

# Process Review

Instructor: Wei Feng



# Questions?

- What is a process?
  - A process is an **instance** of a running program. It contains the program code and its current activity, such as **registers, variables, program counter, input, output and a state.**



# Questions?

- What is a thread?
  - A thread is **the entity within a process** that can be scheduled for execution.

# Questions?

- What is the difference between a process and a thread?
  - Threads are used for **'lightweight'** tasks, whereas processes are used for more **'heavyweight'** tasks.
  - The typical difference is that threads (of the same process) run in **a shared memory space**, while processes run in **separate memory spaces**.



# Questions?

- What is context switching?
  - Transferring **the control from one process to other process** requires **saving** the state of the old process and **loading** the saved state for new process. This task is known as context switching.



# Questions?

- What are the disadvantages of context switching?
  - **Time taken for switching from one process to other is pure overhead.** The system does no useful work while switching, so one of the solutions is to go for threading when ever possible

# Questions?

- What is race condition?
  - A situation, **where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place**, is called race condition.



# Questions?

- What is process synchronization?
  - To guard against the race condition we need to ensure that only one process at a time can be manipulating the same data. The technique we use for this is called **process synchronization**.





# Questions?

- What is busy waiting?
  - The **repeated** execution of a loop of code while **waiting for an event to occur** is called busy waiting.

# Questions?

- What is mutex?
  - A mutex is a **lock**. It is **a program object** that allows multiple program threads to share the same resource, such as file access, **but not simultaneously**.
  - When a program is started, a mutex is created with a unique name. After this stage, any thread that needs the resource must lock the mutex from other threads while it is using the resource. The mutex is set to unlock when the data is no longer needed or the routine is finished.



# Questions?

- What is a semaphore?
  - It is a **synchronization** tool used to solve **complex critical section problems**. A semaphore is an integer variable that, apart from initialization, is accessed only through two standard **atomic** operations: **down, and up**.
  - **Down checks semaphore**. If not zero, decrements semaphore. If zero, process goes to sleep
  - **Up increments semaphore**. If more than one process asleep, one is chosen randomly and enters critical region (first does a down)



# Questions?

- What is the difference between mutex and semaphore?
  - Mutex is an object owned by thread, so there is a **ownership** in mutex. **Mutex allow only one thread to access resource.**
  - Semaphore is a **signaling mechanism**. It allows **a number of thread to access shared resources.**



# Questions?

- What is throughput, turnaround time, waiting time and Response time?
  - **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)



# Questions?

- What are the scheduling algorithm goals for batch, interactive, real-time operating systems?

## **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

Predictability - avoid quality degradation in multimedia systems

# Questions?

- Name at least four the different job scheduling in operating systems?

Scheduling is the activity of the deciding when process will receive the resources they request.

- **FCFS ---> FCSFS** stands for First Come First Served. In FCFS the job that has been waiting the longest is served next.
- **Round Robin Scheduling--->Round Robin scheduling** is a scheduling method where each process gets a small quantity of time to run and then it is preempted and the next process gets to run. This is called time-sharing and gives the effect of all the processes running at the same time
- **Shortest Job First ---> The Shortest job First scheduling** algorithm is a non-preemptive scheduling algorithm that chooses the job that will execute the shortest amount of time.
- **Priority Scheduling--->Priority scheduling** is a scheduling method where at all times the highest priority process is assigned the resource.



# Ending

- [https://  
www.youtube.com/watch?v=bS3QuOQgU  
u8](https://www.youtube.com/watch?v=bS3QuOQgUu8)
- [https://  
www.youtube.com/watch?v=7FRW4iGjLrc](https://www.youtube.com/watch?v=7FRW4iGjLrc)