Scheduling

- When more than one process is runnable, the OS must decide which one to run first
- The **scheduler** is the part of the OS that is concerned with this decision
- The algorithm it uses is called the **scheduling algorithm**
- Scheduler is concerned with policy, not the mechanism
- What is a good scheduling algorithm?

Scheduling

- Some criteria
  - **Fairness**: make sure each process gets its fair share of the CPU
  - **Efficiency**: keep the CPU busy 100% of the time
  - **Response time**: minimise response time for interactive users
  - **Turnaround**: minimise the time batch users must wait for output
  - **Throughput**: maximise the number of jobs processed per hour
- Some goals are contradictory
- Scheduling performance can be evaluated looking at:
  - utilisation, throughput, service time, queueing time, response time etc.
  - the goal is to optimise both the average and the amount of variation
  - difficult

Scheduling Strategy

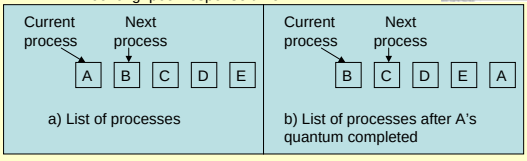
- Most computers have a clock
- At each clock interrupt, the OS decides with process to run (the current one or some other process)
- Involves:
  - Switching context
  - Switching user mode
  - Jumping to the proper location in the user programme to (re)start that programme
- **Dispatch latency**
  - Time it takes for the dispatcher to stop one process and start another running

Scheduling Strategy

- **Nonpreemptive scheduling**
  - also know as “Run to completion” (i.e. allow the current process to complete before running another process
  - Method used in the early days of batch systems
  - Examples:
    - First-In First-Out (FIFO) = First-Come, First-Served (FCFS)
    - Shortest Job First
  - Advantage:
    - Simple and easy to implement
  - Disadvantage:
    - Not suitable for general-purpose systems with multiple competing users

Scheduling Strategy

- **Preemptive scheduling**
  - Strategy of temporarily suspending logically runnable processes
  - Avoids “hogging” of the CPU
- **Round Robin Scheduling**
  - each process is assigned a time interval, called its **quantum**, which it is allowed to run
  - Issue: quantum length
    - Too short: too many process switches
    - Too long: poor response time

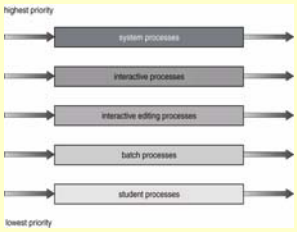


Scheduling

- **Priority Based Scheduling**
  - Each process is assigned a priority
  - Schedule highest priority first
  - All processes within same priority are FCFS (or Round Robin)
  - Priority determined by user or some default mechanism
    - Memory requirements, time limits, other resource usage
  - Starvation: low priority process never runs
    - Solution: build aging into a variable priority
  - Balance required between interactive jobs and batch jobs

Scheduling

- **Multi-level Queues**
  - Each queue has its scheduling algorithm
  - Some other algorithm (e.g. priority based) arbitrates between queues
  - Processes can move between queues
  - Complex, but flexible



Scheduling

- Guaranteed scheduling
  - Make real promises to the user about performance and live up to them
  - Real-time systems
- Policy vs mechanism
  - Scheduling algorithm parameterised, parameters supplied by user
- 2-level scheduling
  - Subset of runnable processes loaded into main memory
  - Scheduler only chooses from this subset
  - Higher-level scheduler manages swapping processes between main memory and secondary memory