

Introduction to Operating Systems (Part II)

Amir H. Payberah
amir@sics.se

Amirkabir University of Technology
(Tehran Polytechnic)

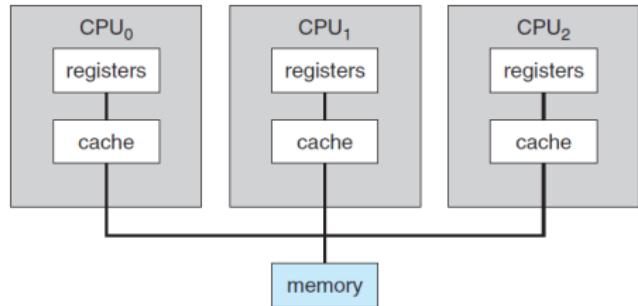


Computer System Architecture

- ▶ A computer system can be organized in a number of different ways, e.g., **number of general-purpose processors**:
 - **Single processor** systems
 - **Multiprocessor** systems
 - **Clustered** systems

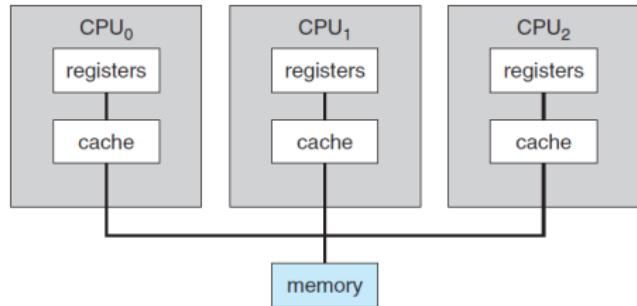
Multiprocessor Systems

- Also known as parallel systems or tightly-coupled systems.



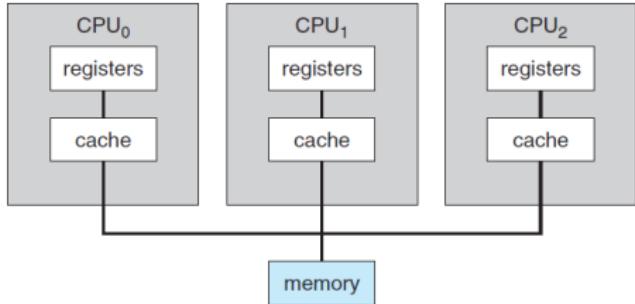
Multiprocessor Systems

- ▶ Also known as **parallel systems** or **tightly-coupled systems**.
- ▶ **Advantages** include:
 - Increased throughput
 - Economy of scale
 - Increased reliability



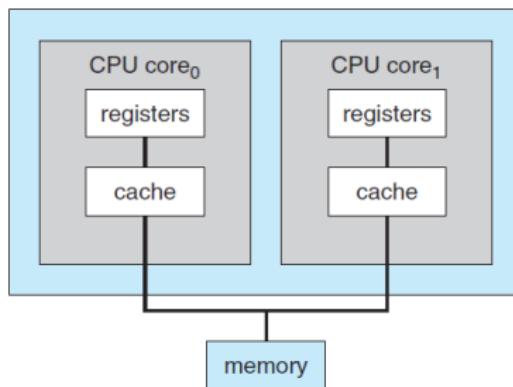
Multiprocessor Systems

- ▶ Also known as **parallel systems** or **tightly-coupled systems**.
- ▶ **Advantages** include:
 - Increased throughput
 - Economy of scale
 - Increased reliability
- ▶ **New trends** in multiprocessor systems:
 - **Multicore** systems
 - **Blade servers**



Multicore Systems

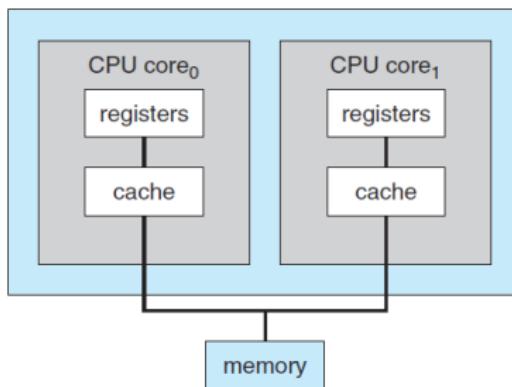
- ▶ Including **multiple** computing cores on a **single chip**.



[A dual-core design with two cores placed on the same chip]

Multicore Systems

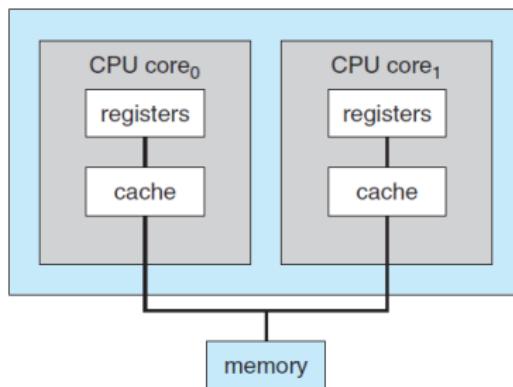
- ▶ Including **multiple** computing cores on a **single chip**.
- ▶ They can be **more efficient** than **multiple chips with single cores**:



[A dual-core design with two cores placed on the same chip]

Multicore Systems

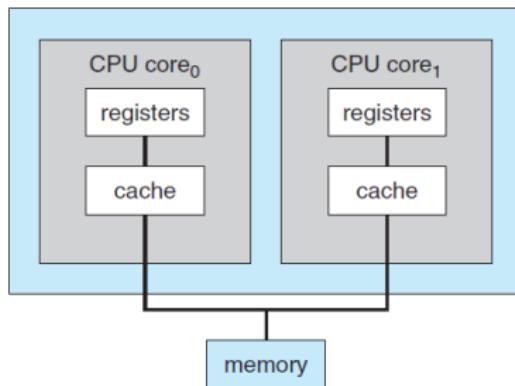
- ▶ Including **multiple** computing cores on a **single chip**.
- ▶ They can be **more efficient** than **multiple chips with single cores**:
 - Faster communication



[A dual-core design with two cores placed on the same chip]

Multicore Systems

- ▶ Including **multiple** computing cores on a **single chip**.
- ▶ They can be **more efficient** than **multiple chips with single cores**:
 - Faster communication
 - Less power consumption



[A dual-core design with two cores placed on the same chip]

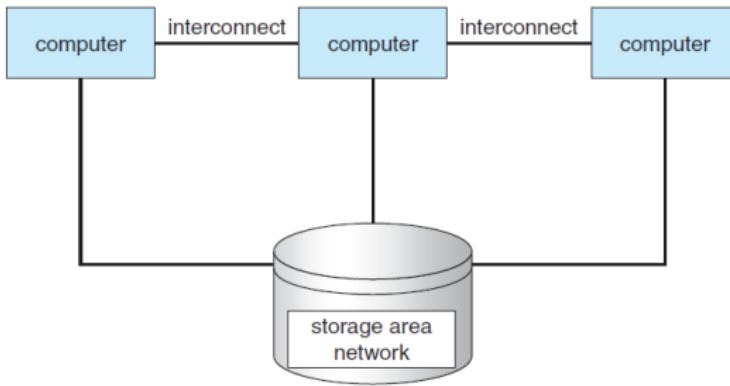
Blade Servers

- ▶ **Multiple** processor boards, I/O boards, and networking boards are placed in the **same chassis**.
- ▶ Each blade-processor board **boots independently** and runs **its own operating system**.



Clustered Servers

- ▶ Like multiprocessor systems, but **multiple systems** working together.
- ▶ Usually **sharing storage** via a storage-area network (SAN).

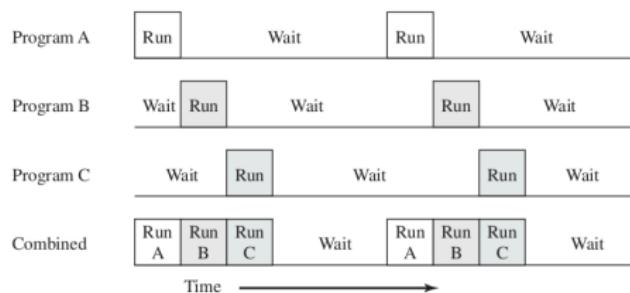


[General structure of a clustered system]

Operating System Structure

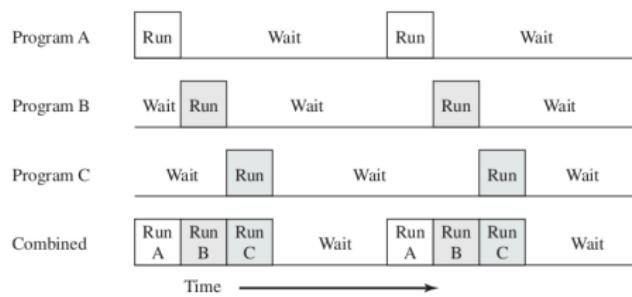
Multiprogramming (1/2)

- Multiprogramming (batch system).



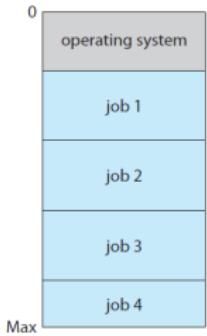
Multiprogramming (1/2)

- ▶ Multiprogramming (batch system).
 - ▶ CPU always has one to execute: efficiency.



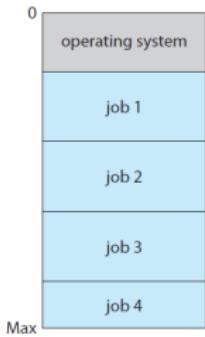
Multiprogramming (2/2)

- ▶ A subset of total jobs in system is kept in memory.



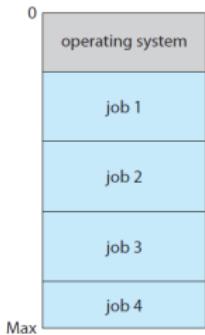
Multiprogramming (2/2)

- ▶ A subset of total jobs in system is kept in memory.
- ▶ One job selected and run via job scheduling.



Multiprogramming (2/2)

- ▶ A subset of total jobs in system is kept in memory.
- ▶ One job selected and run via job scheduling.
- ▶ When it has to wait (for I/O for example), OS switches to another job.



Timesharing

- ▶ Timesharing (multitasking)

► Timesharing (multitasking)

- Providing each user with a **small** portion of a **time-shared** computer.

► Timesharing (multitasking)

- Providing each user with a **small** portion of a **time-shared** computer.
- Each user has at least one separate **program in memory**, called a **process**.

► Timesharing (multitasking)

- Providing each user with a **small** portion of a **time-shared** computer.
- Each user has at least one separate **program in memory**, called a **process**.
- Each **process** typically executes for only a **short time**.

► Timesharing (multitasking)

- Providing each user with a **small** portion of a **time-shared** computer.
- Each user has at least one separate **program in memory**, called a **process**.
- Each **process** typically executes for only a **short time**.
- If several jobs ready to run at the same time → **CPU scheduling**

Dual-Mode Operation (1/2)

- ▶ The OS and the users share the hardware and software resources of the computer system.

Dual-Mode Operation (1/2)

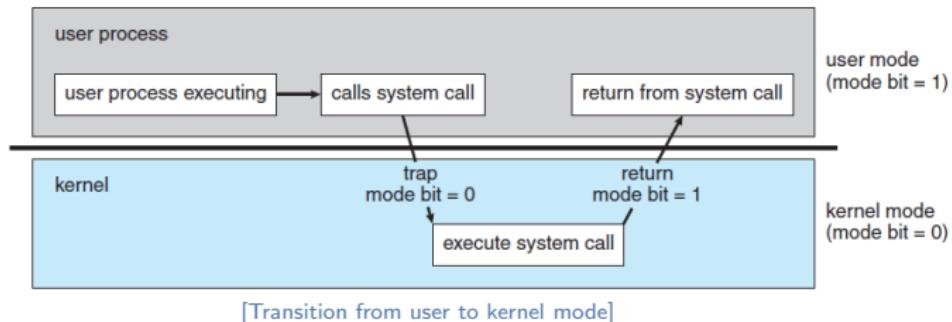
- ▶ The OS and the users share the hardware and software resources of the computer system.
- ▶ We need to make sure that an error in a user program could cause problems only for the one program running.
 - E.g., stuck on an infinite loop

Dual-Mode Operation (2/2)

- ▶ Dual-mode operation allows OS to protect itself and other system components.

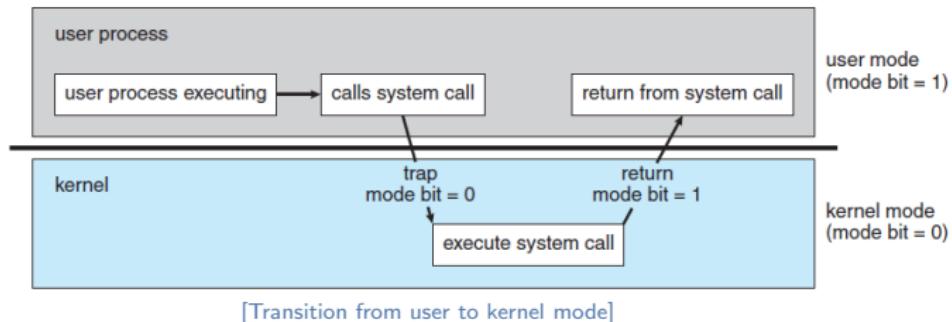
Dual-Mode Operation (2/2)

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

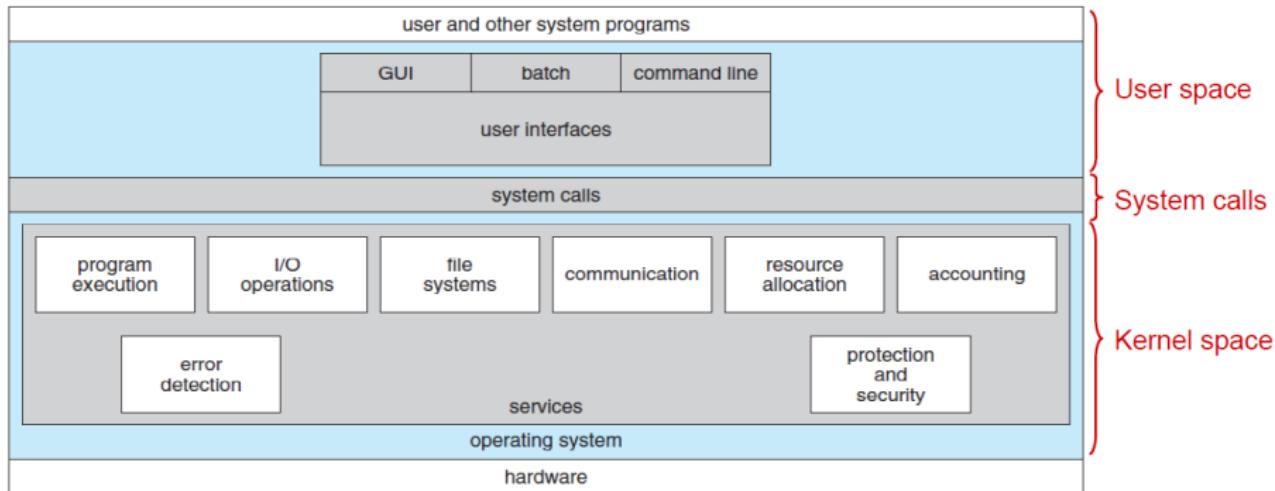


Dual-Mode Operation (2/2)

- ▶ Dual-mode operation allows OS to protect itself and other system components.
 - User mode and kernel mode.
 - System call changes mode to kernel, return from call resets it to user.



Operating System Structure



User Space

Programs

- ▶ **Kernel:** the **program** running at **all times** on a computer.

Programs

- ▶ **Kernel:** the **program** running at **all times** on a computer.
- ▶ Everything else is either:
 - a **system program**, or
 - an **application program**

System Programs (1/4)

- ▶ Most users' view of an OS is defined by system programs.

System Programs (1/4)

- ▶ Most users' view of an OS is defined by **system programs**.
- ▶ System programs include:
 - File manipulation
 - Status information
 - Programming language support
 - Communications
 - Background services

► File management

- Create, delete, copy, rename, print, dump, list, and generally **manipulate files** and **directories**.

► File management

- Create, delete, copy, rename, print, dump, list, and generally **manipulate files** and **directories**.

► File modification

- Text editors to create and modify files.
- Special commands to **search contents** of files or perform **transformations** of the text.

► Status information

- Asking the `system info`: date, time, amount of available memory, ...

► Status information

- Asking the `system info`: date, time, amount of available memory, ...

► Programming-language support

- `Compilers`, `assemblers`, `debuggers` and `interpreters` sometimes provided.

► Communications

- A mechanism to make **connections** among **processes**, **users**, and **computers**.

► Communications

- A mechanism to make **connections** among **processes**, **users**, and **computers**.

► Background services

- Launch at **boot time**, called **services** or **daemons**.
- Provide facilities like disk checking, process scheduling, error logging, printing.

Application Programs

- ▶ Run by users.
- ▶ Not typically considered part of OS.
- ▶ Launched by command line, mouse click, finger poke.
- ▶ Web browsers, word processors, database systems, compilers, games, ...



User Operating System Interface

- ▶ Almost all operating systems have a **user interface** (UI).
 - Command Line Interface (CLI)
 - Graphics User Interface (GUI)
 - Batch

Command Line Interface (1/2)

- ▶ Allows direct command entry.

Command Line Interface (1/2)

- ▶ Allows direct command entry.
- ▶ Sometimes implemented in **kernel**, sometimes by **systems program**.

Command Line Interface (1/2)

- ▶ Allows direct command entry.
- ▶ Sometimes implemented in **kernel**, sometimes by **systems program**.
- ▶ Sometimes commands **built-in**, sometimes just **names of programs**.

Command Line Interface (1/2)

- ▶ Allows direct command entry.
- ▶ Sometimes implemented in **kernel**, sometimes by **systems program**.
- ▶ Sometimes commands **built-in**, sometimes just **names of programs**.
- ▶ Sometimes multiple flavors implemented - **shell**.

Command Line Interface (2/2)

```
amir@rakhsh: ~
File Edit View Search Terminal Help
total 1893048
drwxrwxr-x 2 amir amir      4096 May 13 15:01 3gsim
drwxrwxr-x 3 amir amir      4096 Mar 19 2013 backup
-rw----- 1 amir amir 1936458240 Jun 12 18:58 csl.ova
drwxr-xr-x 2 amir amir      4096 Apr 15 16:39 Desktop
drwxr-xr-x 5 amir amir      4096 Jun 12 19:00 Documents
drwxr-xr-x 16 amir amir     40960 Jul 30 12:07 Downloads
drwx----- 13 amir amir      4096 Jul 25 08:18 Dropbox
drwxr-xr-x 7 amir amir      4096 Mar 14 17:02 Music
drwxr-xr-x 24 amir amir     4096 Jun 23 13:50 Pictures
drwxrwxr-x 11 amir amir     4096 Mar 25 14:01 Projects
drwxr-xr-x 13 amir amir     4096 Jun 12 19:02 Public
drwxr-xr-x 23 amir amir     4096 Nov  5 2010 Src
drwxrwxr-x 5 amir amir      4096 Apr 30 14:08 Temp
drwxr-xr-x 2 amir amir      4096 May 24 2012 Templates
drwxr-xr-x 5 amir amir      4096 Jun 18 16:22 Tools
drwxrwxr-x 6 amir amir      4096 May 30 15:08 trafmod
drwxrwxr-x 2 amir amir      4096 May 31 2012 Ubuntu One
drwxr-xr-x 4 amir amir      4096 Sep 17 2013 University
drwxr-xr-x 2 amir amir      4096 May 24 2012 Videos
drwx----- 5 amir amir      4096 Jun 12 19:00 VirtualBox VMs
drwxrwxr-x 7 amir amir      4096 May 14 11:44 webtrafmod
drwxr-xr-x 25 amir amir     4096 Nov  5 2013 workspace
amir@rakhsh:~ w
13:06:17 up 18 days, 22:49,  3 users,  load average: 0.23, 0.54, 0.70
USER   TTY      FROM             LOGIN@    IDLE   JCPU   PCPU WHAT
amir   pts/0    :0.0          11Jul14 18days  2:47m  3.42s gnome-session --session=g
amir   pts/1    :0.0          08:44     1.00s  0.35s  0.00s w
amir   pts/1    :0.0          10:56     2:10m   0.21s  0.21s /bin/bash
amir@rakhsh:~
```

Graphical Line Interface (1/2)

- ▶ User-friendly desktop metaphor interface.

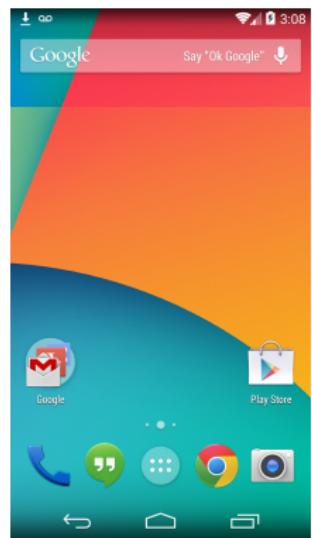
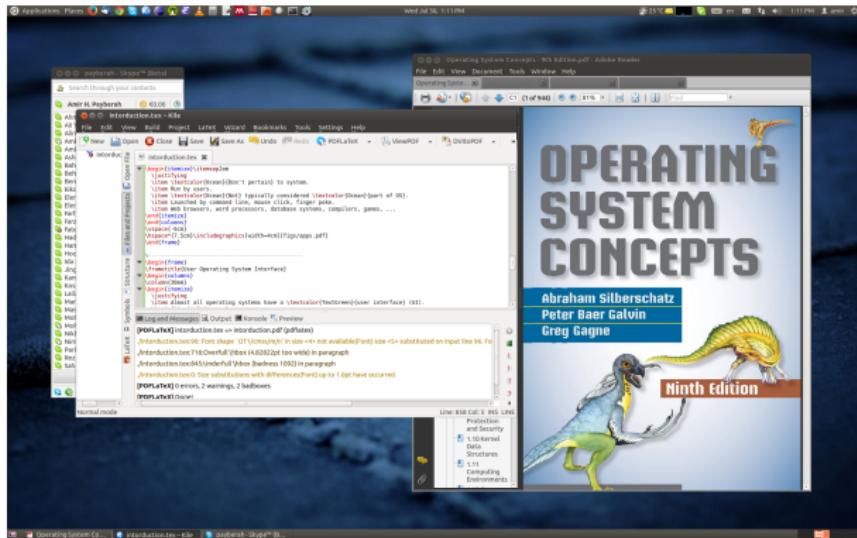
Graphical Line Interface (1/2)

- ▶ **User-friendly** desktop metaphor interface.
- ▶ **Icons** represent files, programs, actions, ...

Graphical Line Interface (1/2)

- ▶ User-friendly desktop metaphor interface.
- ▶ Icons represent files, programs, actions, ...
- ▶ Various mouse buttons over objects in the interface cause various actions.

Graphical Line Interface (2/2)

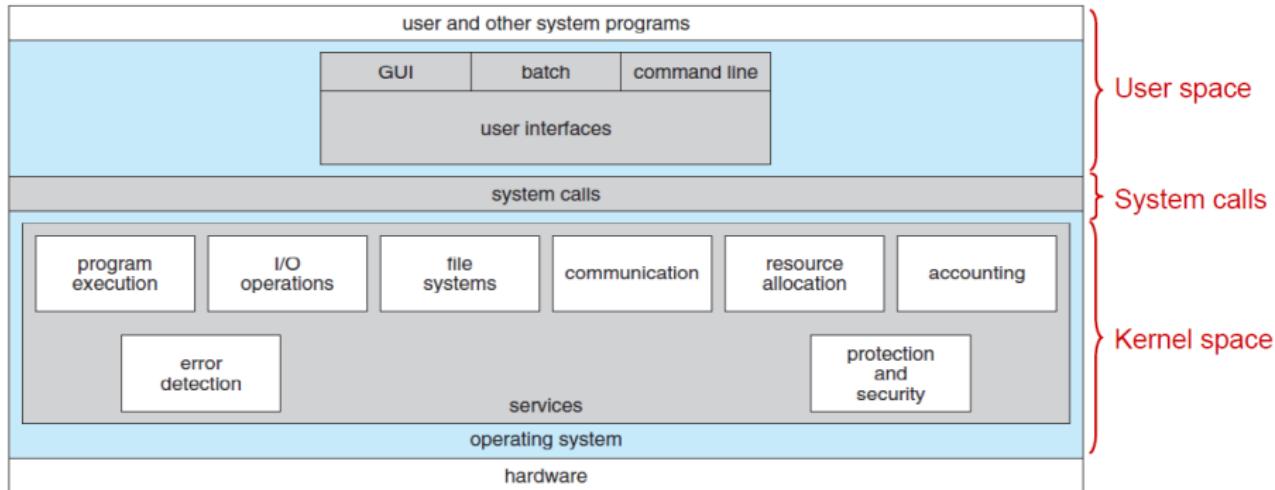


Batch Interface

- ▶ Commands and directives to control those commands are entered into files, and those files are executed.

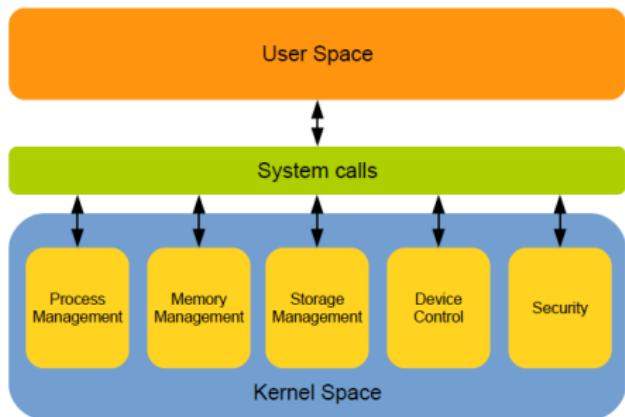
Kernel Space

Operating System Structure



Splitting the Kernel

- ▶ The kernel's role can be **split** into the following parts
 - Process management
 - Memory management
 - Storage management and File system
 - Device control and I/O subsystem
 - Protection and security



Process Management (1/2)

- ▶ A **process** is a **program** in **execution**.

Process Management (1/2)

- ▶ A **process** is a **program** in **execution**.
 - **Program** is a **passive entity**, **process** is an **active entity**.

- ▶ A **process** is a **program** in **execution**.
 - **Program** is a **passive entity**, **process** is an **active entity**.
- ▶ A process needs **resources** to accomplish its task.
 - CPU, memory, I/O, files, initialization data, ...

Process Management (1/2)

- ▶ A **process** is a **program** in **execution**.
 - **Program** is a **passive entity**, **process** is an **active entity**.
- ▶ A process needs **resources** to accomplish its task.
 - CPU, memory, I/O, files, initialization data, ...
- ▶ Process **termination** requires **reclaim** of any reusable resources.

Process Management (2/2)

- ▶ Process management activities:

Process Management (2/2)

- ▶ **Process management** activities:
 - **Scheduling** processes and threads on the CPUs.

- ▶ **Process management** activities:
 - **Scheduling** processes and threads on the CPUs.
 - **Creating** and **deleting** both user and system processes.

- ▶ **Process management** activities:
 - **Scheduling** processes and threads on the CPUs.
 - **Creating** and **deleting** both user and system processes.
 - **Suspending** and **resuming** processes.

- ▶ **Process management** activities:
 - **Scheduling** processes and threads on the CPUs.
 - **Creating** and **deleting** both user and system processes.
 - **Suspending** and **resuming** processes.
 - Providing mechanisms for process **synchronization**.

► **Process management** activities:

- **Scheduling** processes and threads on the CPUs.
- **Creating** and **deleting** both user and system processes.
- **Suspending** and **resuming** processes.
- Providing mechanisms for process **synchronization**.
- Providing mechanisms for process **communication**.

Memory Management (1/2)

- ▶ To execute a **program** all (or part) of the **instructions** must be in **memory**.

Memory Management (1/2)

- ▶ To execute a **program** all (or part) of the **instructions** must be in **memory**.
- ▶ All (or part) of the **data** that is needed by the program must be in **memory**.

Memory Management (1/2)

- ▶ To execute a **program** all (or part) of the **instructions** must be in **memory**.
- ▶ All (or part) of the **data** that is needed by the program must be in **memory**.
- ▶ **Memory management** determines **what** is in memory and **when**.

Memory Management (2/2)

- ▶ Memory management activities:

Memory Management (2/2)

► Memory management activities:

- Keeping track of which parts of memory are currently being used and by whom.

Memory Management (2/2)

► Memory management activities:

- Keeping track of which parts of memory are currently being used and by whom.
- Deciding which processes (or parts of) and data to move into and out of memory.

Memory Management (2/2)

► Memory management activities:

- Keeping track of which parts of memory are currently being used and by whom.
- Deciding which processes (or parts of) and data to move into and out of memory.
- Allocating and deallocated memory space as needed.

Storage Management (1/5)

- ▶ OS provides **uniform** and **logical view** of **information** storage.

Storage Management (1/5)

- ▶ OS provides **uniform** and **logical view** of **information** storage.
- ▶ OS abstracts **physical** properties to **logical** storage unit, called **file**.

Storage Management (1/5)

- ▶ OS provides **uniform** and **logical view** of **information** storage.
- ▶ OS abstracts **physical** properties to **logical** storage unit, called **file**.
 - A **file** is a **collection** of **related information** (programs or data).

Storage Management (1/5)

- ▶ OS provides **uniform** and **logical view** of **information** storage.
- ▶ OS abstracts **physical** properties to **logical** storage unit, called **file**.
 - A **file** is a **collection** of **related information** (programs or data).
 - **Files** usually organized into **directories**.

Storage Management (1/5)

- ▶ OS provides **uniform** and **logical view** of **information** storage.
- ▶ OS abstracts **physical** properties to **logical** storage unit, called **file**.
 - A **file** is a **collection** of **related information** (programs or data).
 - **Files** usually organized into **directories**.
- ▶ OS maps files onto **physical media** and accesses these files via the **storage devices**, e.g., disk drive, tape drive.

Storage Management (2/5)

- ▶ File management activities:

Storage Management (2/5)

- ▶ **File management** activities:

- **Creating** and **deleting** **files** and **directories**.

Storage Management (2/5)

- ▶ **File management** activities:

- **Creating** and **deleting files** and **directories**.
- Primitives to **manipulate** files and directories.

Storage Management (2/5)

- ▶ **File management** activities:

- **Creating** and **deleting files** and **directories**.
- Primitives to **manipulate** files and directories.
- **Mapping** files onto secondary storage.

► **File management** activities:

- **Creating** and **deleting files** and **directories**.
- Primitives to **manipulate** files and directories.
- **Mapping** files onto secondary storage.
- **Backup** files onto stable (non-volatile) storage media.

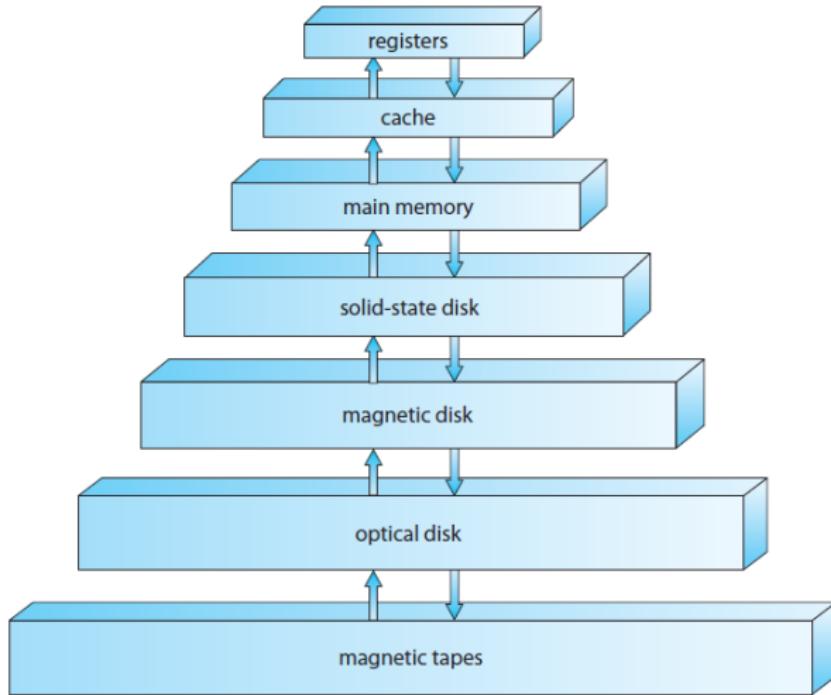
Storage Management (3/5)

- ▶ Usually **disks** used to **store** data that does not fit in main memory or data that must be kept for a **long** period of time.

Storage Management (3/5)

- ▶ Usually **disks** used to **store** data that does not fit in main memory or data that must be kept for a **long** period of time.
- ▶ **Disk management** activities:
 - Free-space management
 - Storage allocation
 - Disk scheduling

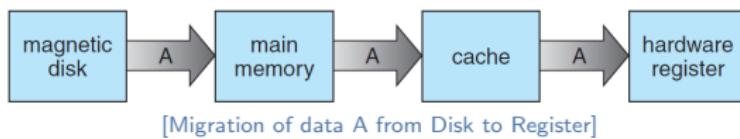
Storage Management (4/5)



[Cost vs. Speed of various levels of storage]

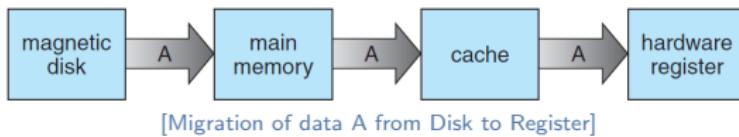
Storage Management (5/5)

- ▶ Caching
- ▶ Copying information from **slower** to **faster** storage temporarily.



Storage Management (5/5)

- ▶ **Caching**
- ▶ Copying information from **slower** to **faster** storage temporarily.
- ▶ Faster storage (**cache**) checked first to determine if information is there
 - If it is, information used directly from the cache (**fast**).
 - If not, data copied to cache and used there.



I/O Subsystem

- ▶ Hides the details of the hardware devices from users.
- ▶ The I/O subsystem consists of several components, e.g., device drivers for specific hardware devices.

- ▶ **Protection:** any mechanism for controlling **access** of **processes** or **users** to **resources** defined by the OS.

- ▶ **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**.
 - E.g., denial-of-service, worms, viruses, identity theft, theft of service, ...

Protection and Security (2/2)

- ▶ Systems generally first distinguish among **users**, to determine **who can do what**.

Protection and Security (2/2)

- ▶ Systems generally first distinguish among **users**, to determine **who can do what**.
- ▶ **User IDs** include name and associated number, **one per user**.
 - The User ID is then **associated** with all **files** and **processes** of that user to determine **access control**.

Protection and Security (2/2)

- ▶ Systems generally first distinguish among **users**, to determine **who can do what**.
- ▶ **User IDs** include name and associated number, **one per user**.
 - The User ID is then **associated** with all **files** and **processes** of that user to determine **access control**.
- ▶ **Group ID** allows **set of users** to be defined and controls managed, then also associated with each process, file.

Summary

Summary

- ▶ Computer-system architecture: single-processor, multiprocessor (multi-core, blade), clustered

Summary

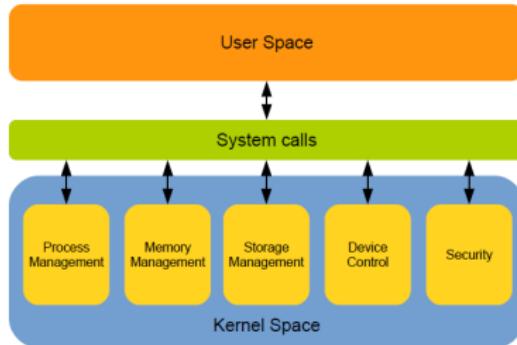
- ▶ **Computer-system architecture:** single-processor, multiprocessor (multi-core, blade), clustered
- ▶ **Operating-system structure:** user-space, system calls, kernel-space

Summary

- ▶ **Computer-system architecture:** single-processor, multiprocessor (multi-core, blade), clustered
- ▶ **Operating-system structure:** user-space, system calls, kernel-space
- ▶ **User-space:** system programs, application programs
- ▶ **Kernel-space**

Summary

- ▶ **Computer-system architecture:** single-processor, multiprocessor (multi-core, blade), clustered
- ▶ **Operating-system structure:** user-space, system calls, kernel-space
- ▶ **User-space:** system programs, application programs
- ▶ **Kernel-space**



Questions?

Acknowledgements

Some slides were derived from Avi Silberschatz slides.