Operating Systems

Lecture 3b

Dr Ronald Grau   School of Engineering and Informatics   Spring term 2018

What is going to happen with operating systems during the strike?

- Lectures and Labs will keep running as usual

- Lectures will be recorded as usual, too – those who want to support their industrial action can do so without disadvantage

Image source: The Guardian

- If you want to help getting the current problems resolved as soon as possible, consider sending your suggestions to the University's Vice-Chancellor:

  Professor Adam Tickell: vc@sussex.ac.uk

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Previously

On programming with threads

- Parallel vs concurrent execution
- Data vs. Task vs. Pipeline parallelism
- Thread Safety
- Limits of parallelisation
- Hyperthreading
- Java thread library

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Scheduling

- Long/mid/short-term scheduling
- CPU vs I/O-bound processes
- Scheduling criteria
- Scheduling policies: FCFS, SJF, SRT, RR

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Benefits of multithreading

○ Responsiveness

○ Resource sharing (communication)

○ Economy (creation, context switching, . . .)
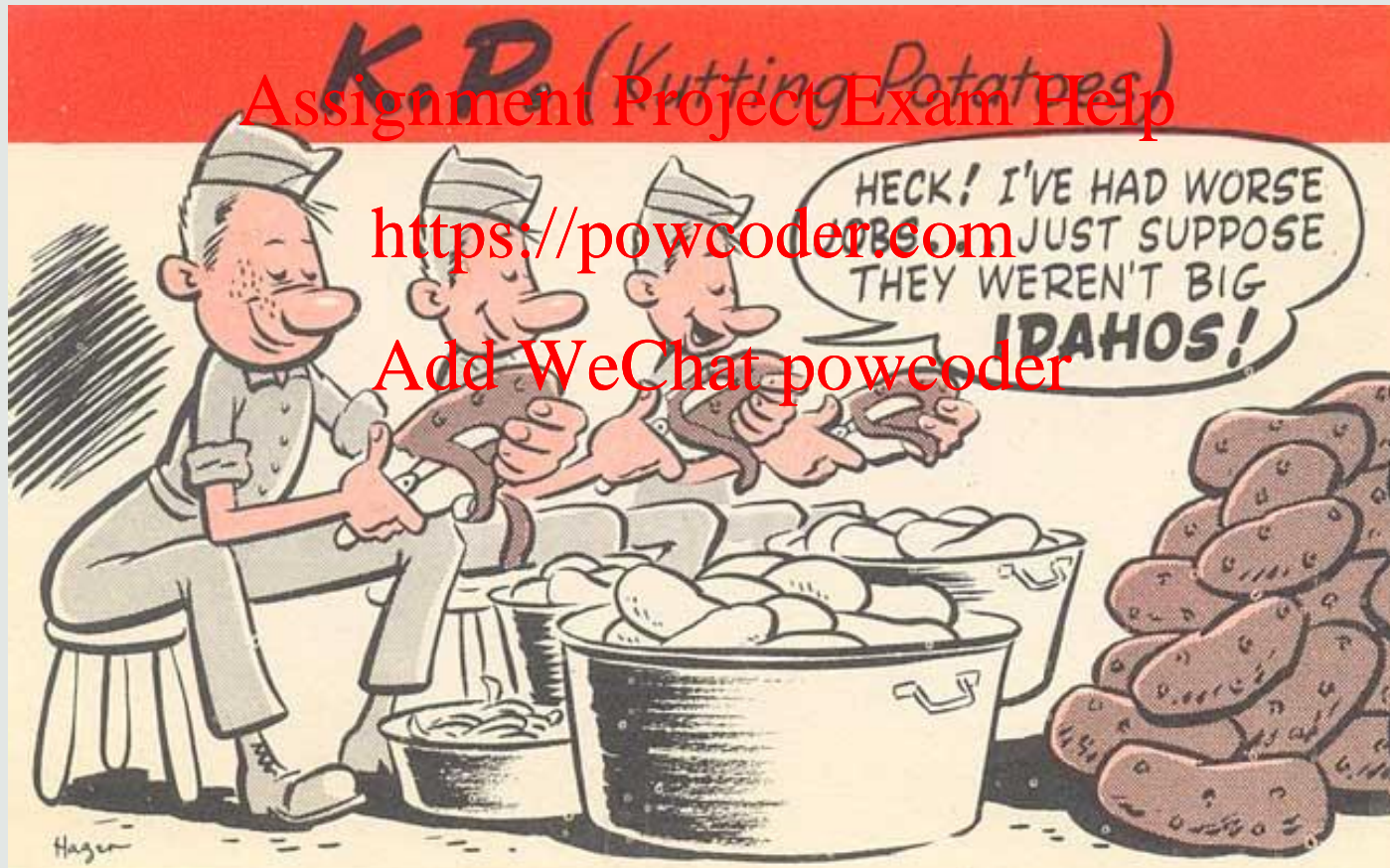
○ Scalability (parallel architectures)

Parallelism vs Concurrency

Patterns for parallelisation: pipelines, tasks, data splitting

What type of parallelism is this?

What type of parallelism is this?

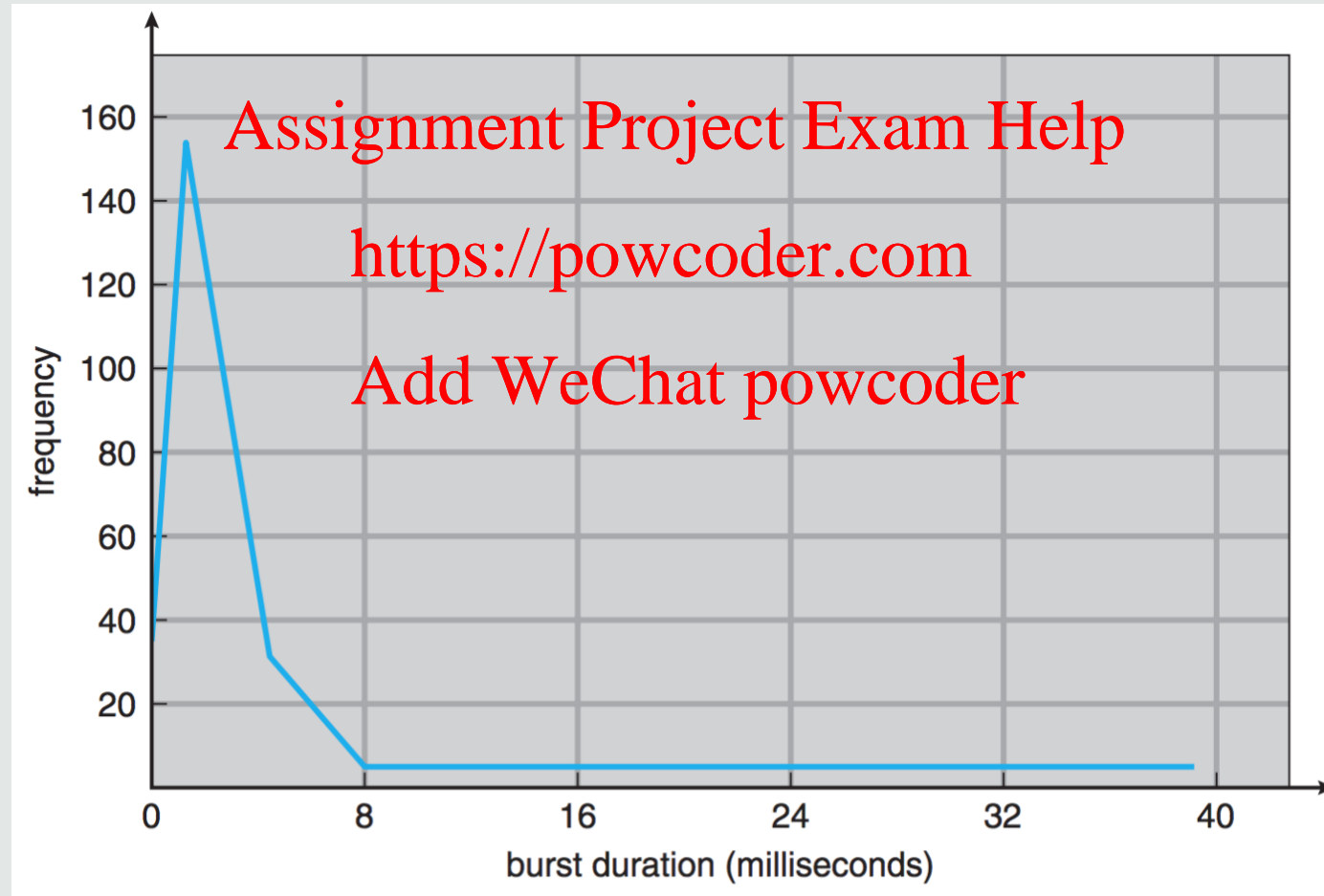What type of parallelism is this?

Determines the execution order of processes
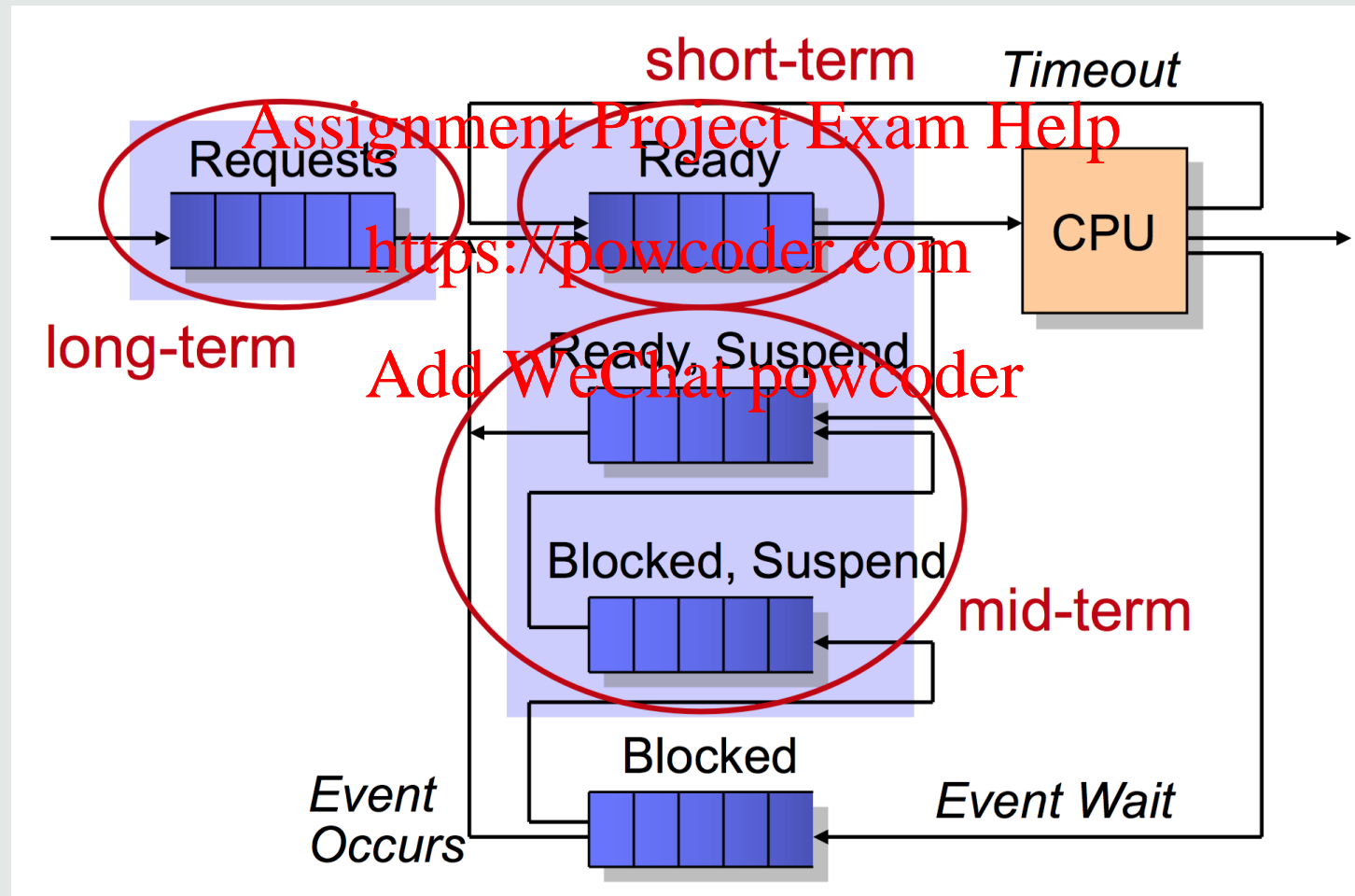
a) CPU-bound process   b) I/O-bound process
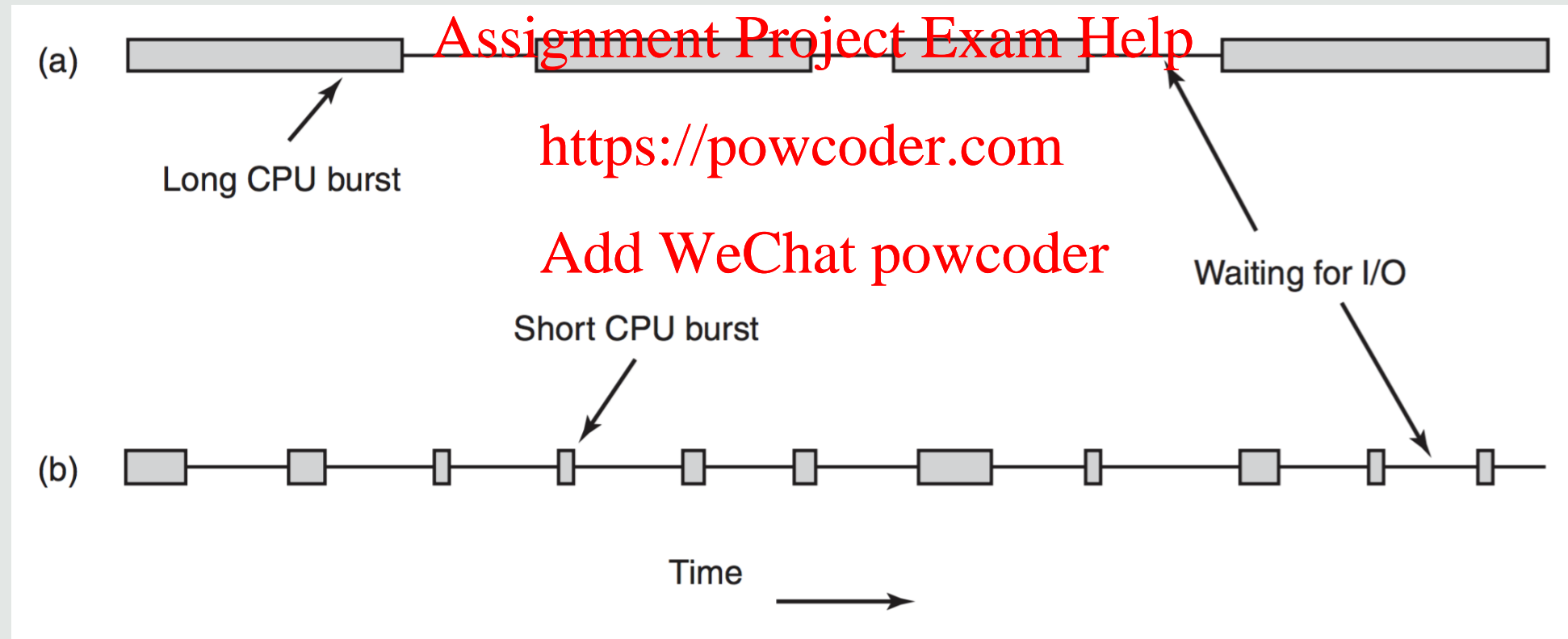
What kind of process is this?

○ Scheduler: selects process from Ready queue

○ Dispatcher: performs the context switch

Short-term Scheduling

- Scheduler: selects process from Ready queue

- Dispatcher: performs the context switch

  Assignment Project Exam Help

- When is scheduling happening?
  https://powcoder.com

  Add WeChat powcoder
  - After process creation
  - After ISR completion
  - A process blocks (I/O request, synchronisation, . . .)
  - At end of time slice
  - After process termination
  - On a yield system call (voluntary release of CPU)

- Non-preemptive

  The current process retains a hold over the CPU until returning control

  Assignment Project Exam Help
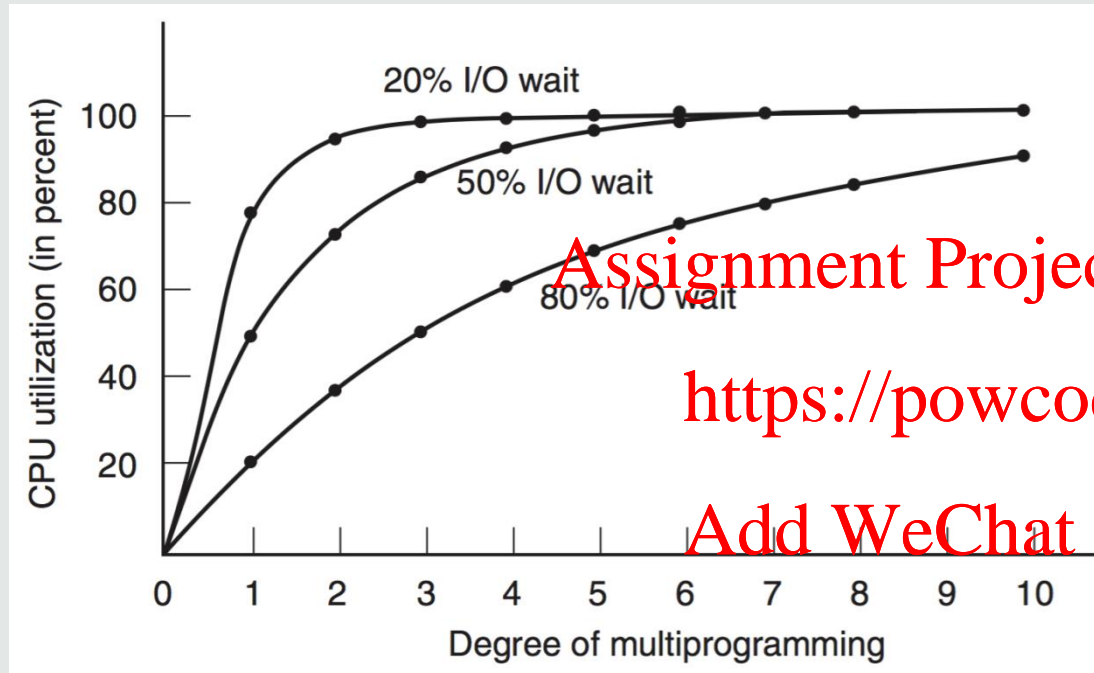
- Preemptive

  https://powcoder.com

  Scheduler takes control of the CPU even though the current process could continue executing Add WeChat powcoder

utilisation = $1 - p^n$

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- p is the fraction of time processes spend waiting for I/O
- n is the number of processes
- This model assumes independent processes.

Scheduling Criteria

- **CPU utilisation** ("Load"): Percentage of time the CPU is busy

- **Throughput**: Processes per second handled

- **Turnaround time**: Time from submission to completion

- **Waiting time**: Time processes spend in the Ready queue

- **Response time**: Time from submission to the first response (Interactive systems)

- **Meeting deadlines**: Operating within required time constraints (real-time systems)

- **Predictability**: Avoiding erratic behaviour that needs frequent correction

- **Fairness**: Every process gets a turn.

- **Balance**: All system components well-used, not just the CPU.

- **Policy enforcement**: Some things are more important than others (e.g. make sure critical processes can run when needed)

Which criteria are most important depends on the kind of system used

Scheduling Algorithms

- Batch Systems
  - First-Come First-Served
  - Shortest Job First
  - Shortest Remaining Time Next

- Interactive Systems
  - Round Robin
  - Priority Scheduling
  - Feedback Scheduling

- Real-Time Systems
  - Earliest Deadline First
  - Rate Monotonic Scheduling

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

… and many more!

First-Come First-Served (FCFS)

Process	Burst time
P1	24
P2	3
P3	3

+ Simple implementation
(FIFO queue)

- Long average waiting times

P1			P2	P3
0 | | 24 | 27 | 30

P2	P3	P1	
0 | 3 | 6 | 30
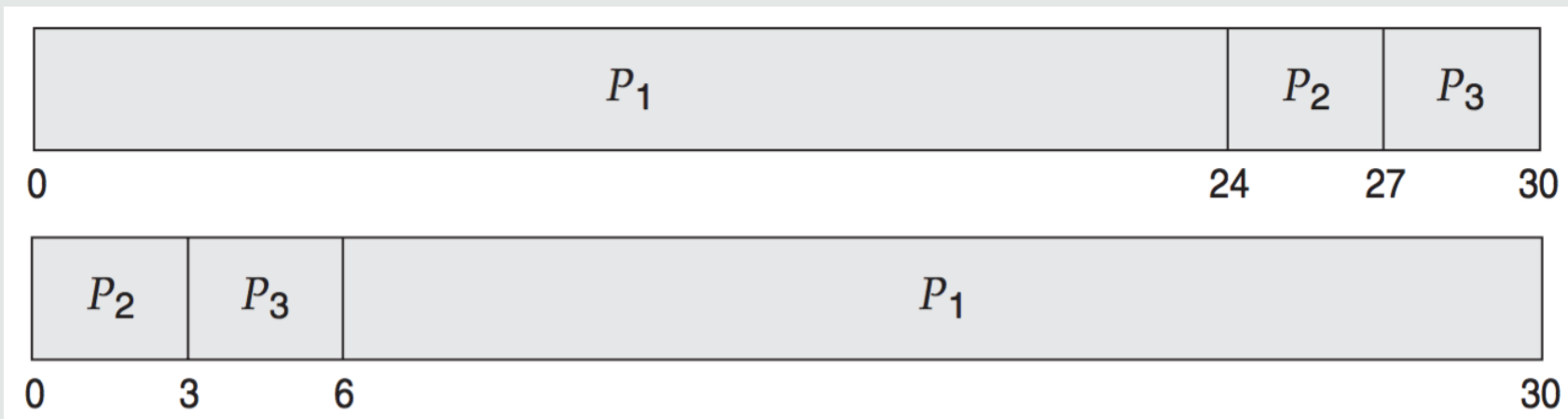
Shortest Job First (SJF)

Process	Burst time
P1	6
P2	8
P3	7
P4	3

„Shortest CPU burst first"

+ is optimal: minimal average waiting time

How do we know the length of the
next CPU burst of a process?

Requires good predictability!
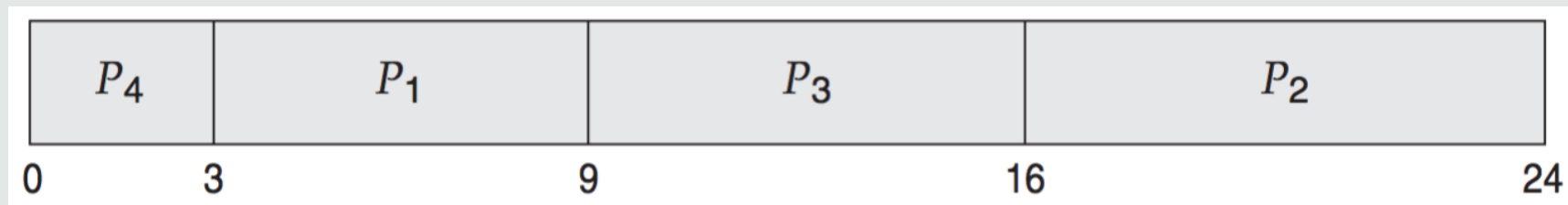
P4	P1	P3	P2
0 3	9	16	24
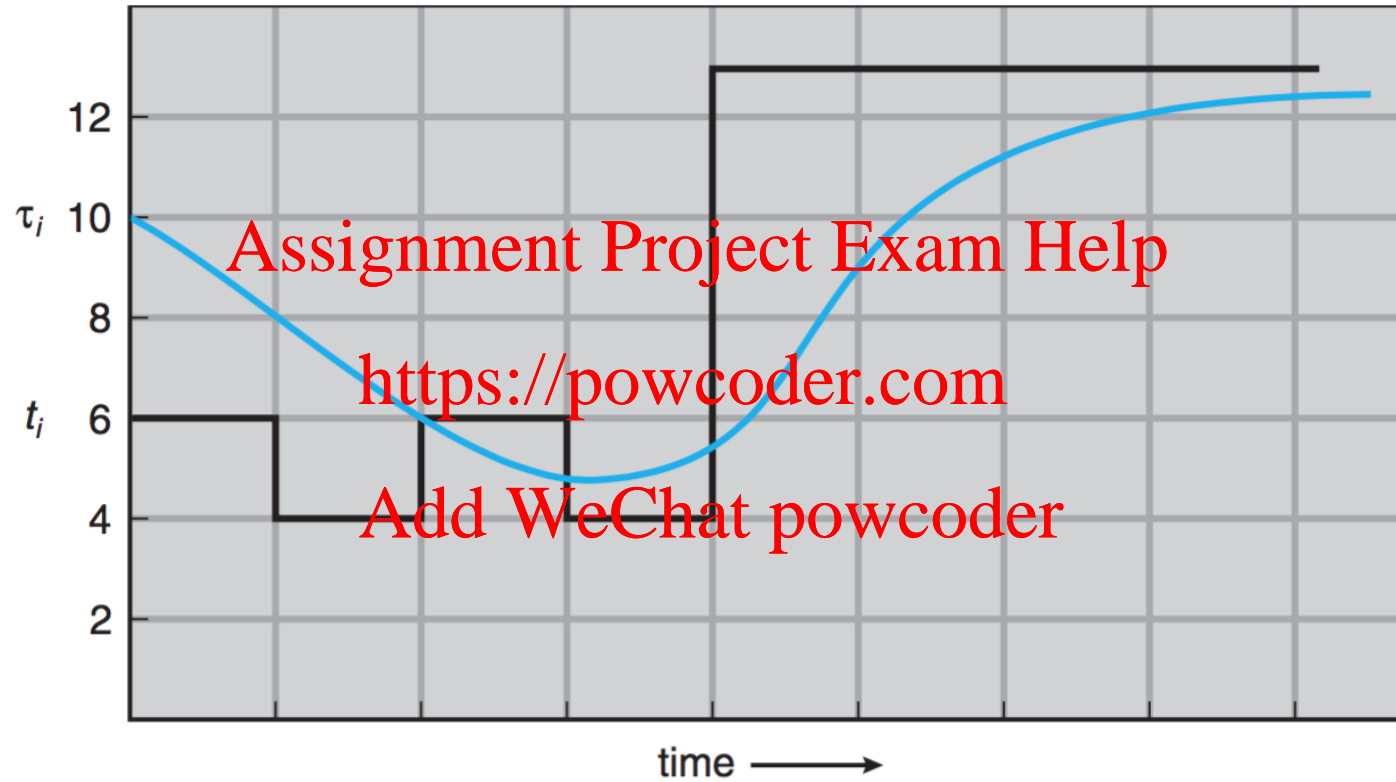
CPU burst (t_i)	6	4	6	4	13	13	13	...
"guess" (τ_i)	10	8	6	6	5	9	11	12	...

CPU burst (t_i)   6   4   6   4   13   13   13   ...
"guess" (τ_i) 10   8   6   6   5   9   11   12   ...
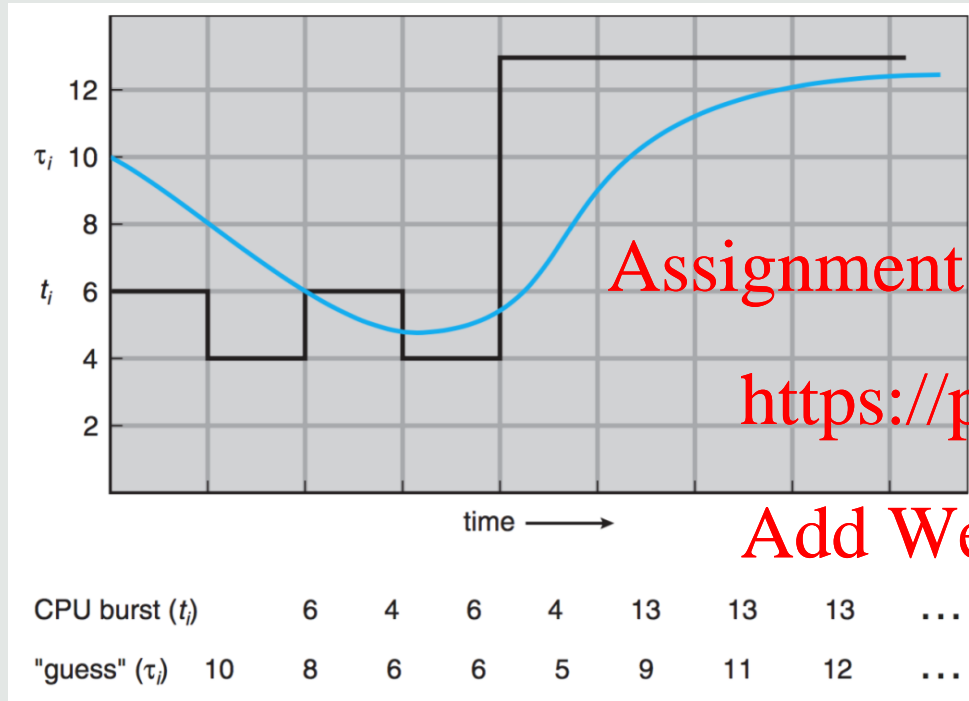
$$T_{n+1} = \alpha * t_n + (1 - \alpha) * T_n$$

- t_n is the (known) length of the most recent burst
- T_{n+1} is the prediction of the next burst
- T_0 is initialised e.g. to average expected burst length
- $\alpha \in [0, 1]$ gives weight to the history, e.g. 1/2

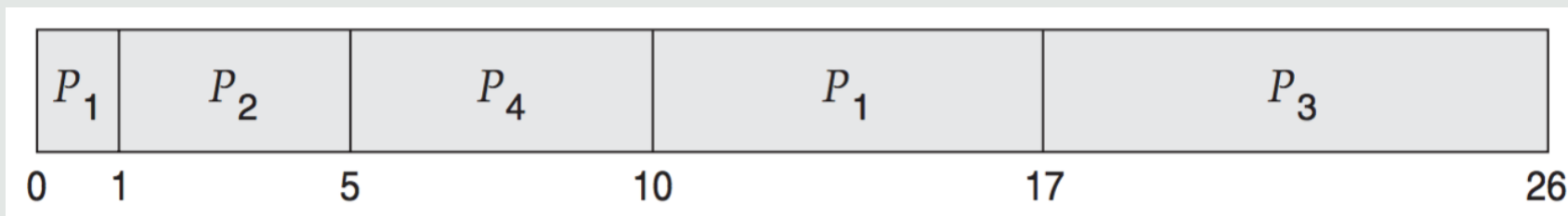| Process	Arrival time	Burst time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

Preemptive version of SJF

P_1	P_2	P_4	P_1	P_3
0 1	5	10	17	26

Priority Scheduling

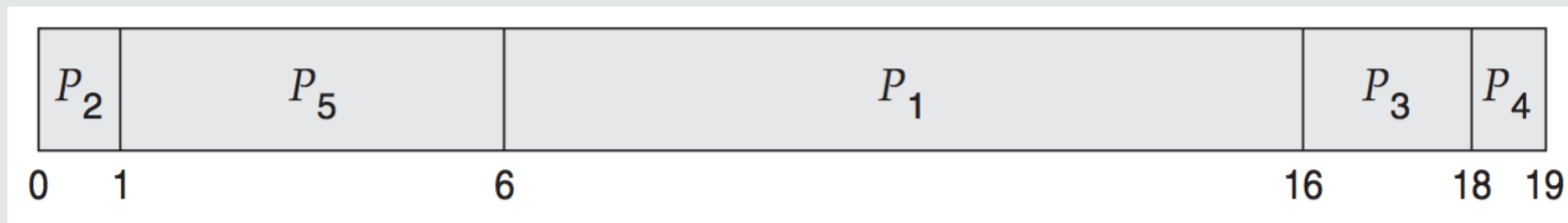| Process	Priority	Burst time
P1	3	10
P2	1	1
P3	4	2
P4	5	1
P5	2	5

Lowest number is highest priority

Processes with same
priority scheduled FCFS

Implementation:
Priority queue

Starvation possible, i.e. indefinite waiting

→ Aging: gradually increase priority

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

P_2	P_5	P_1	P_3	P_4
0 1	6	16	18	19

Process	Burst time
P1	24
P2	3
P3	3

FCFS + preemption at end of time slice (time quantum)

Implementation: FIFO queue + Timer interrupt
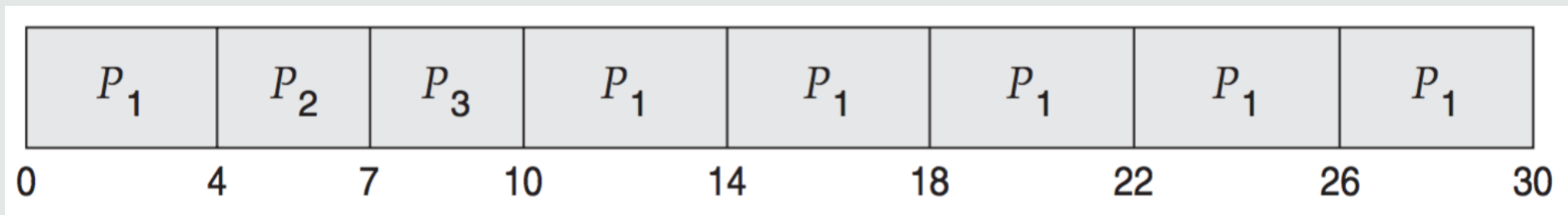
Time quantum: 4ms

process	time
P_1	6
P_2	3
P_3	1
P_4	7

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder
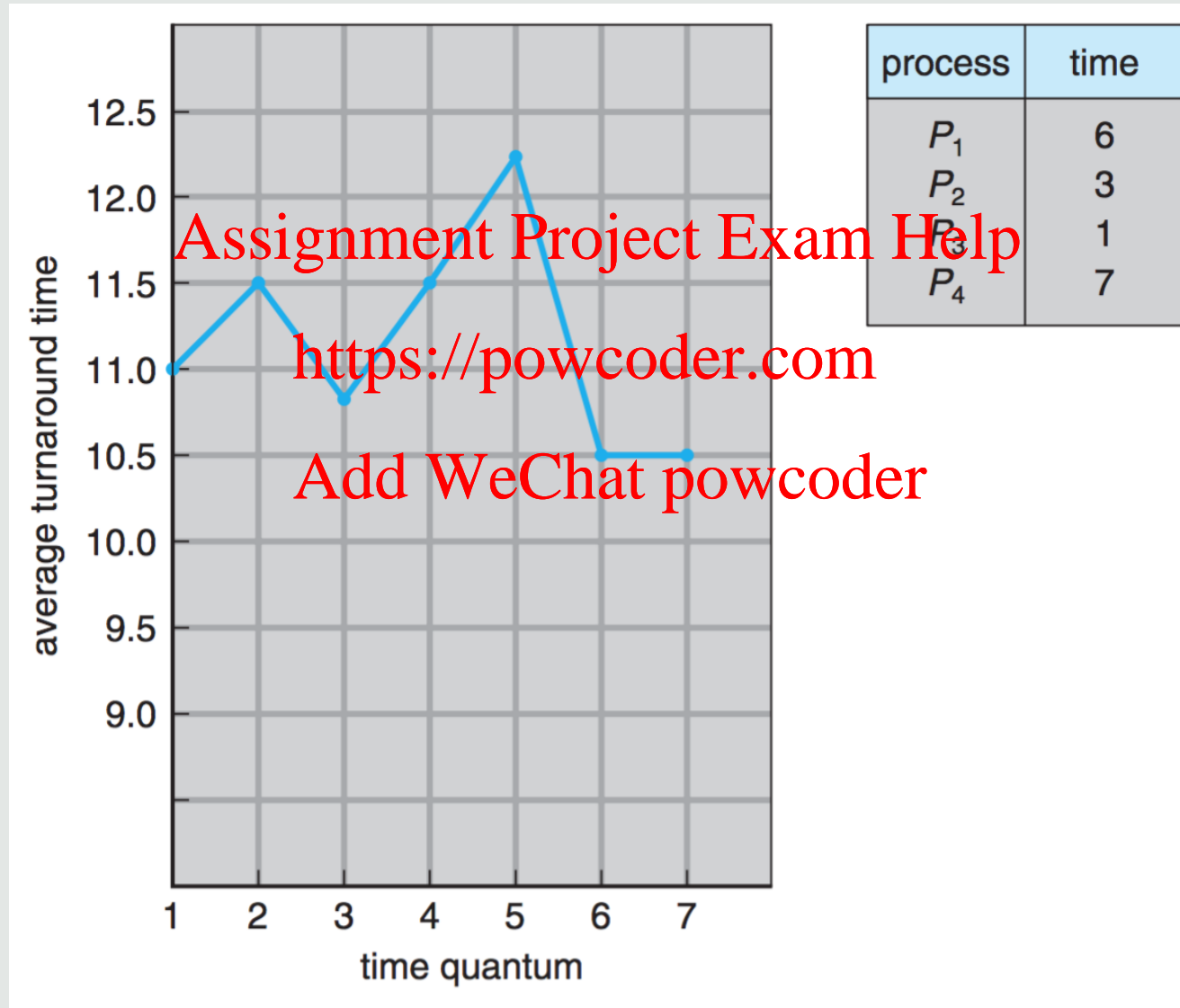
Scheduling

- Long/mid/short-term scheduling
- Scheduler and dispatcher
- CPU vs I/O-bound processes
- Scheduling criteria
- Scheduling policies: FCFS, SJF, SRT, RR

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

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

Assignment Project Exam Help

- Silberschatz et al., Operating System Concepts

  https://powcoder.com

  - Chapter 5

  Add WeChat powcoder

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

- Introduction
- Operating System Architectures
- Processes
- Threads - Programming
- **Process Scheduling (continued)**
- Process Synchronisation

- Deadlocks
- Memory Management
- File Systems
- Input / Output
- Security and Virtualisation

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder