

CST 347 ?? Real Time OS

CST 347 – Real Time OS

Oregon TECH

Lecture 07 – CPU Scheduling I Troy Scevers



CST 240 – Real Time OS



Topics

Topics

- Introduction
- Basic Concepts
- Scheduling Criteria ←
- Scheduling Algorithms
- Thread Scheduling
- Multiple-Processor Scheduling
- Real-Time CPU Scheduling ↗
- Operating Systems Examples
- Algorithm Evaluation

Introduction

Introduction

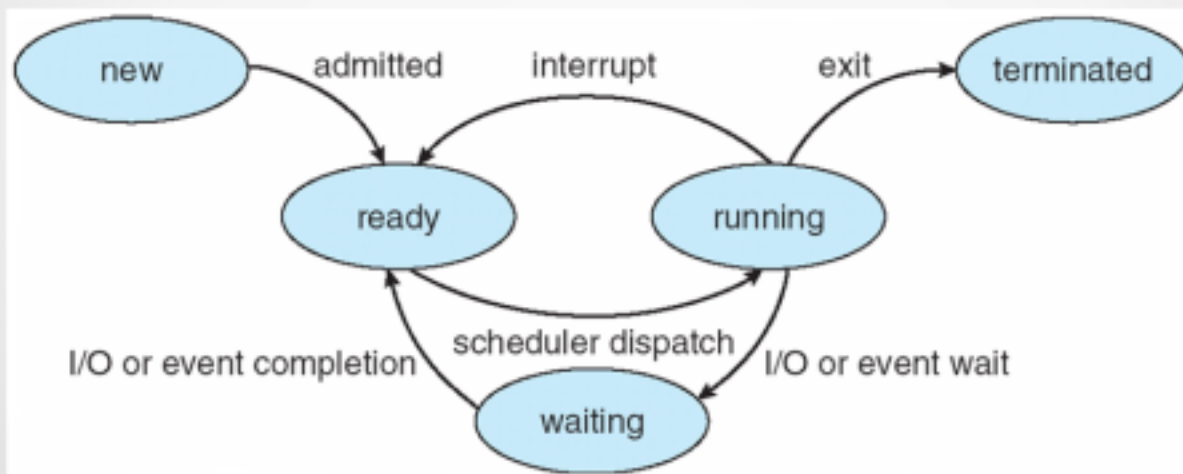
- Objectives

- To introduce CPU scheduling, which is the basis for multiprogrammed operating systems
- To describe various CPU-scheduling algorithms
- To discuss evaluation criteria for selecting a CPU-scheduling algorithm for a particular system
- To examine the scheduling algorithms of several operating systems

Introduction

Introduction

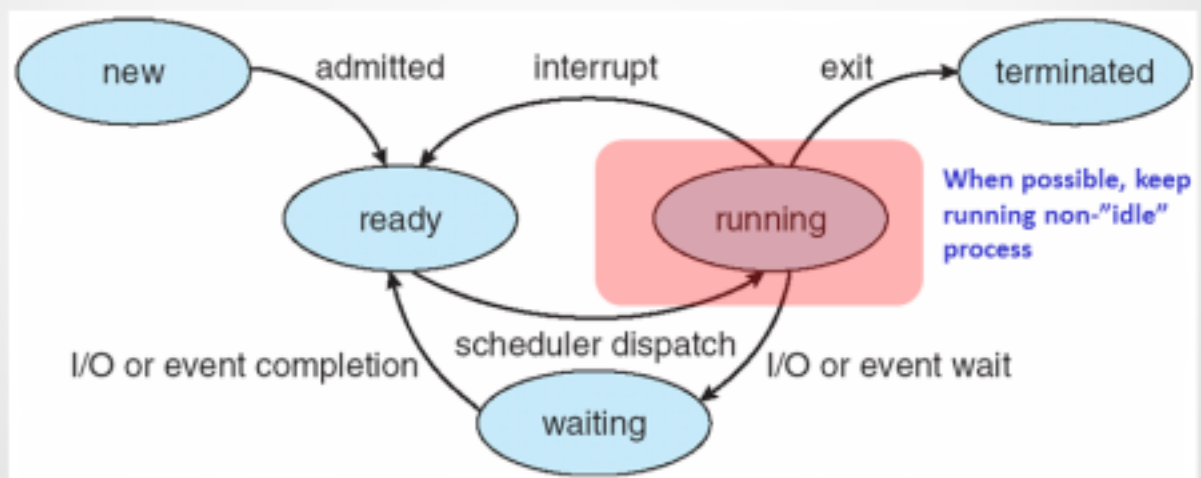
Recall the state of a Process *A SINGLE*



Introduction

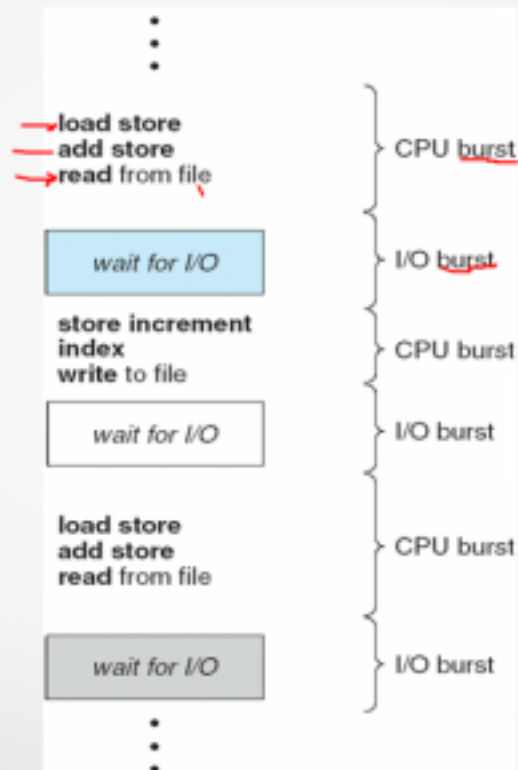
Introduction

Recall the state of a Process

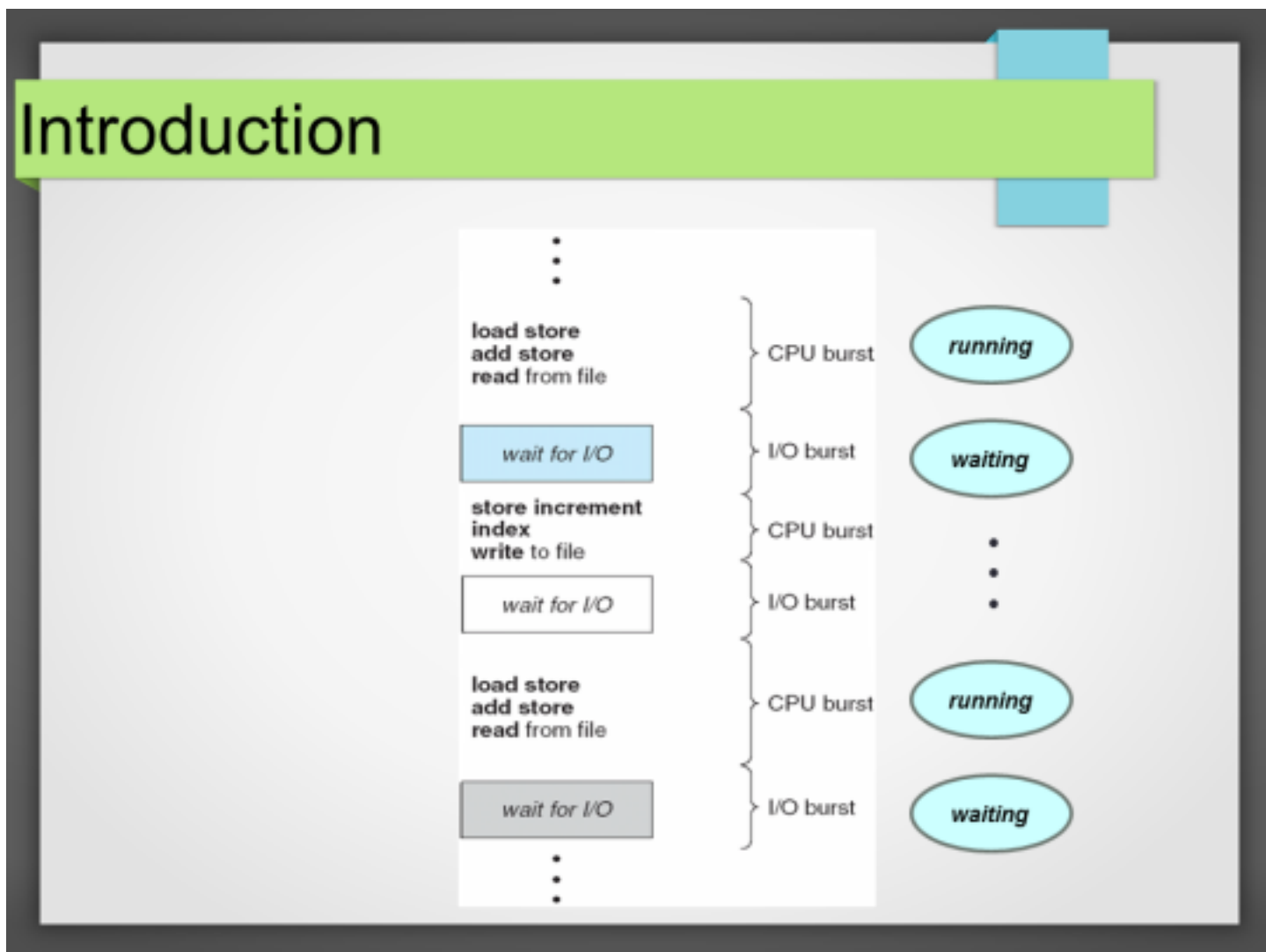


Introduction

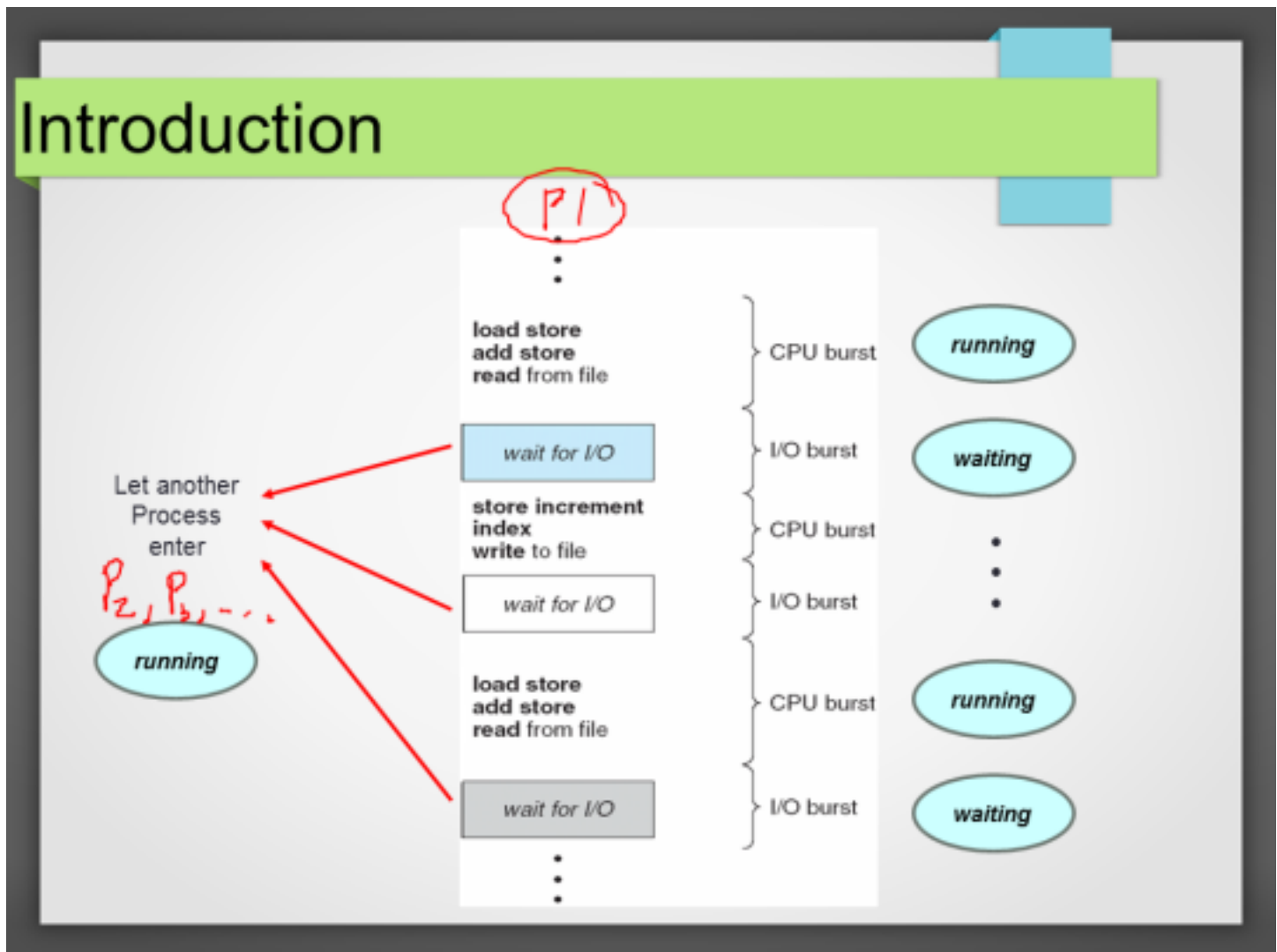
Introduction



Introduction

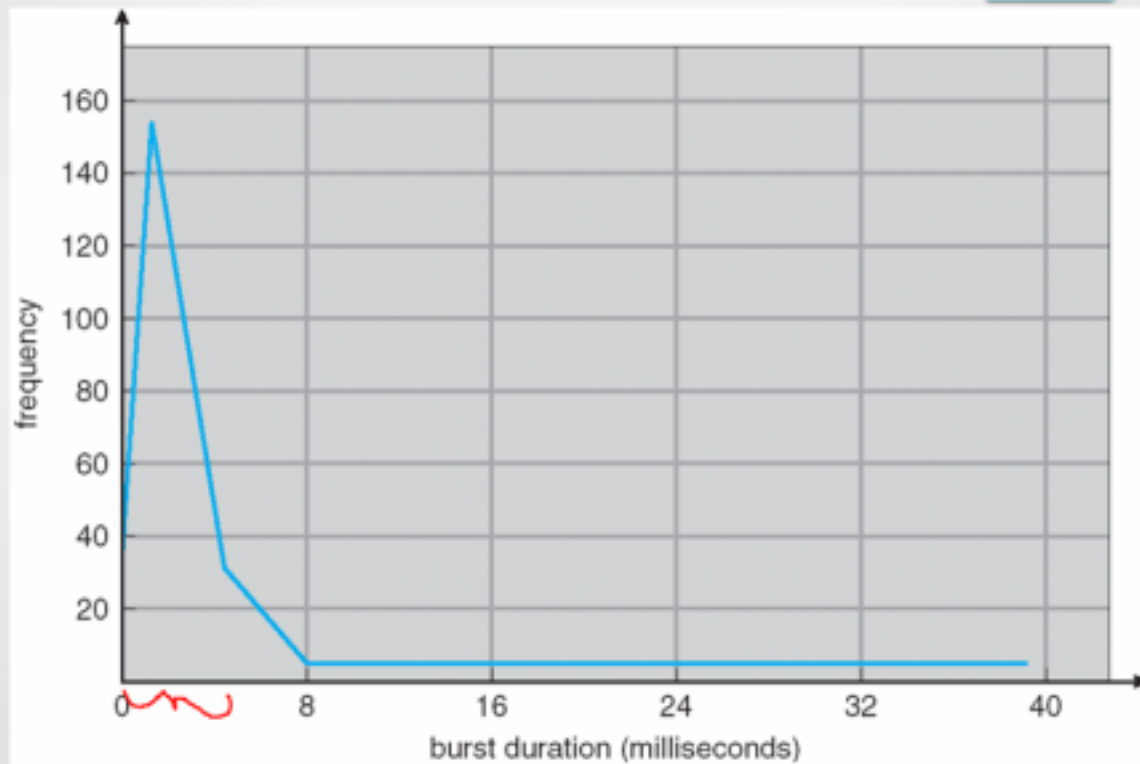


Introduction



Histogram of CPU-burst Times

Histogram of CPU-burst Times



CPU Scheduler

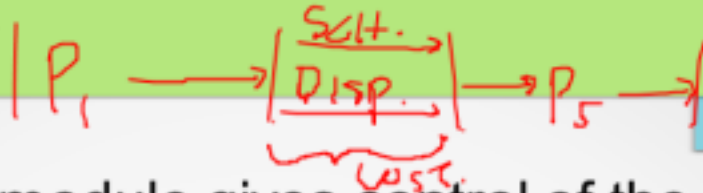
CPU Scheduler

- Short-term scheduler selects from among the processes in ready queue, and allocates the CPU to one of them
 - Queue may be ordered in various ways
- CPU scheduling decisions may take place when a process:
 - 1. Switches from running to waiting state (BLOCK)
 - 2. Switches from running to ready state
 - 3. Switches from waiting to ready
 - 4. Terminates
- Scheduling under 1 and 4 is non-preemptive ONLY
- All other scheduling is preemptive {INCL. 2/3}
- Consider access to shared data
- Consider preemption while in kernel mode
- Consider interrupts occurring during crucial OS activities



Dispatcher

Dispatcher



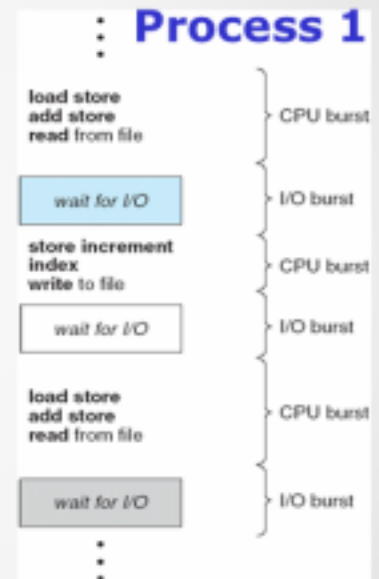
- Dispatcher module gives control of the CPU to the process selected by the short-term scheduler; this involves:

- switching context ← { RESTORING ALL REGs. }
- switching to user mode ←
- jumping to the proper location in the user program to restart that program ← PC ← from (PCB)

- Dispatch latency – time it takes for the dispatcher to stop one process and start another running

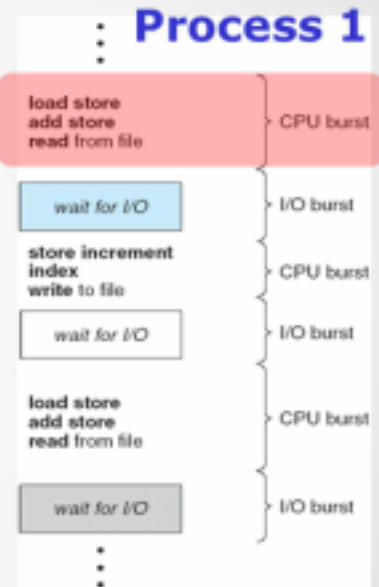
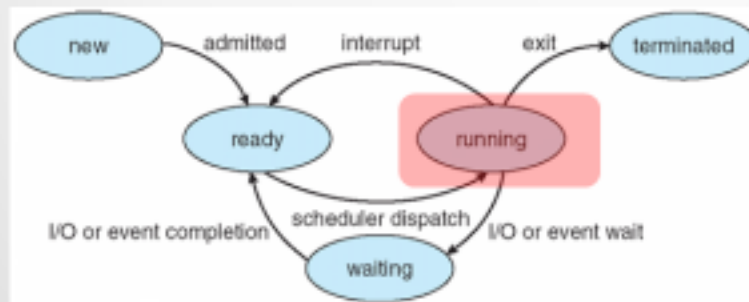
Introduction

Introduction



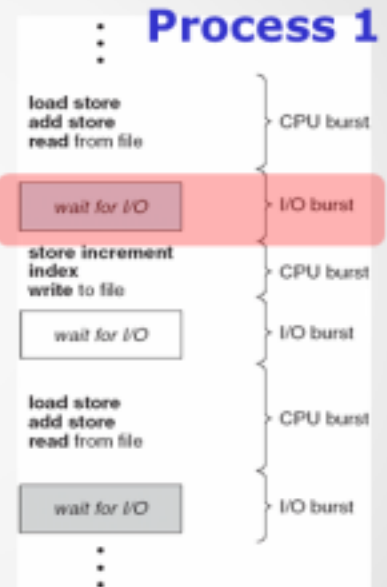
Introduction

Introduction



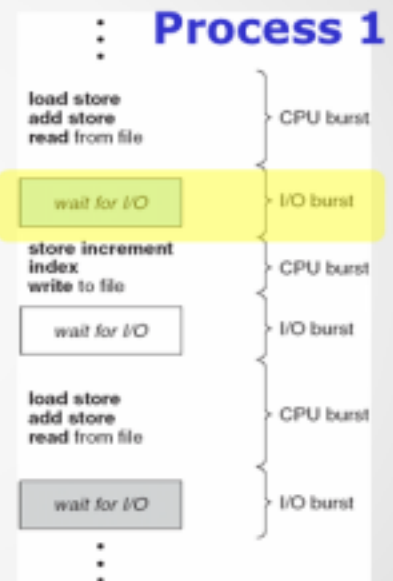
Introduction

Introduction



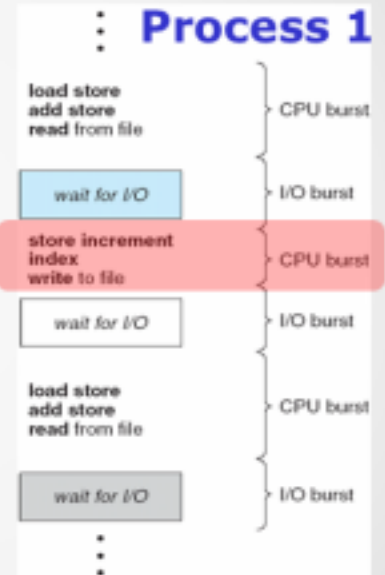
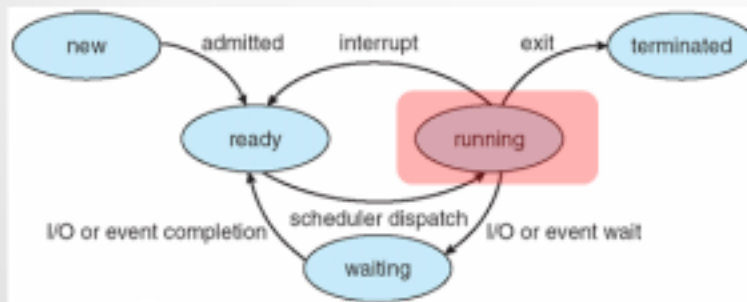
Introduction

Introduction



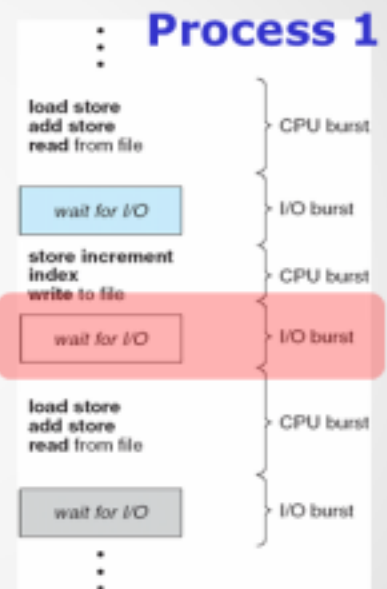
Introduction

Introduction



Introduction

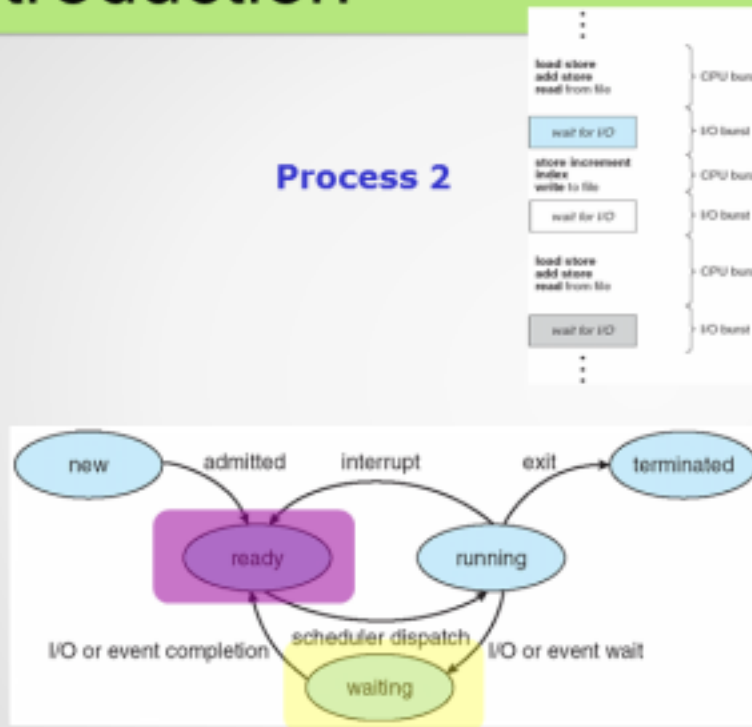
Introduction



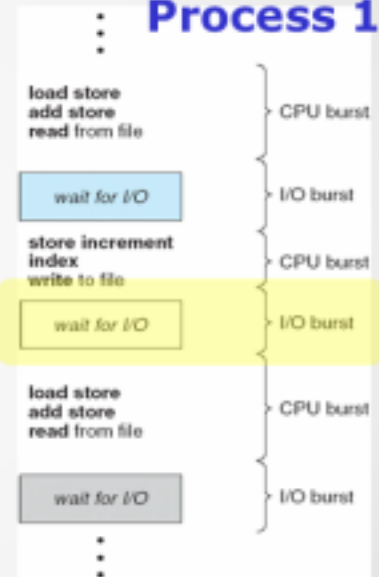
Introduction

Introduction

Process 2



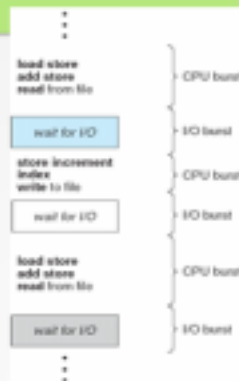
Process 1



Introduction

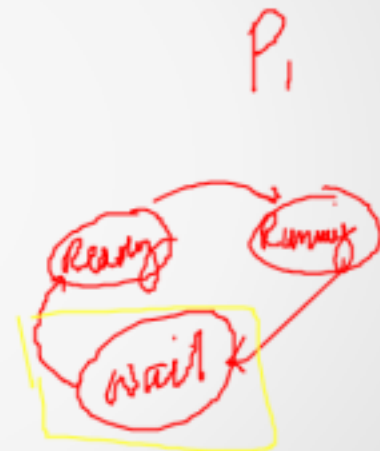
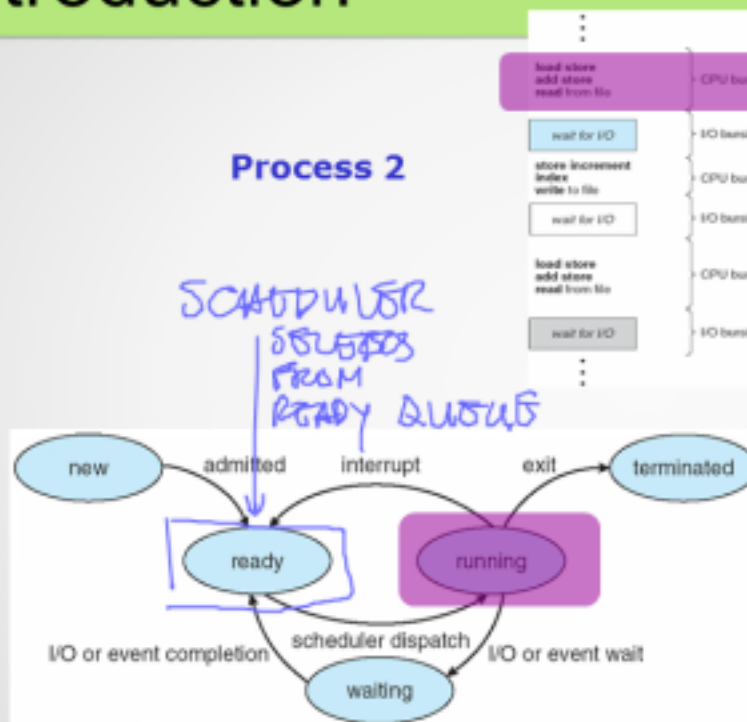
Introduction

Process 2



Introduction

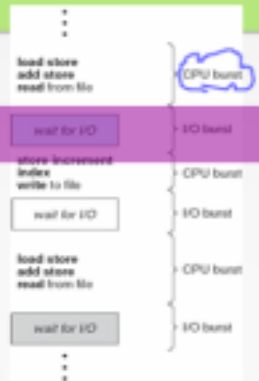
Introduction



Introduction

Introduction

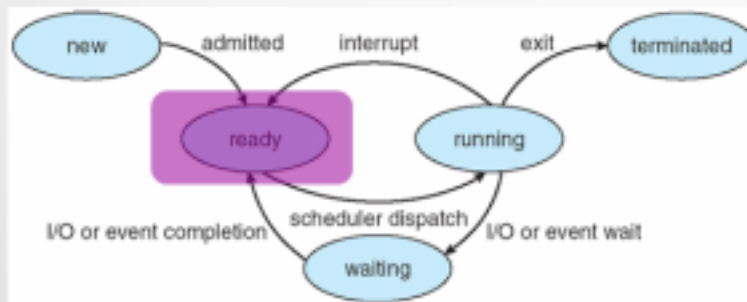
Process 2



Introduction

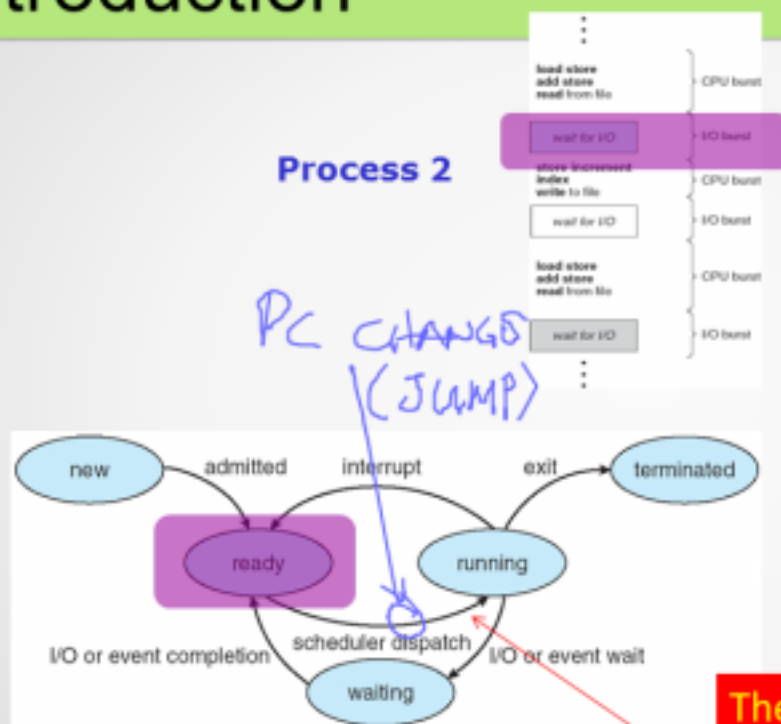
Introduction

Process 2



Introduction

Introduction



The dispatcher gives control of the CPU to the next Process to run

Scheduler Dispatcher

Scheduler Dispatcher

Switching context
Switching to user mode
Reentering the user program at proper location (PC
modification - jump)



Scheduling Criteria

Scheduling Criteria

- **CPU utilization** – keep the CPU as busy as possible
- **Throughput** – # 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
 - CREATION
 - RUNNING
 - WAITING
- **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 Algorithm Optimization Criteria

Scheduling Algorithm Optimization Criteria

- Max CPU utilization
- Max throughput
- Min turnaround time
- Min waiting time
- Min response time

Scheduling Algorithms

Scheduling Algorithms

- First-Come, First-Served (FCFS)
- Shortest-Job-First (SJF) Scheduling
- Priority Scheduling ← IMPORTANT FOR REAL TIME
- Round-Robin Scheduling
- Multilevel Queue Scheduling