Operating Systems W4L2 - Scheduling and Race Conditions

Class	Operating Systems
Created	@Sep 23, 2020 4:20 PM
Materials	04 - Deadlocks.pdf
Reviewed	☐
Type	Lecture

Scheduling

- We started by wrapping up last class's discussion

▼ What is preemptive scheduling?

OS can stop a process to run another process

▼ What is the most common? Why?

Interactive, because managing user input and user interaction is the average use for an OS

▼ Goals of the 4 different scheduling categories

All systems

Fairness - giving each process a fair share of the CPU
Policy enforcement - seeing that stated policy is carried out
Balance - keeping all parts of the system busy

Batch systems

Throughput - maximize jobs per hour
Turnaround time - minimize time between submission and termination
CPU utilization - keep the CPU busy all the time

Interactive systems

Response time - respond to requests quickly
Proportionality - meet users' expectations

Real-time systems

Meeting deadlines - avoid losing data

See slides on the course site for all the algorithms and their strategies. Also elaborated upon in the textbook.

▼ Batch System Algorithm Diagrams

Scheduling in Batch Systems: First-Come First-Served

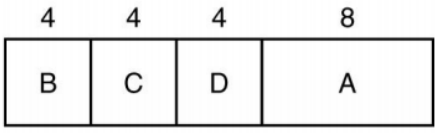
- Non-preemptive
- Processes ordered as queue
- A new process added to the end of the queue
- A blocked process that becomes ready added to the end of the queue
- Main disadv: Can hurt I/O bound processes

Scheduling in Batch Systems: Shortest Job First

- Non-preemtive
- Assumes runtime is known in advance
- Is only optimal when all the jobs are available simultaneously

(a)

Run in original order



(b)

Run in shortest job first

Scheduling in Batch Systems: Shortest Remaining Time Next

- Preemptive
- Scheduler always chooses the process whose remaining time is the shortest.
- Runtime must be known in advance.

▼ Interactive System Algorithm Diagrams

Scheduling in Interactive Systems: Round-Robin

- Each process is assigned a time interval: quantum (or time slice)
- After this quantum, the CPU is given to another process
- What is the length of this quantum?
  - too short -> too many context switches -> lower CPU efficiency
  - too long -> poor response to short interactive

Scheduling in Interactive Systems: Priority Scheduling

- Each process is assigned a priority.
- Ready process with the highest priority is allowed to run.
- Priorities are assigned statically or dynamically.
- Must not allow a process to run forever
  - Can decrease the priority of the currently running process.
  - Use time quantum for each process.

▼ Real-Time Scheduling

Scheduling in Real-Time

- Process must respond to an event within a deadline.
- Hard real-time vs soft real-time
  - Hard: Result/Response becomes incorrect if you miss the deadline. Used in critical applications like medical, air-traffic control, … .
  - Soft: If deadline is missed, system is still correct but with degraded performance. Example: computer games.
- Periodic vs aperiodic events
- Processes must be schedulable
- Scheduling algorithms can be static or dynamic

- The operating system's main job is dealing with processes, and everything can be applied to threads in the same way
    - However, it depends on whether the OS manages threads or the process is managing them

---

Race Conditions

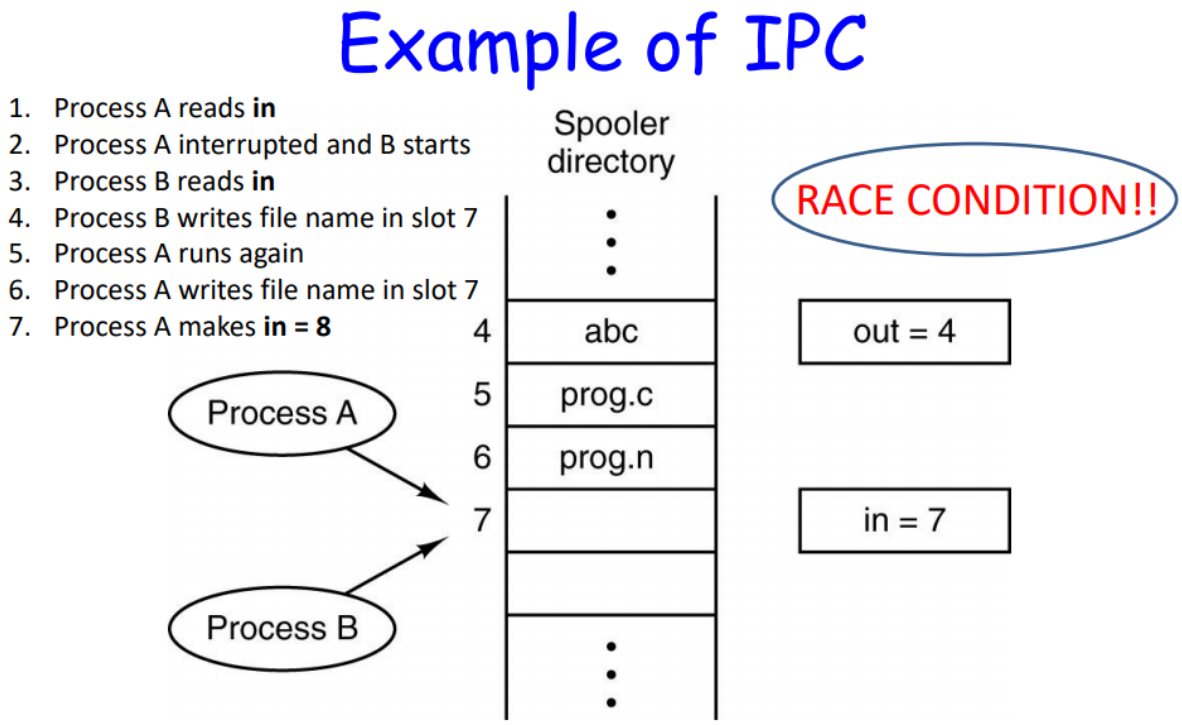
Interprocess Communication (IPC)

- Frequently, processes need to talk to each other, but they don't share memory
    - Each process is also undet the assumption it is the only process running (due to virtualization)

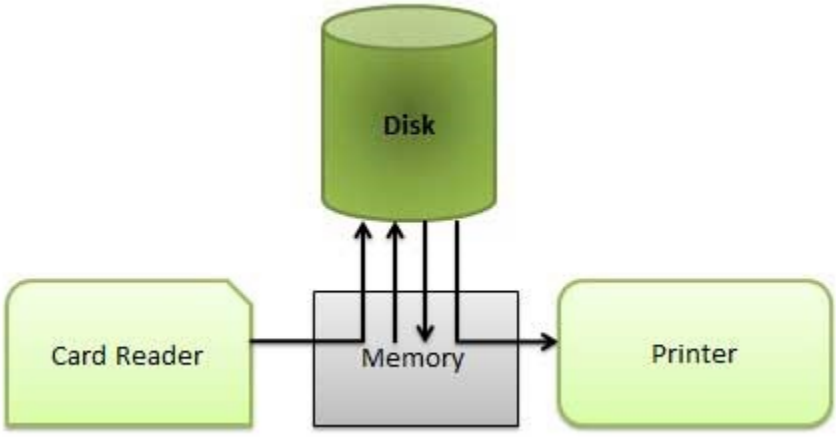
▼ What are the 3 main issues?

1. How can info be passed amongst processes?
2. Ensure two or more processes don't clash
3. Ensure proper sequencing (for scheduling) in case of any dependencies, such as a process needing another process

▼ IPC Diagram



▼ **Spooling:** A term that refers to a queue going to an I/O device



- A library will either be added...
  1. Via static linking
  2. When the executable runs and becomes a process
- Race conditions are when you don't know the order of particular operations, i.e. Process A and Process B both writing to block 7 (see above diagram) or forked

processes printing something

Avoiding Race Conditions

- **Mutual Exclusion:** Prohibit more than one process reading and writing to one part of memory
- The choice of *primitive operations* is key

▼ When two processes are attempting to read and write to one part of memory, what is that part of memory called?

The **critical region**