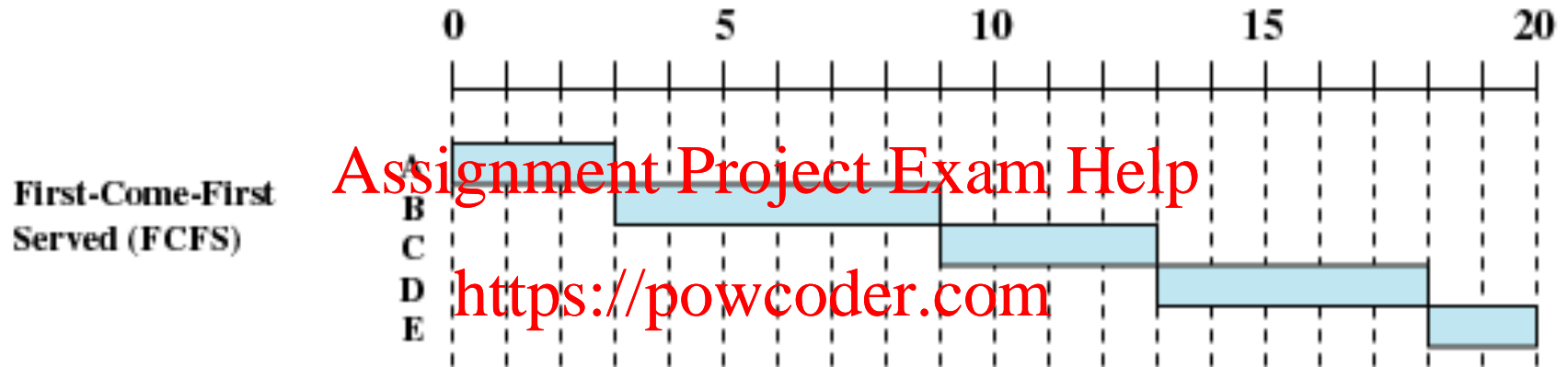
Add WeChat powcoder

Process Scheduling Example

Table 9.4 Process Scheduling Example

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

First-Come-First-Served (FCFS)



Add WeChat powcoder

- Each process joins the Ready queue
- When the current process ceases to execute, the oldest process in the Ready queue is selected

First-Come-First-Served (FCFS)

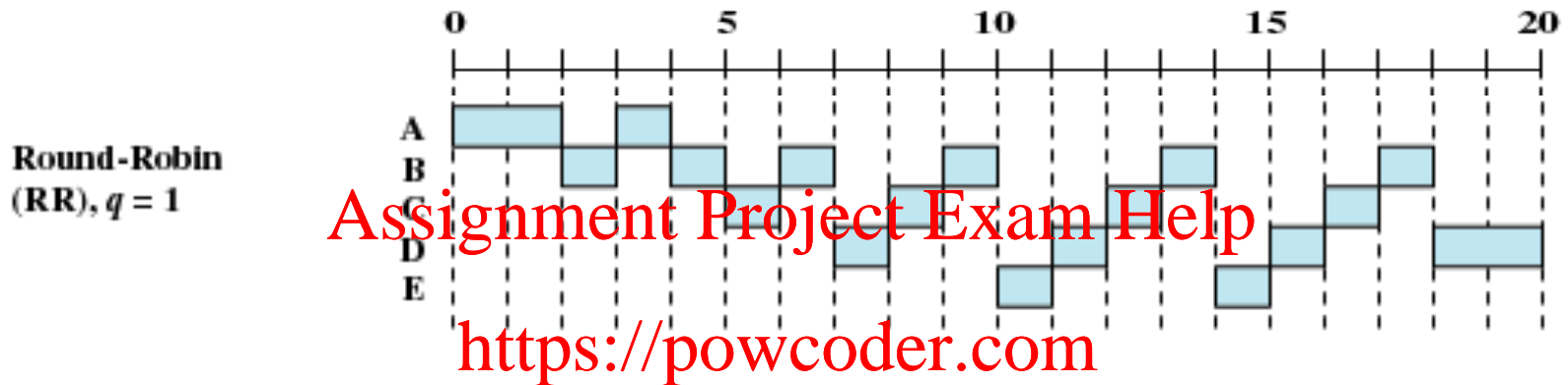
- A short process may have to wait a very long time before it can execute
- Favors CPU-bound processes
 - I/O processes have to wait until CPU-bound process completes

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Round-Robin



Add WeChat powcoder

- Uses *preemption* based on a clock
- An amount of time is determined that allows each process to use the processor for that length of time

Round-Robin

- Clock interrupt is generated at periodic intervals
- When an interrupt occurs, the currently running process is placed in the ready queue
 - Next ready job is selected
- Known as time slicing

Virtual Round-Robin Scheduling

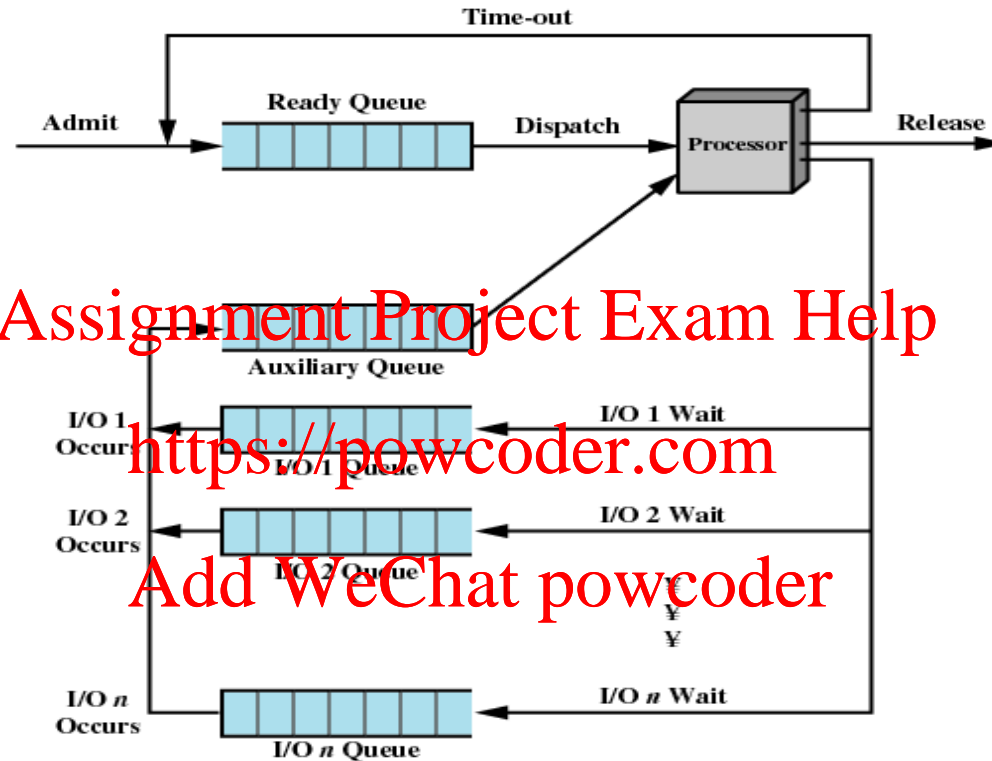
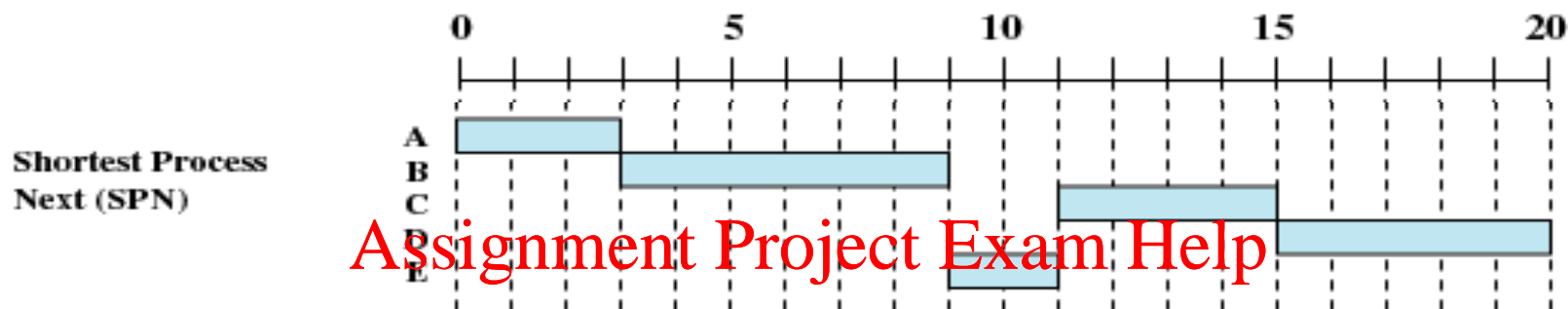


Figure 9.7 Queuing Diagram for Virtual Round-Robin Scheduler

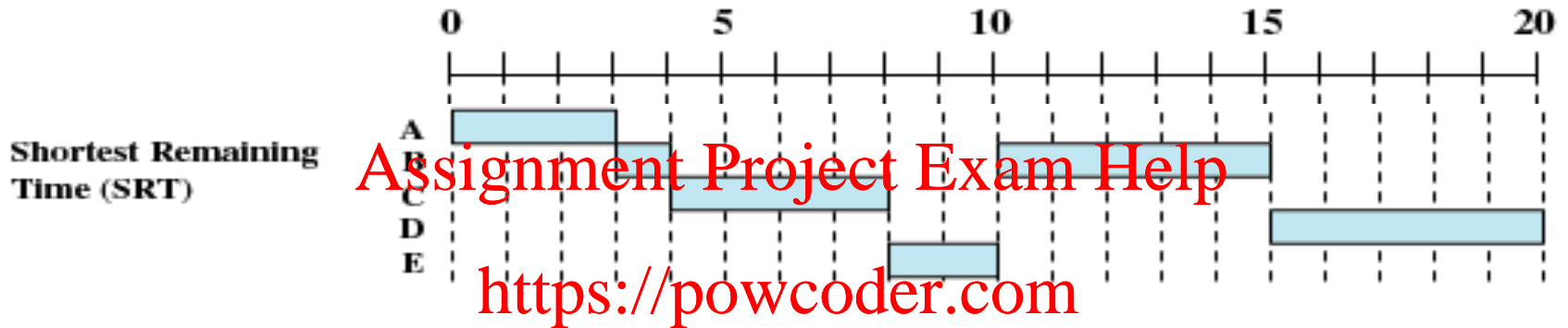
- Processes in the Auxiliary Queue get preference over those in the Ready Queue

Shortest Process Next



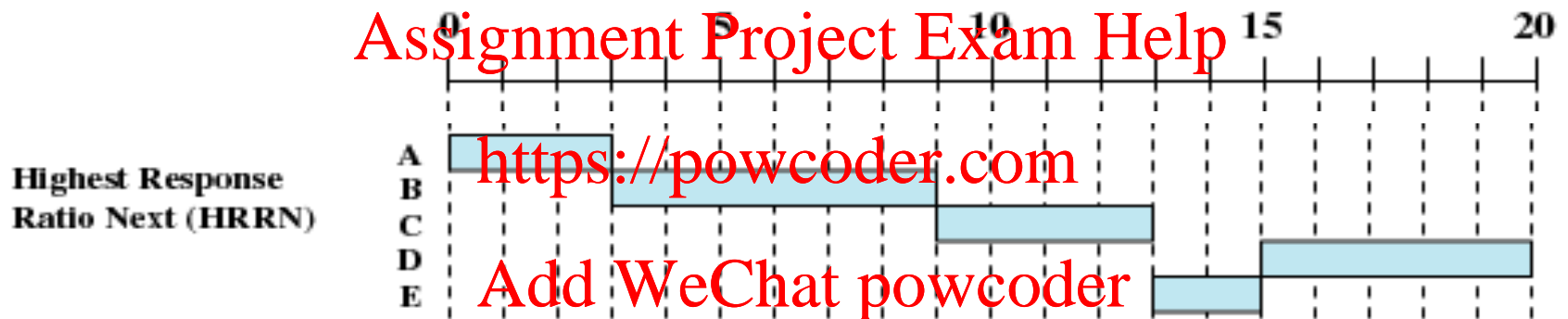
- Nonpreemptive policy
- Process with shortest expected processing time is selected next
- Short process jumps ahead of longer processes
- Must estimate the processing time
- Possibility of starvation for longer processes

Shortest Remaining Time



- Preemptive version of shortest process next policy
- Must estimate processing time

Highest Response Ratio Next (HRRN)



- Choose next process with the greatest ratio

$$\text{ratio} = \frac{\text{time spent waiting} + \text{expected service time}}{\text{expected service time}}$$

Multilevel Feedback

- Penalize jobs that have been running longer
- Don't know remaining time process needs to execute

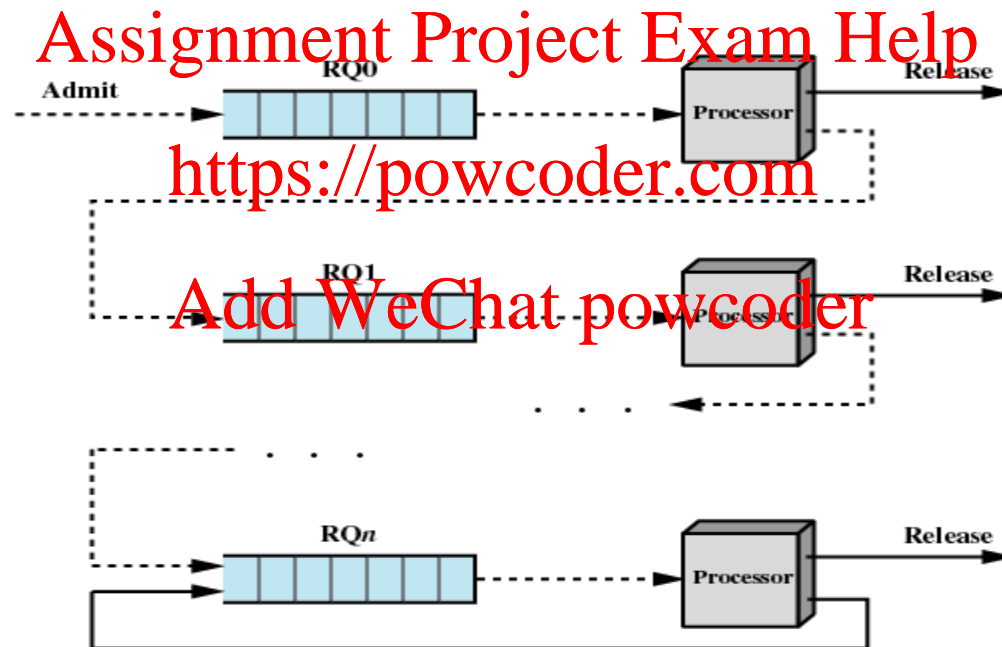


Figure 9.10 Feedback Scheduling

Table 9.3 Characteristics of Various Scheduling Policies

	Selection Function	Decision Mode	Throughput	Response Time	Overhead	Effect on Processes	Starvation
FCFS	$\max[w]$	Nonpreemptive	Not emphasized	May be high, especially if there is a large variance in process execution times	Minimum	Penalizes short processes; penalizes I/O bound processes	No
Round Robin	constant	Preemptive (at time quantum)	May be low if quantum is too small	Provides good response time for short processes	Minimum	Fair treatment	No
SPN	$\min[s]$	Nonpreemptive	High	Provides good response time for short processes	Can be high	Penalizes long processes	Possible
SRT	$\min[s - e]$	Preemptive (at arrival)	High	Provides good response time	Can be high	Penalizes long processes	Possible
HRRN	$\max\left(\frac{w + s}{s}\right)$	Nonpreemptive	High	Provides good response time	Can be high	Good balance	No
Feedback	(see text)	Preemptive (at time quantum)	Not emphasized	Not emphasized	Can be high	May favor I/O bound processes	Possible

w = time spent waiting

e = time spent in execution so far

s = total service time required by the process, including e

Traditional UNIX Scheduling

- Multilevel feedback using round robin within each of the priority queues
<https://www.powcoder.com>
Assignment Project Exam Help
- If a running process does not block or complete within 1 second, it is preempted
Add WeChat powcoder
- Priorities are recomputed once per second
- Base priority divides all processes into fixed bands of priority levels

Bands

- Decreasing order of priority
 - Swapper
 - Block I/O device control
 - File manipulation
 - Character I/O device control
 - User processes

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder