Comp 346 – Fall 2019

Tutorial 8

CPU Scheduling

Why do we need scheduling?

- To manage processes according to requirements of a system, like:
  - User responsiveness or
  - Throughput
- **Performance of a scheduler** is determined mainly by:
  - Context switch time
  - Scheduling policy

Université Concordia University

Context Switch!

- **Switching from one process running on the CPU to another process**:
- Saves all the registers of **outgoing** process (**to memory**), then loads all the registers of **incoming** process (**from memory**)
- Can be time-costly; mostly hardware-dependent

UNIVERSITÉ Concordia UNIVERSITY

SCHEDULING

- **The mechanism that determines <u>when</u> the CPU will be allocated to processes, and in <u>what</u> order.**

- Two classes of scheduling strategies:
  - **Non-preemptive**
  - **Preemptive**

NON-PREEMPTIVE POLICIES

- Allow any process **to run to completion** once it has been allocated to the CPU. **Current process does not get interrupted.**
- Some examples:
  - First Come First Serve (**FCFS**)
  - Shortest Job First (**SJN**)
  - **Priority** scheduling

UNIVERSITÉ Concordia UNIVERSITY

Preemptive policies

- **Allow another process to interrupt current process** if:
  - It has **a higher priority**
  - The **time quantum** has elapsed
- Some examples:
  - **Round Robin**
  - **Multiple-level Queues**

Scheduling Criteria

- **CPU utilization** – keep the CPU as busy as possible
- **Throughput** – number of processes that complete their execution per time unit
- **Turnaround time** – amount of time to execute a particular process
- **Waiting time** – amount of time a process has been waiting in the ready queue
- **Response time** – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment)

Scheduling examples

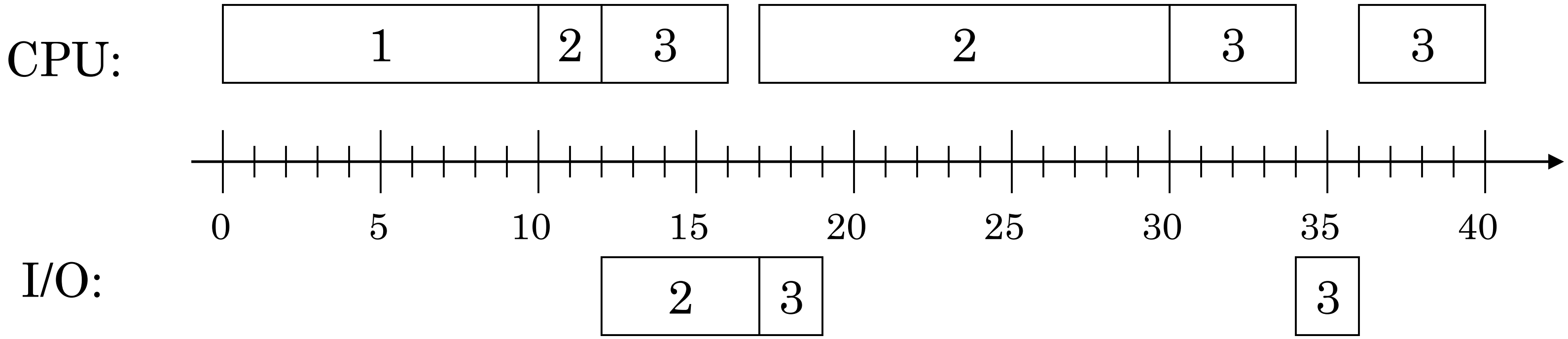
- Given **three threads**, their execution times and I/O needs, apply scheduling policies:
- Threads are placed on ready queue in order: T1, T2 then T3

- Specific to Round Robin:
  - Time Quantum of 3ms.
  - Context switch time considered negligible in this example.

UNIVERSITÉ Concordia UNIVERSITY

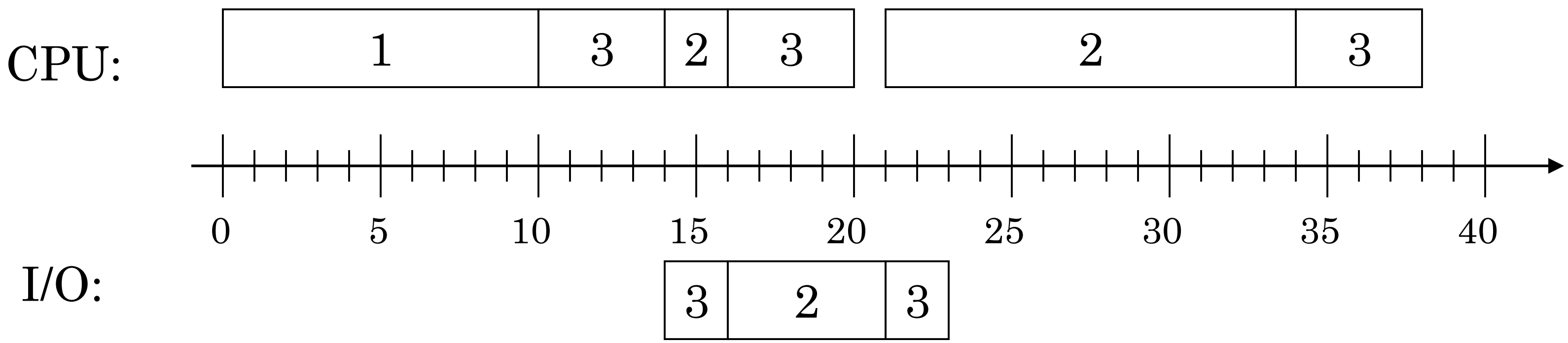
First Come First Serve (FCFS)

Thread	CPU	1^{st} I/O	2^{nd} I/O
T1	10 ms	n/a	n/a
T2	15 ms	at 2ms for 5ms	n/a
T3	12 ms	at 4ms for 2ms	at 8ms for 2ms



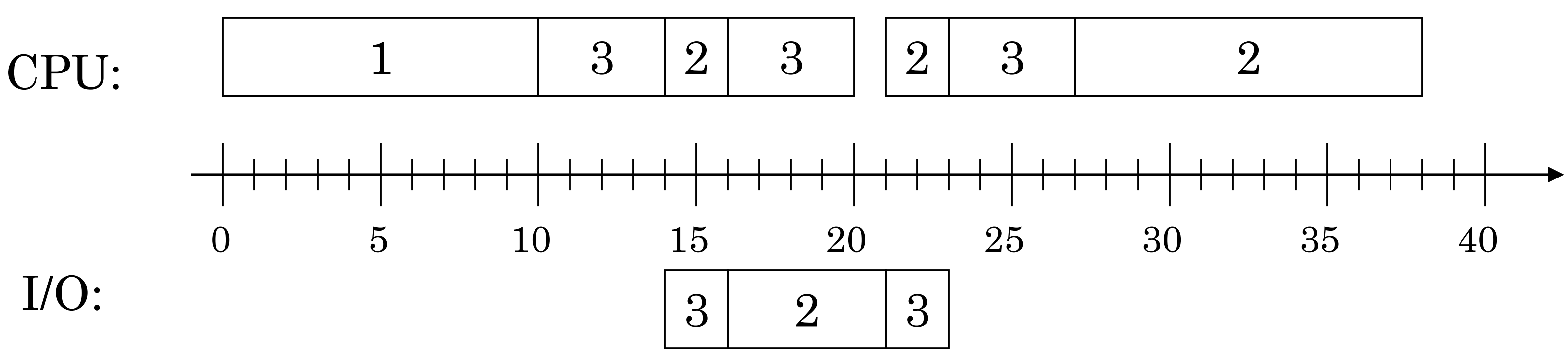
SHORTEST JOB FIRST (NONPREEMPTIVE)

Thread	CPU	1st I/O	2nd I/O
T1	10 ms	n/a	n/a
T2	15 ms	at 2ms for 5ms	n/a
T3	12 ms	at 4ms for 2ms	at 8ms for 2ms



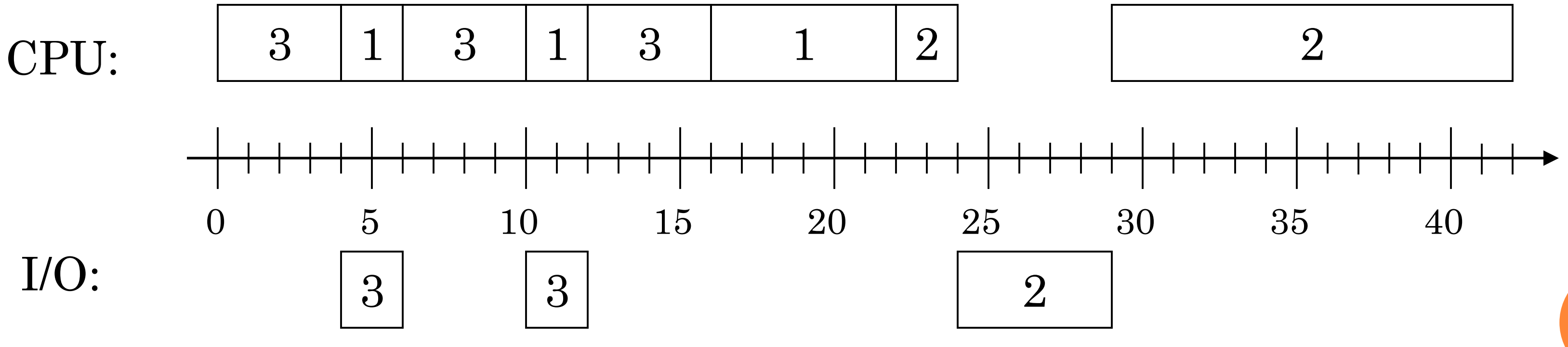
Shortest Job First (Preemptive)

Thread	CPU	1^{st} I/O	2^{nd} I/O
T1	10 ms	n/a	n/a
T2	15 ms	at 2ms for 5ms	n/a
T3	12 ms	at 4ms for 2ms	at 8ms for 2ms



PRIORITY (PREEMPTIVE)

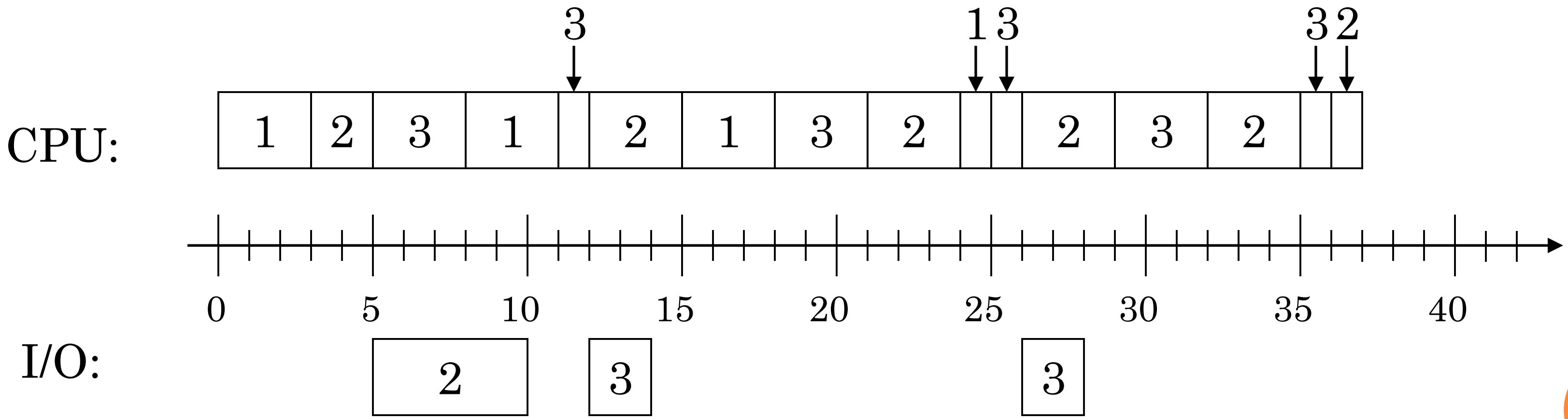
Thread	Prio	CPU	1^{st} I/O	2^{nd} I/O
T1	2	10 ms	n/a	n/a
T2	3	15 ms	at 2ms for 5ms	n/a
T3	1	12 ms	at 4ms for 2ms	at 8ms for 2ms



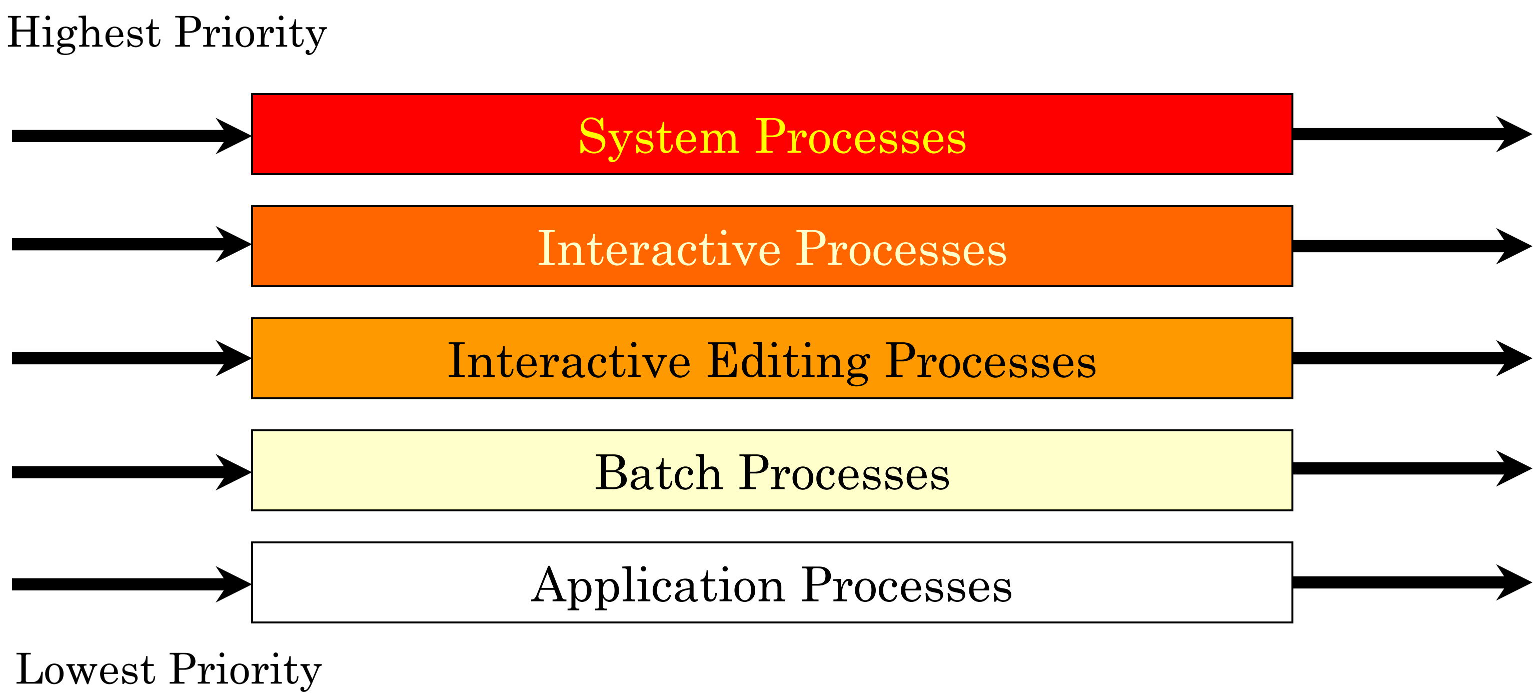
UNIVERSITÉ Concordia UNIVERSITY

Round Robin (3ms)

Thread	CPU	1^{st} I/O	2^{nd} I/O
T1	10 ms	n/a	n/a
T2	15 ms	at 2ms for 5ms	n/a
T3	12 ms	at 4ms for 2ms	at 8ms for 2ms



MULTILEVEL QUEUE (2)



Multilevel Feedback Queue (1)

- Another multilevel queuing strategy allows processes to change their priorities over time, thus, changing levels of the queue.
- E.g.: an interactive editing process might become compute intensive, so it's priority may change to higher.
- A strategy, that allows processes change queues is called **multilevel feedback queue**. (Used in BSD UNIX, for example).

Université Concordia University

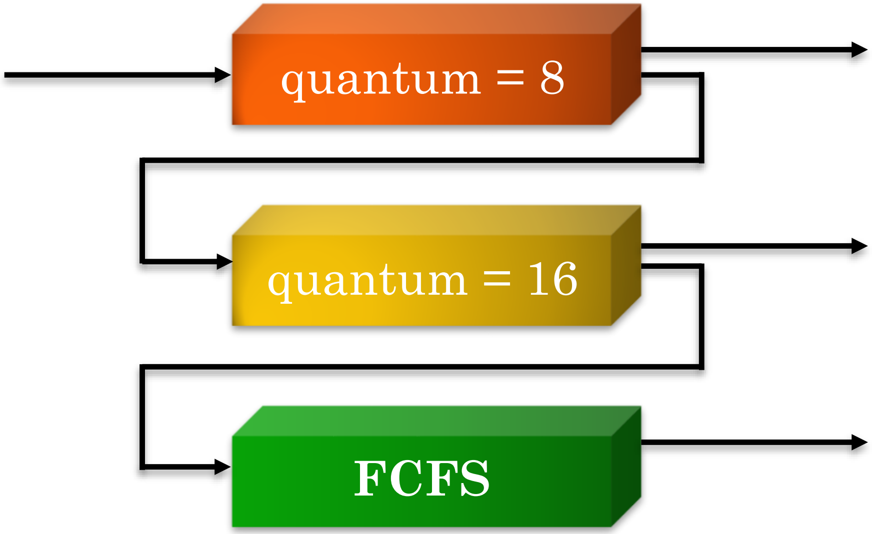
Multilevel Feedback Queue (2)

- There is a notion of **external priority**, that, in addition to the main priority, influence process's queue.
- It's denoted by a nice value of process. Nice value decreases if a process used CPU for many times during last N time units, and increases if the process hasn't run for relatively long time, making scheduling more fair.
- The scheduler recomputes processes' priorities once per quantum.
- Different quantum lengths on different queue levels attempt to shorten execution of short jobs, make it more fair for longer jobs and reducing frequency of context switching at the same time.

Université Concordia University

Multilevel Feedback Queue (3)

- Multilevel feedback queues.



Problem-Solving 1 (take home exercise) → Solution will be provide by next session

- Determine the throughput for the following scenario given a Round Robin scheduling algorithm with quantum time = 3 ms, with context switching time = 1 ms

Thread	CPU	1^{st} I/O	2^{nd} I/O
T1	10 ms	n/a	n/a
T2	15 ms	at 2ms for 5ms	n/a
T3	12 ms	at 4ms for 2ms	at 8ms for 2ms

UNIVERSITÉ Concordia UNIVERSITY

Problem-Solving 2

- Consider a system having the following characteristics :
  i. 2 parallel CPUs.
  ii. 1 ready queue that is shared by both CPUs.
  iii. Round-robin scheduling is used with a time quantum of 5 units.
  iv. Context switching time is infinitely small (i.e. 0).
  v. 2 instances of an I/O device that is used by all the processes for their I/O operations.
  vi. A load of 4 processes as described in table 1.

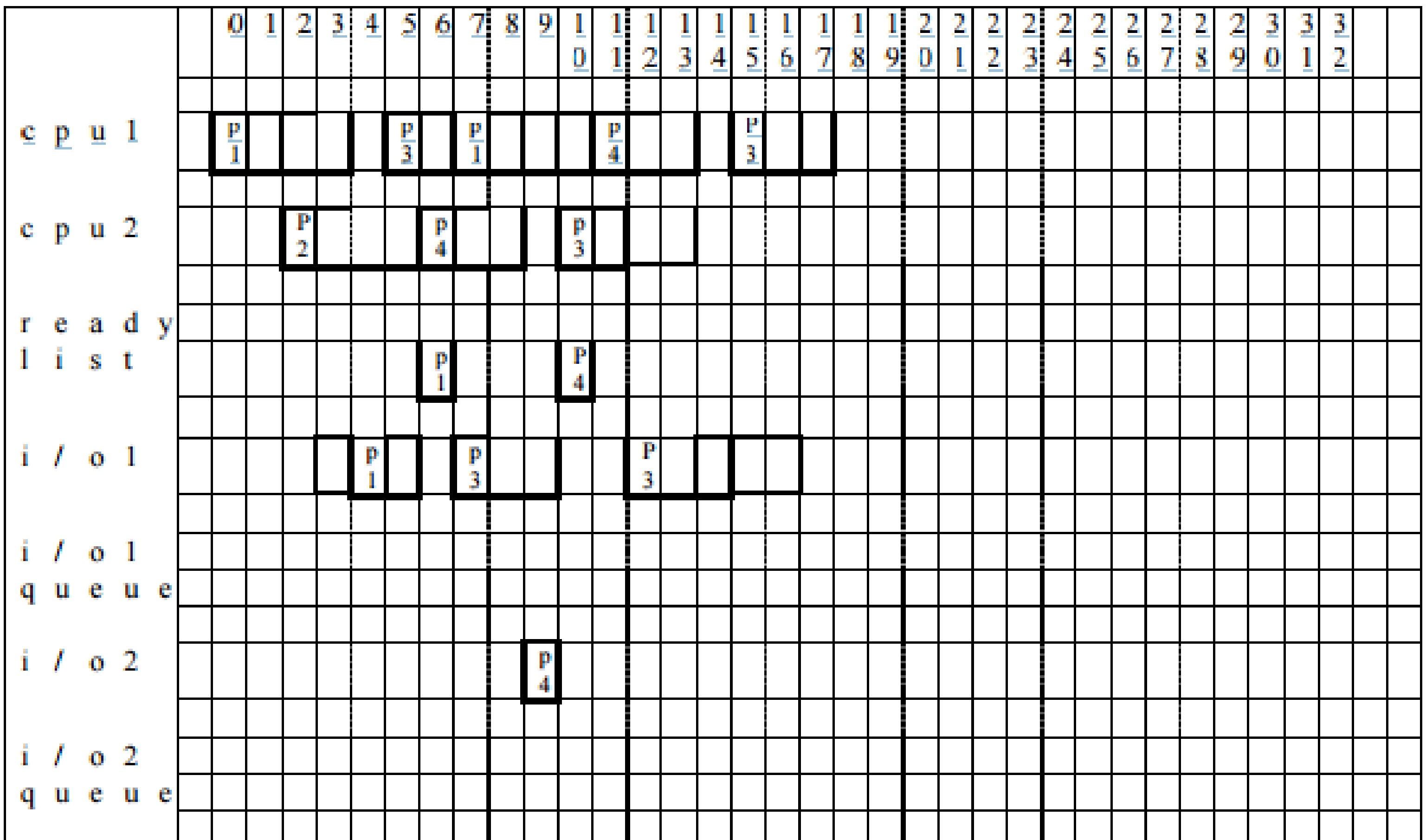
Process	Arrival time	Service time	Duration of each I/O phase	I/O periods
P1	0	8	2	4
P2	2	4	0	-
P3	5	7	3	2, 4
P4	6	6	1	3

- **Table 1 – Process load**

Université Concordia University

Problem-Solving 2 - Solution

(i) Draw a Gantt chart. Show the 2 CPUs, the ready queue, each I/O device and its associated waiting queue.



Problem-Solving 2 - Solution

(ii) Calculate the average turnaround time.

- $T_t = ((11\text{-}0) + (6\text{-}2) + (18\text{-}5) + (14\text{-}6)) / 4 = (11 + 4 + 13 + 8) / 4 = 36/4$
  $= 9$ *time units*

(iii) Calculate the average waiting time.

- $T_w = (1 + 0 + 0 + 1) / 4 = 2/4 = 0.5$ *time units*

(iv) Calculate the CPU utilization percentage for the total time period.

- $U_{CPU1} = (16/18)\times 100 = 88.88\%$
- $U_{CPU2} = (9/18)\times 100 = 50\%$

(v) Calculate the throughput for the total time period.

- $T_{tp} = 4/18 = 0.22$

Université Concordia University

Problem-Solving 3

- The procedure ContextSwitch is called whenever there is a switch in context from a running program A to another program B. The procedure is a straight-forward assembly language routine that saves and restores registers, and must be atomic (i.e., unbreakable). Something disastrous can happen if the routine ContextSwitch is not atomic.

  a) Explain why ContextSwitch must be atomic, possibly with an example.
  b) Explain how the atomicity can be achieved in practice (Hint: It has something to do with interrupts).

PROBLEM-SOLVING 3 (SOLUTION)

a) *If the context switch routine is interrupted then the system may become unstable because the next process may resume with an inconsistent state. Context switch must be atomic in order to avoid the system instability.*
b) *Atomicity is achieved mostly by disabling the interrupts.*

UNIVERSITÉ Concordia UNIVERSITY

PROBLEM-SOLVING 4

- Consider a system having the following characteristics :
  - 2 parallel CPUs.
  - 1 ready queue that is shared by both CPUs.
  - Round-robin scheduling is used with a time quantum of 5 units.
  - Context switching time is infinitely small (i.e. 0).
  - No I/O operations.
  - A load of 4 processes as described in table 1.

Process	Arrival time	Service time	Duration of each I/O phase	I/O periods
P1	0	8	0	0
P2	2	4	0	0
P3	5	7	0	0
P4	6	6	0	0

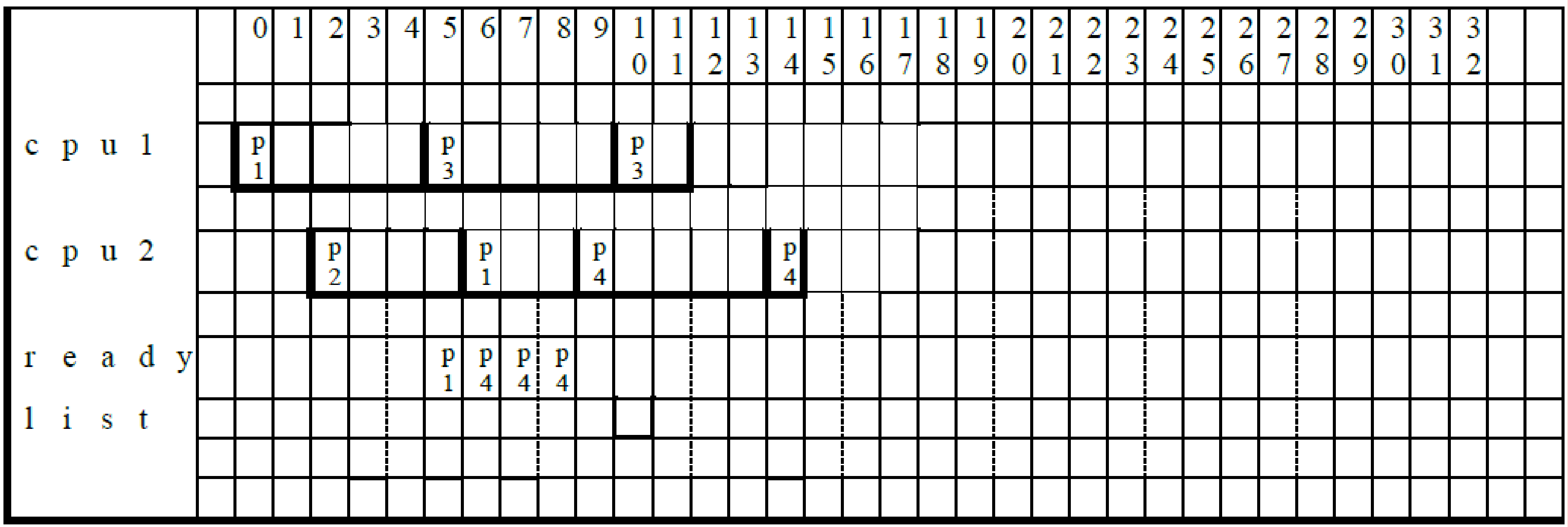
- **Table 1 – Process load**

PROBLEM-SOLVING 4

i. Draw a Gantt chart. Show the 2 CPUs and the ready queue.
ii. Calculate the average turnaround time.
iii. Calculate the average waiting time.
iv. Calculate the CPU utilization percentage for the total time period.
v. Calculate the throughput for the total time period.
vi. Suppose we had a single CPU, would the overall performance be worst, same, or better? Explain for each case (i through v) above.

PROBLEM-SOLVING 4 (SOLUTION)

(i)



UNIVERSITÉ Concordia UNIVERSITY

Problem-Solving 4 (Solution)

(ii) Calculate the average turnaround time.

- $Tt = ((9\text{-}0) + (6\text{-}2) + (12\text{-}5) + (15\text{-}6)) \ / \ 4 = (9 + 4 + 7 + 9) \ / \ 4 = 29/4 = 7.25$

(iii) Calculate the average waiting time.

- $Tw = (1 + 0 + 0 + 3) \ / \ 4 = 4/4 = 1.0$

(iv) Calculate the CPU utilization percentage for the total time period.

- $Tcpu1 = 12/15 = 80\%$ and $Tcpu2 = 13/15 = 86.6\%$
- $Tcpu = 25/30 = 0.83.33\%$

- (v) Calculate the throughput for the total time period.
- $Ttp = 4/15 = 0.26$

Université Concordia University

Problem-Solving 4 (Solution)

- (vi)

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
cpu		1	1	1	1	1	2	2	2	**2**	1	1	**1**	3	3	3	3	3	4	4	4	4	4	3	**3**	**4**								
ready				2			1	1	1	1	3	3	3	4	4	4	4	4	3	3	3	3	3	4	4									
list							3	3	3	3	4	4	4																					
								4	4	4																								

Université Concordia University

PROBLEM-SOLVING 4 (SOLUTION)

(vi)

- *The average turnaround time would be (12+7+19+19)/4 = 57/4 = 19.25.*
- *The average waiting time would be (4+1+12+13)/4 = 30/4 = 7.5.*
- *The percent cpu utilization would be 25/25 = 100%.*
- *The throughput would be 4/25 = 0.16.*

UNIVERSITÉ Concordia UNIVERSITY

References

- **Previous Tutorial Materials** (By Mr. Antonio Maiorano & Mr. Paul Di Marco).
- A. Silbershatz, P.B. Galvin, G. Gagne: Operating System Concepts, 9th Ed. John Wiley & Sons, Inc. 2013.
- Problem solving questions by François Gingras Myriam Kharma and Kerly Titus