Operatines ignifest to the Help

https://powcoder.com

Lecture 4a

Previously

Scheduling

- O Time scales: Long-, madising in the heroteth expension property in the scales in the scale in the sc
- O Scheduling criteria
- O Scheduling algorithms: FCFS, SJF, SRI, RR oder.com

Today

Scheduling

- O Performance overview of lest week proheduling polities; FCFS, SJF, SRT, RR
- O Another look at Round Robin
- O Multi-level queue scheduling https://powcoder.com
- O Feedback scheduling Add WeChat powcoder
- O Parametrised Scheduling Algorithms
- O Real-time scheduling
- O Java thread scheduling

Recap: Scheduling basics

Questions:

- O What are CPU-bound Activity of the Project seems Help
- O What is starvation?
- https://powcoder.com
- O What is throughput?
- O What is turnaround time? Add WeChat powcoder
- O What is waiting time?
- O Why is it important for a scheduling algorithm to be fair?

Recap: Scheduling basics

Throughput and Turnaround Time



	First-Come First- Served		Shortest Remaining Time	Round Robin
Selection function	max waiting time Assignme	min execution time ent Project Exam	min remaining Oution time	max waiting time

https://powcoder.com

	First-Come First- Served		Shortest Remaining Time	Round Robin
Selection function	max waiting time Assignme	min execution time ent Project Exam	min remaining Oution time	max waiting time
Decision mode	non-preemptive https	non-preemptive ://powcoder.com	preemptive	Preemptive (at time quantum)

	First-Come First- Served	Shortest Job First	Shortest Remaining Time	Round Robin
Selection function	max waiting time Assignme	min execution time ent Project Exam	min remaining Pution time	max waiting time
Decision mode	non-preemptive https	non-preemptive ://powcoder.com	preemptive	Preemptive (at time quantum)
Overhead	very small Add	can be high WeChat powcode	can be high	very small

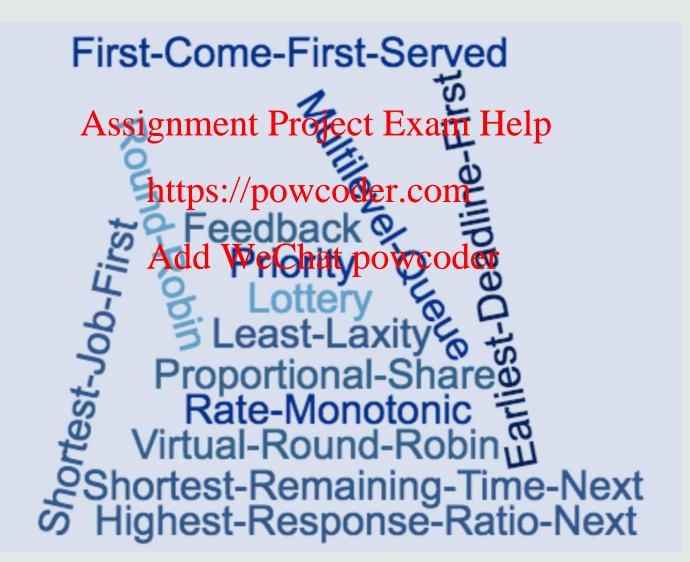
	First-Come First- Served	Shortest Job First	Shortest Remaining Time	Round Robin
Selection function	max waiting time Assignme	min execution time ent Project Exam	min remaining Oution time	max waiting time
Decision mode	non-preemptive https	non-preemptive ://powcoder.com	preemptive	Preemptive (at time quantum)
Overhead	very small Add	can be high WeChat powcode	can be high	very small
Fairness	penalises short processes	penalises long processes	penalises long processes	fair

	First-Come First- Served	Shortest Job First	Shortest Remaining Time	Round Robin
Selection function	max waiting time Assignme	min execution time ent Project Exam	min remaining Pution time	max waiting time
Decision mode	non-preemptive https	non-preemptive ://powcoder.com	preemptive	Preemptive (at time quantum)
Overhead	very small Add	can be high WeChat powcode	can be high	very small
Fairness	penalises short processes	penalises long processes	penalises long processes	fair
Starvation	no	possible	possible	no

	First-Come First- Served	Shortest Job First	Shortest Remaining Time	Round Robin
Selection function	max waiting time Assignme	min execution time ent Project Exam	min remaining Pution time	max waiting time
Decision mode	non-preemptive https	non-preemptive ://powcoder.com	preemptive	Preemptive (at time quantum)
Overhead	very small Add	can be high WeChat powcode	can be high	very small
Fairness	penalises short processes	penalises long processes	penalises long processes	fair
Starvation	no	possible	possible	no
Throughput	variable	high	high	depends on time quantum

	First-Come First- Served	Shortest Job First	Shortest Remaining Time	Round Robin
Selection function	max waiting time Assignme	min execution time ent Project Exam	min remaining Oution time	max waiting time
Decision mode	non-preemptive https	non-preemptive ://powcoder.com	preemptive	Preemptive (at time quantum)
Overhead	very small Add	can be high WeChat powcode	can be high	very small
Fairness	penalises short processes	penalises long processes	penalises long processes	fair
Starvation	no	possible	possible	no
Throughput	variable	high	high	depends on time quantum
Waiting time	can be high	good for short proc.	better than SJF	depends on time quantum

Scheduling Algorithms



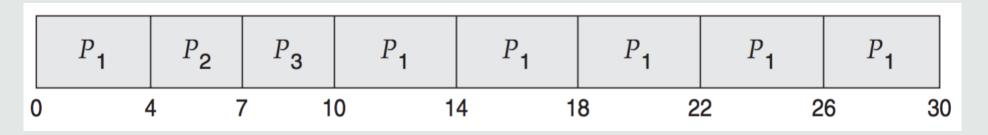
Another look at Round Robin

Recap:

- O FCFS + preemption at the end of the time quantum)
- O Implementation: FIFO que het psimpointe oder.com

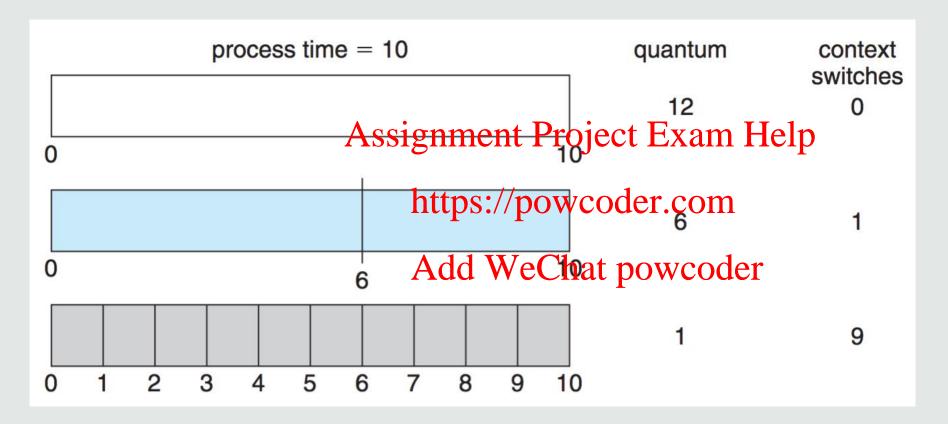
Add WeChat powcoder

	Process	Burst time
e	P	24
	P2	3
	P3	3



Time quantum: 4ms

How to choose the time quantum?



Quantum too large: Scheduling degenerates and works like FCFS

Quantum too short: Too many context switches

Rule of thumb: 80% of CPU bursts <= time quantum

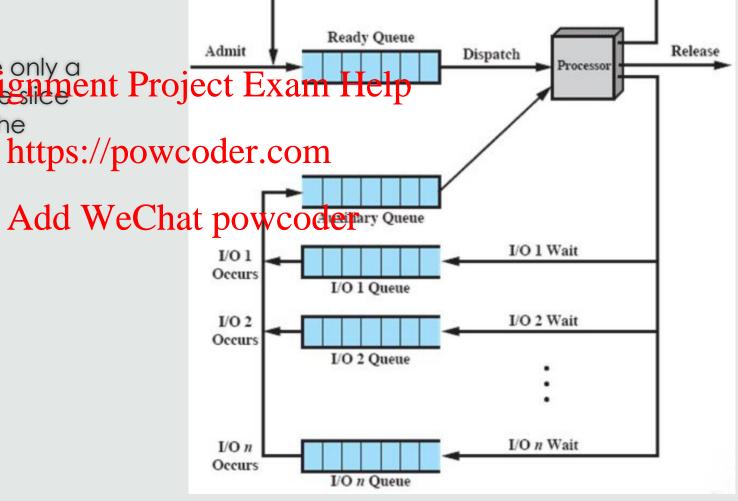
Virtual Round Robin

O Problem of RR

o I/O-bound processes use only a small fraction of their this gament Project Exam Help → spend a long time in the Ready queue

O VRR

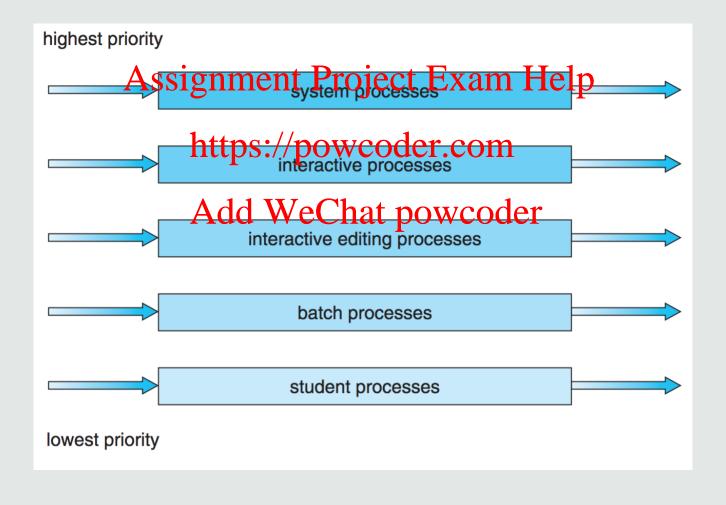
O Auxiliary queue → higher priority



Time-out

Multi-level Queue Scheduling

O Separate processes according to their expected behaviour



Multi-level Feedback Queue Scheduling

- Dynamically adapt to process behaviour
 - → move processes between queues as they change

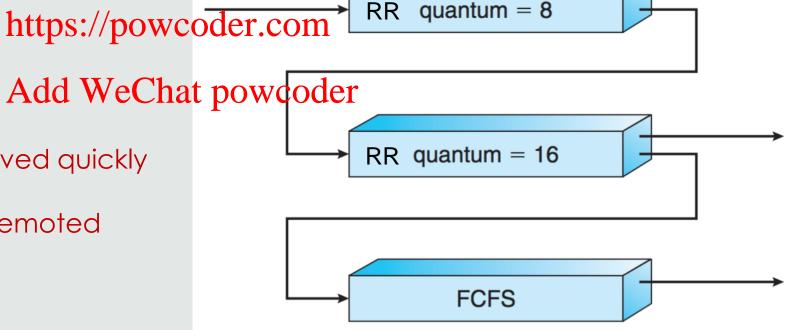
Assignment Project Exam Help

Example:

Add WeChat powdoder

I/O-bound processes get served quickly

CPU-bound processes get demoted



Parameterised Scheduling Algorithms

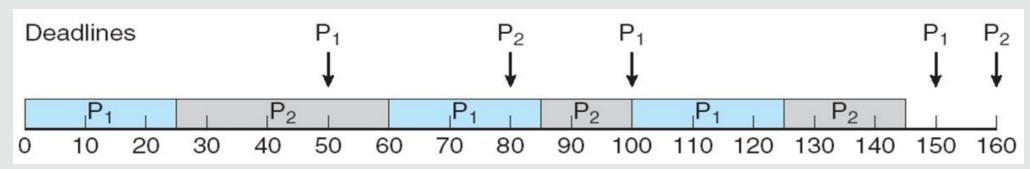
Scheduling policies can be flexibly designed and highly configurable

E.g. Multi-level Feedassignment Project Exam Help

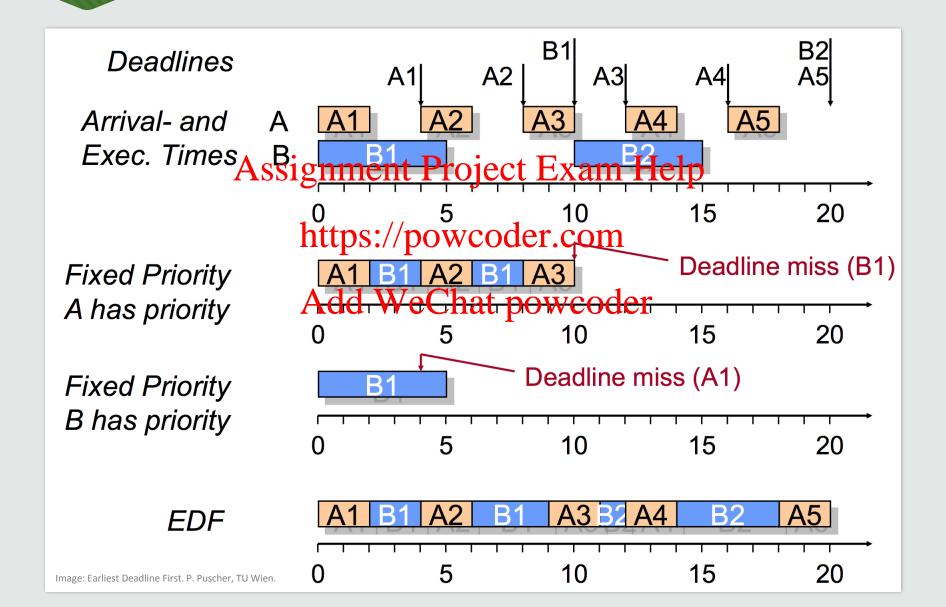
- O Number of levels in the scheduling system der.com
- O Methods for determining at which level a process is admitted
- O Methods for upgrading and demoring properties
- O Choice of scheduling algorithm at every level
 - O e.g. Round Robin
 - O time quantum parameter

Real-time Scheduling

- O Processes with periodic arrival times
- O Deadlines
- O Hard real-time: Guarantee that there are no deadline misses Assignment Project Exam Help
- O Soft real-time: Minimise number of deadline misses
- O Preemptive scheduling withttpstic BPWS996E-SPONties



Example: Earliest Deadline First



Real-time Scheduling

O A set of tasks is schedulable if and only if

$$\sum_{i} \frac{C_i}{T_i} \le 1$$

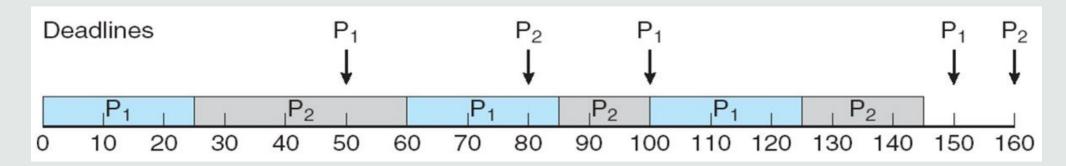
Assignment Project Exam Help

where

Worst-Case Execution time: t_i ://powcoder.com $\frac{-}{T} = \frac{-}{50} = 0.5$ Task deadline = period T_i Add WeChat powcoder

$$\frac{C_2}{T_2} = \frac{35}{80} = 0.4375$$

$$0.5 + 0.4375 < 1 \rightarrow \text{all good!}$$



Java Thread Scheduling

```
«interface»
    java.lang.Runnable
                       Assignment Project Exam Help
                             https://powcoder.com
       java.lang.Thread
+Thread()
                                   Creates an empty thread.
                              Add-WeChat powcoder
+Thread(task: Runnable)
+start(): void
                                    Starts the thread that causes the run() method to be invoked by the JVM.
+isAlive(): boolean
                                    Tests whether the thread is currently running.
+setPriority(p: int): void
                                    Sets priority p (ranging from 1 to 10) for this thread.
+join(): void
                                    Waits for this thread to finish.
+sleep(millis: long): void
                                    Puts a thread to sleep for a specified time in milliseconds.
+yield(): void
                                    Causes a thread to pause temporarily and allow other threads to execute.
+interrupt(): void
                                    Interrupts this thread.
```

Java Thread Scheduling

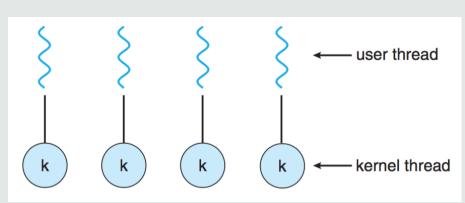
Scheduling policy

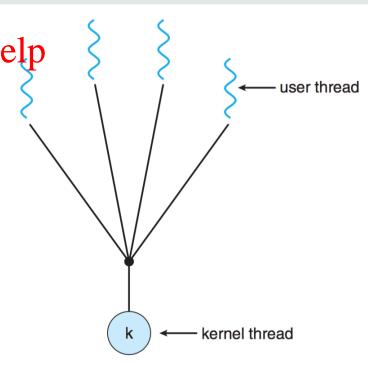
O Early Java versions: user threads map to one kernel threads project Exam Help FCFS + priorities, preemptive https://powcoder.com

O Today:

Depends on operating system,

user threads map 1:1 to kernel-threads





Java Thread Scheduling

Scheduler is run when

- O A thread terminates Assignment Project Exam Help
- O A higher priority thread becomes runnable (ready)
- O A thread calls yield() https://powcoder.com

Summary

Scheduling algorithms

- O Performance measurassignment Project Exam Help
- O Round-Robin scheduling
- O Multi-level queue scheduling https://powcoder.com
- O Feedback scheduling Add WeChat powcoder
- O Real-time scheduling
- O Java thread scheduling

Read

- O Tanenbaum & Bos., Modern Operating Systems
 - O Chapter 2.4

Assignment Project Exam Help

- O Silberschatz et al., Operatihttps://epowsouden.com
 - O Chapter 5

- O Further reading: Love. Linux Kernel Development:
 - O Chapter 4

Next Lecture

- O Introduction O Deadlocks
- O Operating System Architectures Assignment Project Exam Help
- O Processes O File Systems
- o Threads Programming https://powcederscombutput
- O Process Scheduling Evaluation WeCharpswerth end Virtualisation
- O Process Synchronisation