1878

Western

UNIVERSITY · CANADA

Chapter 5a – Basic Concepts and Scheduling Criteria

Spring 2023

Western

Basic Concepts and Scheduling Criteria

- Basic Concepts
  - CPU-I/O Burst Cycle
  - CPU Scheduler
  - Preemptive and Nonpreemptive Scheduling
  - Dispatcher
- Scheduling Criteria

Basic Concepts

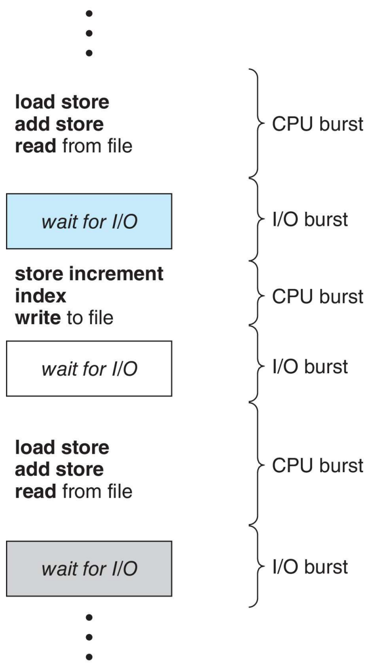
- (Since modern operating systems are multithreaded, we may still use the term "processes", but we actually mean "threads". Treat them interchangeably.)
- (Since modern computers are usually multi-core machines, we may still use the term CPU or processor, but we actually mean "cores")
- On a single-core machine, only one process can run at a time. Multi-core machines can run multiple processes at once
- Regardless of the number of cores, multiprogramming tries to keep all cores working if there are processes to be run
- The CPU is a primary resource so proper scheduling is of high importance

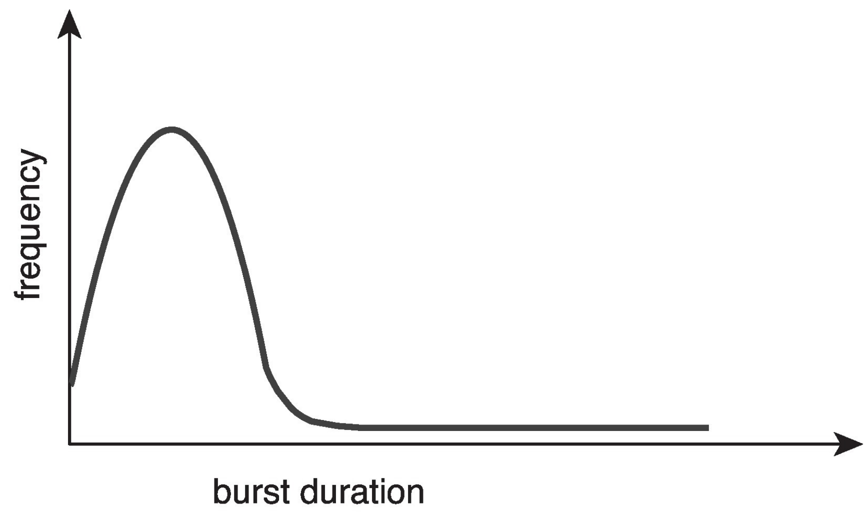
CPU-I/O Burst Cycle

- **CPU–I/O Burst Cycle** – Process execution consists of a cycle of CPU execution and I/O wait
- CPU burst followed by I/O burst
- CPU burst distribution is of main concern

CPU-I/O Burst Cycle

- CPU–I/O Burst Cycle and CPU Burst Frequency





CPU Scheduler

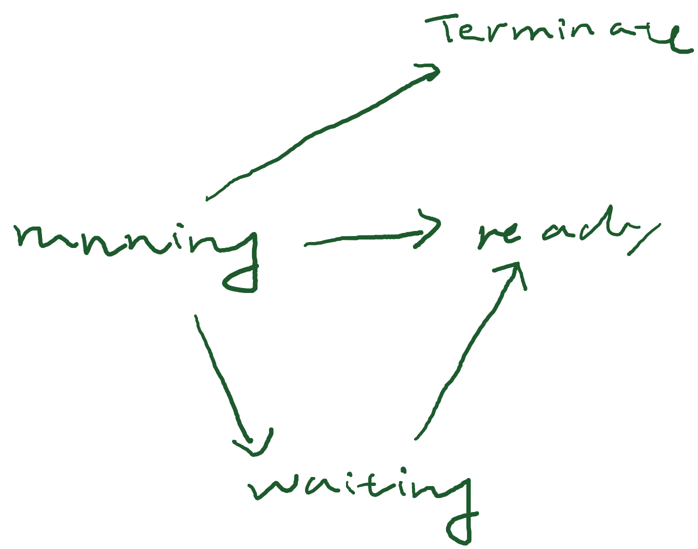
Process control box.

- The CPU Scheduler is tasked with selecting a process (actually, a PCB) from the ready queue and allocating the CPU to that process
    - The ready queue *could* be a simple first-in, first-out (FIFO) queue. It could be a tree, priority queue, or completely unordered

Preemptive and Nonpreemptive Scheduling

- CPU scheduling decisions may take place when a process:

1. Switches from running to waiting state
2. Switches from running to ready state
3. Switches from waiting to ready state
4. Terminates



Preemptive and Nonpreemptive Scheduling

- For situations 1 and 4, there is no choice in terms of scheduling. A **new** process (if one exists in the ready queue) must be selected for execution.
  - This is a nonpreememptive scheduling scheme
- For situations 2 and 3, however, there is a choice about which process goes next
  - This is a preemptive scheduling scheme

Preemptive and Nonpreemptive Scheduling

- Under a nonpreemptive scheduling scheme a process keeps the CPU until it releases it (by terminating or by "voluntarily" switching to a waiting state)

- Under a preemptive scheduling scheme, another process could forcibly interrupt a running process depending on what scheduling algorithm is used

- Virtually all modern operating systems including Windows, MacOS, Linux, and UNIX use preemptive scheduling algorithms

  - User interaction and real-time computing would essentially require preemptive scheduling to work

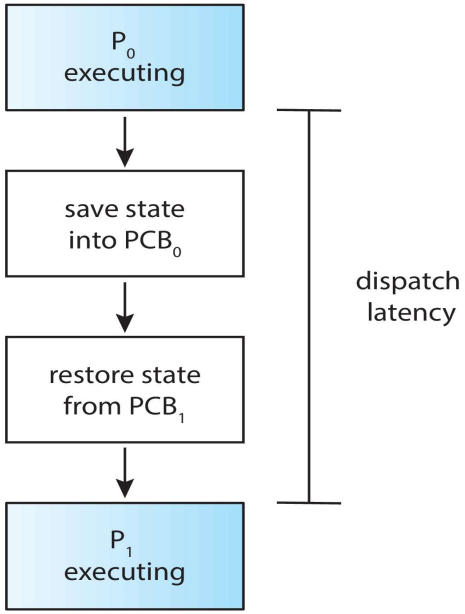
Preemptive and Nonpreemptive Scheduling

- Preemptive scheduling can result in race conditions when data are shared among several processes.
- Consider the case of two processes that share data. While one process is updating the data, it is preempted so that the second process can run. The second process then tries to read the data, which are in an inconsistent state.
- This issue will be explored in detail in Chapter 6.

Dispatcher

- Dispatcher module gives control of the CPU to the process selected by the CPU scheduler; this involves:
  - Switching context
  - Switching to user mode
  - Jumping to the proper location in the user program to restart that program
- **Dispatch latency** – time it takes for the dispatcher to stop one process and start another running
  - This should be as fast as possible!
  - Use `vmstat` or `/proc/<pid>/status` to see context switch performance

Dispatcher



P_0 executing
save state into PCB_0
restore state from PCB_1
P_1 executing
dispatch latency

Western

UNIVERSITY · CANADA