6 Operating System

6.1 Introduction

Definition	A level of abstraction between hardware and software
OS's tasks	• Process management • Memory management • File system management (and I/O)

Goals of an OS

End users	• Provide consistent user interface • Manage few applications • Protect users from malicious code
Programmers	• Provide programming interface • Enable access to hardware and I/O • Manage system resources

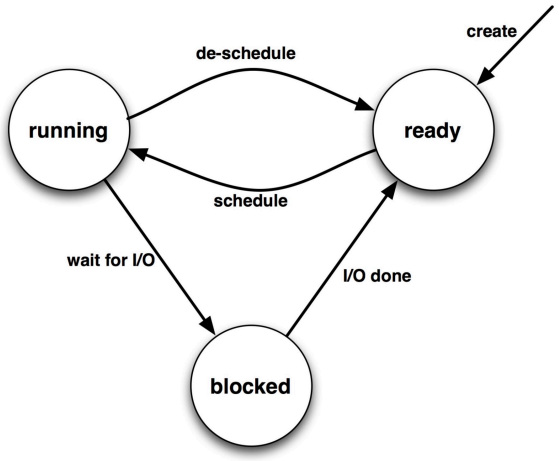
Abstraction in OS
Through virtualization to provide fake resources:
- Entire CPU
- Large, contiguous memory
- Other system resources (graphics, disk, keyboard)

6.2 Virtualizing the CPU

Goals of Virtualization
1. Prevent process from running continuously
2. Provide a fair share of CPU time to each process

Virtualization Mechanisms

Ready State	• Process is ready for execution but not executed • OS schedules all process in order
Running State	• Process is executed Two things can happen: • Process is de-schedulesd and returns to "Ready" • Process requests some I/O to happen and transfer to "Blocked"
Blocked State	• I/O is being executed • Returns back to "Ready"



Limited Direct Execution (LDE)
Two challenges during virtualization

Performance	CPU time should be spent on running processes
Control	• Enable fair scheduling • Protect users against malicious code

LDE Modes

User Mode	• A subset of the instructions • No I/O execution • Only normal applications
Kernel Mode	• Codes have no restrictions • OS runs in kernel mode • Interrupts trigger CPU switches into kernel mode

System Calls

Function	Switches user mode to kernel mode
Table of System Calls Handlers	1. Save process context (registers) into memory 2. Switch CPU to kernel mode 3. Jump to handler n and execute it 4. Restores process context 5. Switch CPU back to user mode 6. Return to calling process

Process Switching

Cooperative	Switch to kernel mode when: • User makes system calls • Hardware interrupt happens
Preemptive	Wait for timer interrupt

Cooperative and Preemptive Timesharing

	Function	Disadvantage
Cooperative	• All processes cooperate with OS • Make system calls in regular intervals	• Process might not cooperate • Malicious process might cause bug
Preemptive	• Hardware generates an interrupt in regular intervals • Interrupt handler switch to different process	-

Process Scheduling
Scheduling Goals
1. Turnaround time
- How long it takes to complete a process

2. Fairness
- Each process will be allocated a fair share of processing time

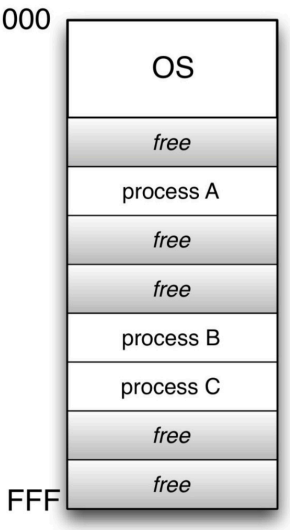
Scheduling Policies

Policy	Method	Conclusion
First-come First-served	Order according to sequence	Poor turnaround
Shortest Job First	Order from the shortest time first	Good turnaround
Robin-robin Scheduling (modern PCs)	Split processes into time slices	• Smaller time slices create better fairness • Context switching takes time • OS needs to compromise

6.3 Virtualizing the Memory

Goal of Virtualization
1. Protect memory from malicious process
2. Ease programming
3. Use more memory than physical available RAM

Multiprogramming
- Each process has its own address space
- Every process gets a fixed region of the memory
- Let each process "thinks" that address start at 0
- OS converts the virtual address to physical address



Virtual Memory and Protection

Virtual Memory	Memory Protection
Uses base register to prevent access lower than it	Uses bound register to prevent access more than it
Instruction will add base register	Instruction will check between base and bounds

Realistic Virtual Memory Systems
- OS allocates smaller memory chunks called "pages"
- Each process has a set of pages to access
- Pages are modifiable

Virtual Memory Usage
- Unused pages will be stored in hard disk or reuse it for other process
- If a process tries to access that page, hardware creates interrupt "page fault"
- OS loads back the page from hard disk