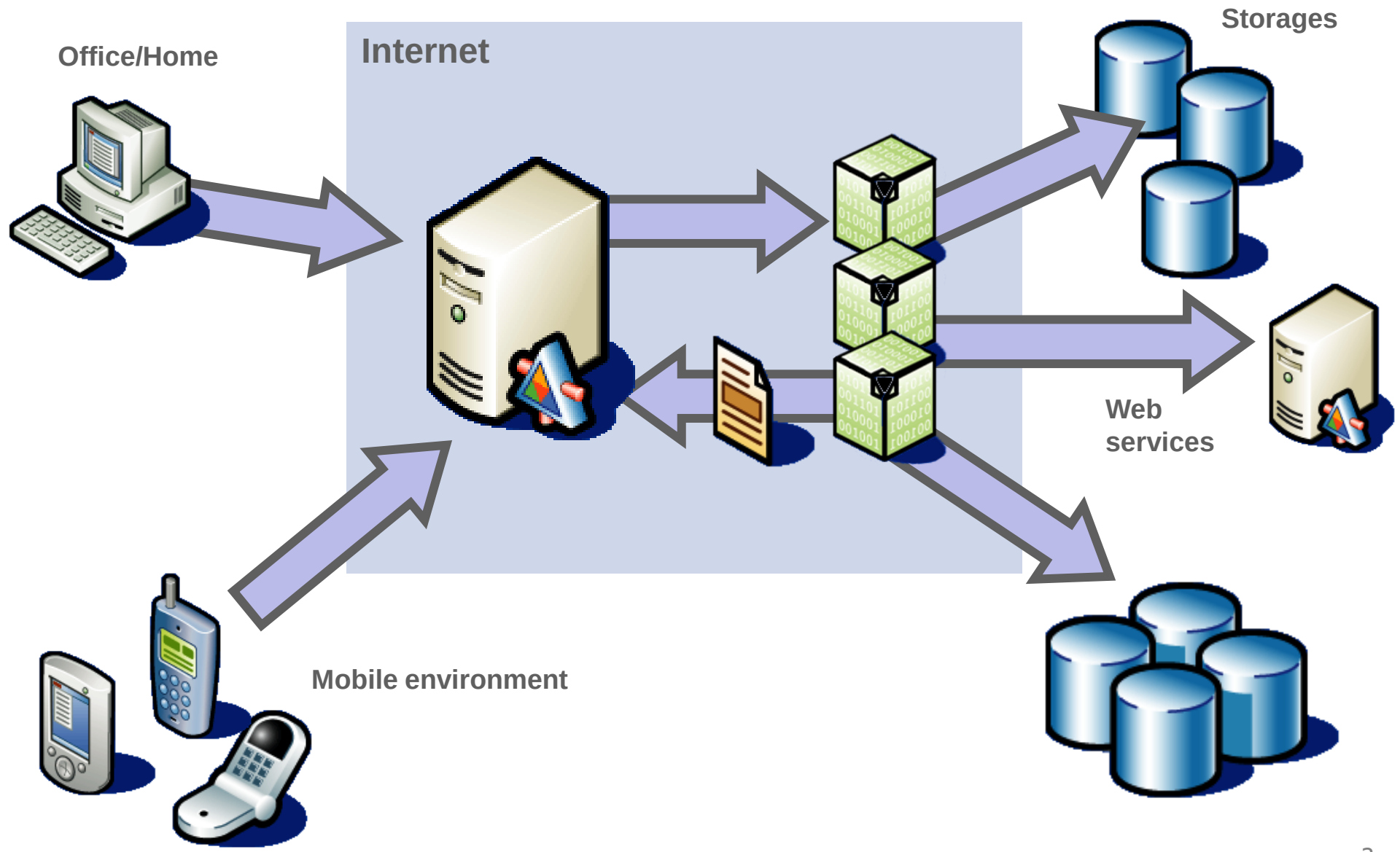
Introduction (2022-Spring)

Text Books and Grading

- Major text: A. Silberschatz, P.B. Galvin, and G. Gagne, "Operating System Concepts"
- Grading
  - Mid-term: 30%
  - Final term: 30%
  - 4 Projects: 30%
    - Your grade will be limited below C+ if you do not submit any projects.
    - Source code cheating: grades will be deducted.
    - No co-work!!!!!
  - Attendance: 10%

The Internet of Things(IoT)



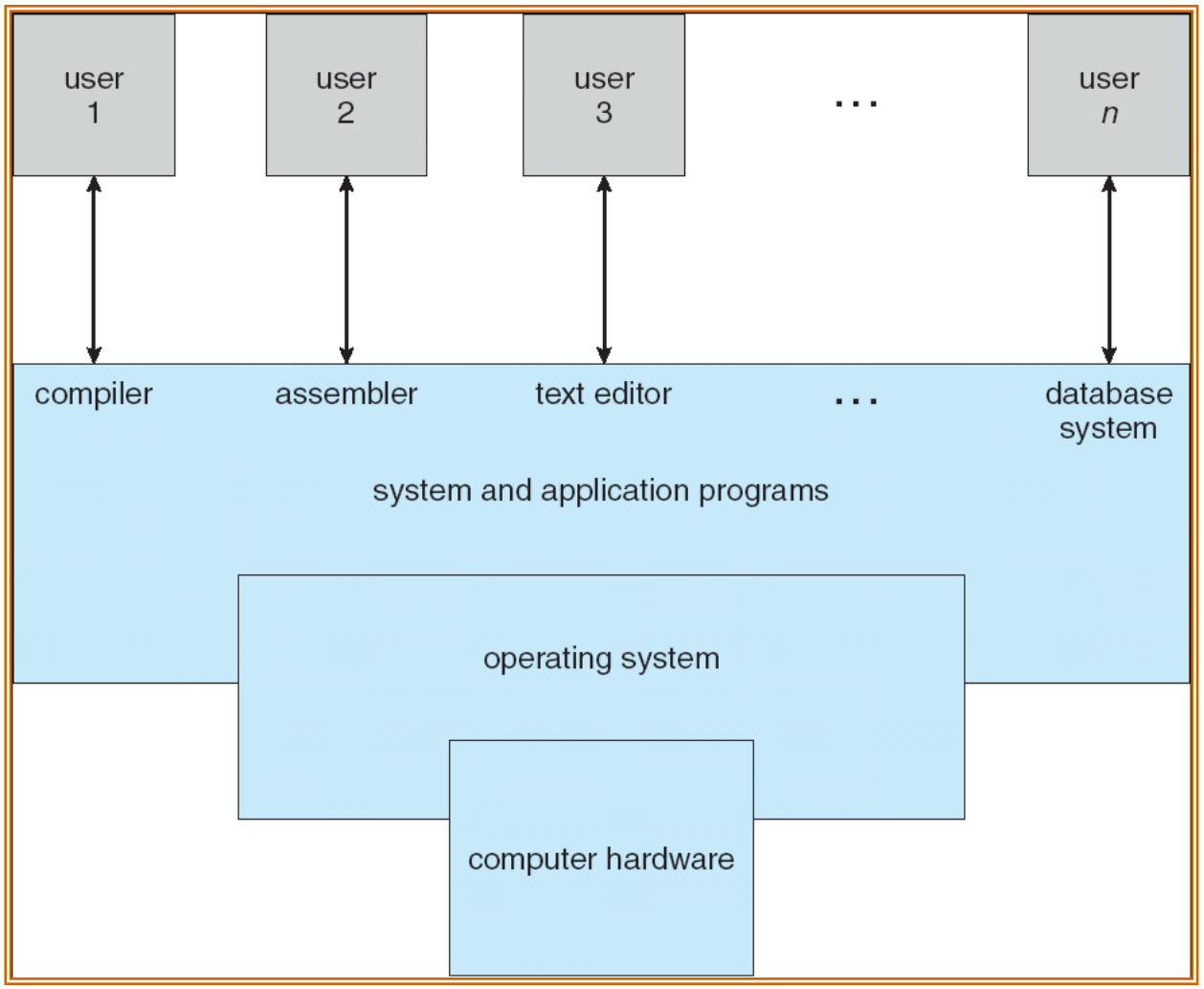
Modern Operating Systems

- Operating systems for servers and desktops

- Mobile operating systems
- Embedded systems

- Operating systems for web servers and multimedia systems

Four Components of a Computer System



Computer System Structure

- Computer system can be divided into four components
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - System and application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers

What if we didn't have an Operating System?

- Source Code ⇒ Compiler ⇒ Object Code ⇒Hardware
- How do you get object code onto the hardware?
- How do you print out the answer?

What is Operating System?

- Operating system goals
  - Execute user programs
  - Provide libraries of standard services and make solving user problems easier.
  - Make the computer system convenient to use.
  - Use the computer hardware in an efficient manner.

- **Definition**
  - A program that acts as an intermediary between a user of a computer and the computer hardware.
  - Common functions of controlling and allocating resources are brought together into one piece of software
  - The one program running at all times on the computer, usually called *kernel*

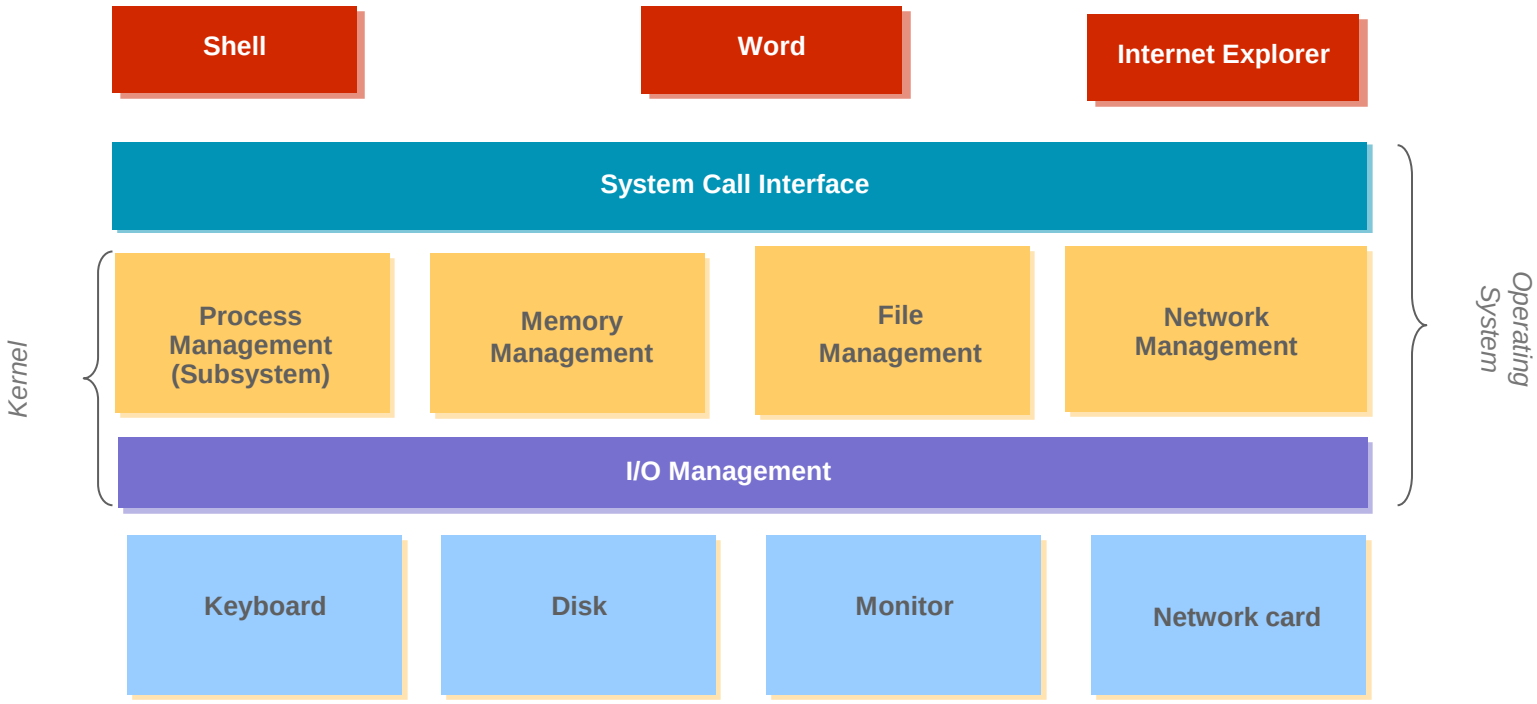
- No universally accepted definition
  - “Everything a vendor ships when you order an operating system” is good approximation, but varies wildly

Operating System Roles

- User view
  - To maximize works (play) that users are performing. In this case, the operating system is designed mostly for ease of use.

- System view
  - OS is a **resource allocator**
    - Manages all resources
    - Decides between conflicting requests for efficient and fair resource use
    - Process management, memory management, etc.
  - OS is a **control program**
    - Controls execution of programs to prevent errors and improper use of the computer
    - Process management, memory management, etc

Operating System Structure

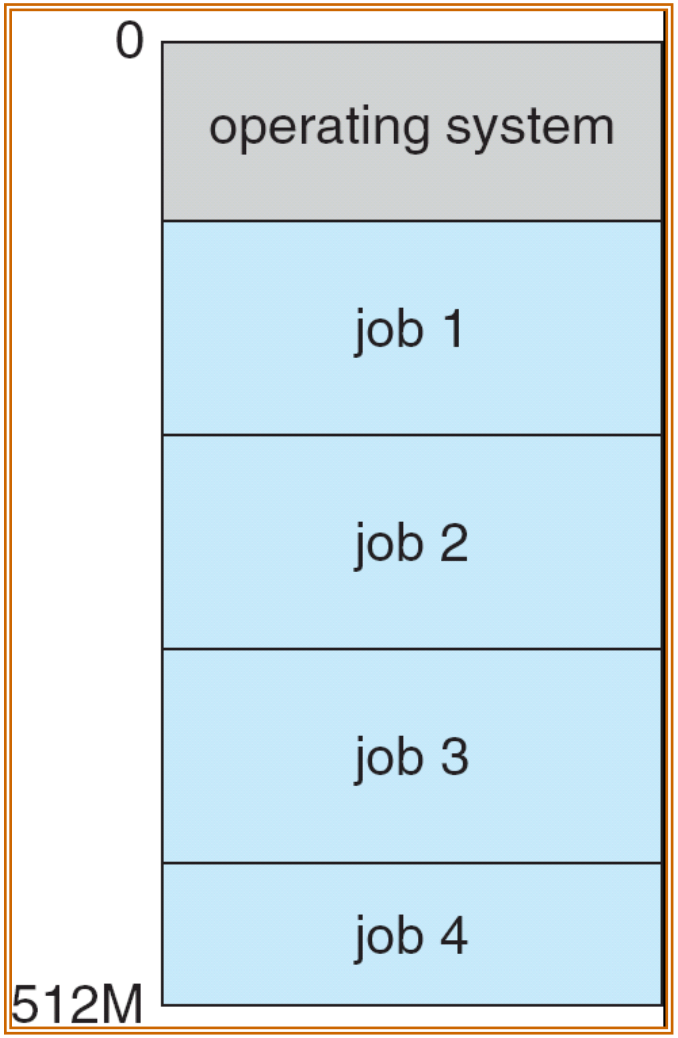


Operating System Structure (Cont'd)

- **Multiprogramming**/Multitasking
  - Single user or job cannot keep CPU and I/O devices busy at all times
  - A subset of total jobs in system is kept in memory
  - When one job has to wait (for I/O for example), OS switches to another job

- **Subsystems required to support multiprogramming**
  - program execution in memory ⇨ **process**
    - **Process management**
  - If several jobs ready to run at the same time ⇨ **CPU scheduling**
    - **Process management**
  - If processes don't fit in memory, **swapping** moves them in and out to run
    - Memory management
  - **Virtual memory** allows execution of processes not completely in memory
    - Memory management

Memory Layout for Multiprogrammed System



Process Management

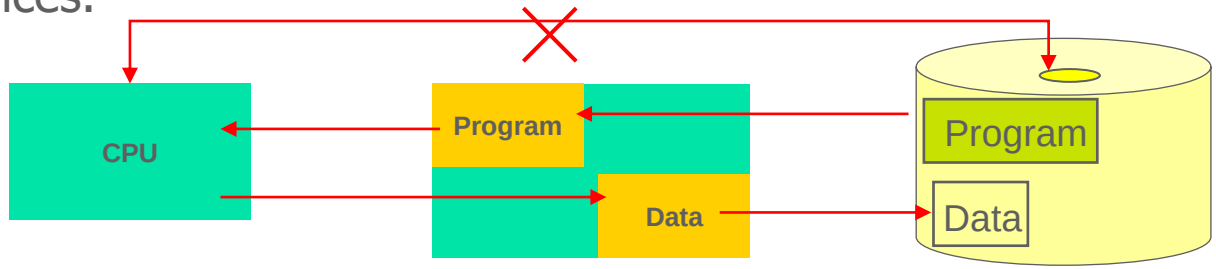
- What is a process ?
  - a program in execution.
  - Process has several resources: CPU, memory, I/O, files, data
- Process termination requires reclaim of any reusable resources
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads

Process management is responsible for the following activities

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling

Memory Management

- The usage of main memory
  - A repository of quickly accessible data that is shared among the CPU and I/O devices.



- Memory management has an important effect on
  - CPU utilization and computer response to users

- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Automatically freeing memory space occupied by a process when it is terminated (Garbage collection)
  - Deciding which processes and data to move into and out of memory
  - Allocating and deallocating memory space as needed

File and Storage Management

- File-System management
  - Abstracts physical properties to logical storage unit - **file**
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
- Activities in file management
  - Creating and deleting files and directories
  - Primitives to manipulate files and directories
  - Mapping files onto secondary storage

- Storage Management
  - Disks and SSDs are used to store data that does not fit in main memory or data that must be kept for a “long” period of time.
  - Entire speed of computer operation depends on disk subsystem and its algorithms
- Activities in storage management
  - Free-space management
  - Storage allocation
  - Disk scheduling

I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Caching (storing parts of data in faster storage for performance)
  - General device-driver interface
  - Drivers for specific hardware devices

Protection and Security

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service

Dual-mode Operation

- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**

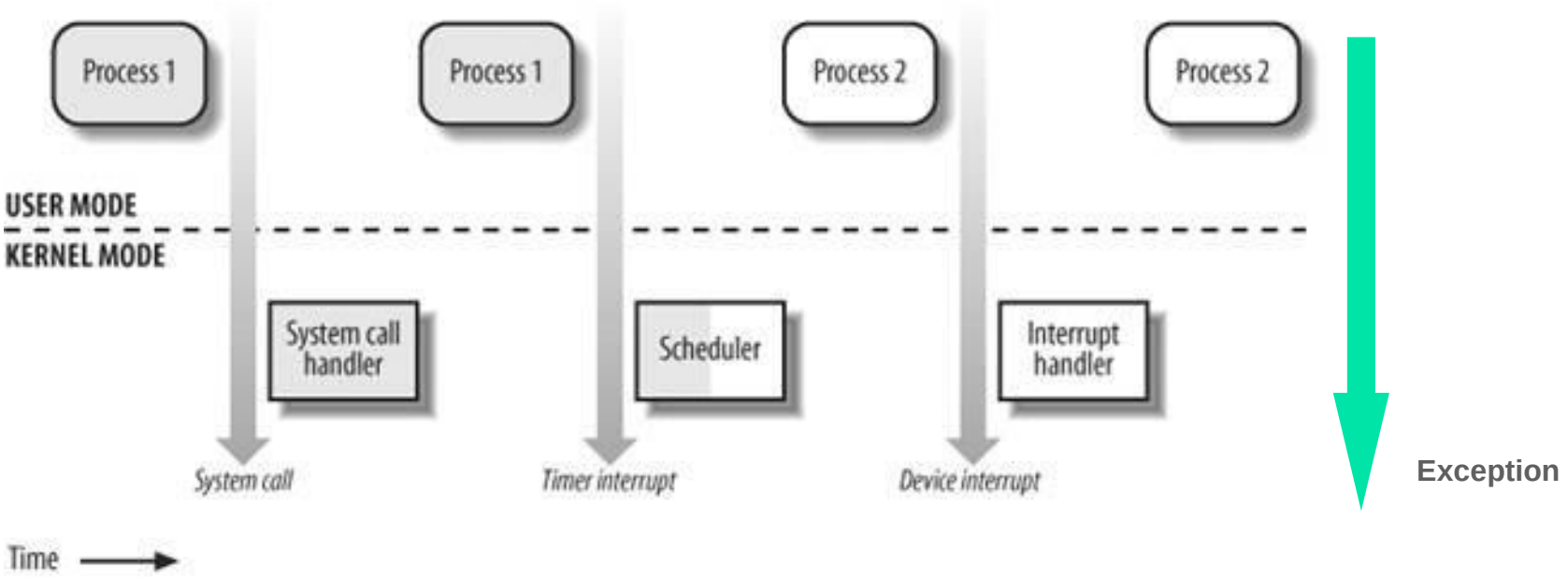
- Hardware provides at least two modes:
  - "Kernel" mode (or "supervisor" or "protected")
  - "User" mode: Normal programs executed

- Some instructions/ops prohibited in user mode:
  - Example: cannot modify page tables in user mode
    - Attempt to modify $\Rightarrow$ Exception generated

Dual-mode Operation (Cont'd)

- Exceptions/Interrupts
  - Any event that disturbs the normal execution of the processor and focuses the processor into execution of special instructions.
- Exceptions
  - internal events such as events generated by the execution of processor instructions.
- Examples of exceptions
  - Alignment exceptions: processors read/writes at the beginning of an odd memory.
  - Arithmetic operations: division by zero.
  - TRAP
- Interrupts
  - Electric signal raised by external events, which do not relate to the execution of processor instructions.
  - The source of external events are typically external hardware devices.
- Examples of interrupts
  - The reset button on embedded boards.
  - Electrical signals generated on devices.

Transition from User to Kernel Mode



Transition from User to Kernel Mode (Cont'd)

- **Implementation**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
    - System call changes mode to kernel, return from call resets it to user