Introduction to Operating System

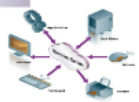
What OS do you use?

- Linux
- Windows
- Mac
- Unix
- Solaris

Roles of OS

Managing multi tasks

Managing resources



Syllabus

- Introduction to Operating Systems
- Process Management
- Concurrent Management
- Critical Section Problem
- Dead Lock
- Memory Mangement
- Cache Memory
- Secondary Storage Memories
- Buffer Cache
- File Representation
- File System Architecture
- Introduction to Distribution Systems
- Conclusion

Online Resources

P.K Biswas: IIT KGP

Kishore Ramachandran, Georgia Tech

https://www.udacity.com/wiki/ud156

Prior Knowledge

Basics

- Data Structure
- Object Oriented Programming
- Programming C and Java

Evaluation Scheme

- Mid sems: 30%
- End sems: 30%
- Programming assignments: 20% (4-5)
- Quiz: 10% (2)
- Class participation: 10%

Attendance Policy

Attendance will be taken everyday and missing class can be expected to significantly reduce your chances of success. There will be no repetition.

Missing Exams

- If you miss a exam due to an unexcused absence, you will receive a grade of 0 for that quiz/exam.
- The make-up exam may be **SIGNIFICANTLY MORE DIFFICULT** than the original exam.

Office Hours

- Wednesday 2:00-5:00 PM

What is an Operating System?

What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware

Operating system goals:

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

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
  - 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

Operating System Definition

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer

Operating System Definition (Cont)

- No universally accepted definition
- "Everything a vendor ships when you order an operating system" is good approximation
  - But varies wildly
- "The one program running at all times on the computer" is the **kernel**. Everything else is either a system program (ships with the operating system) or an application program

Computer Startup

- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Operating System Structure

- **Multiprogramming** needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - In multiprogramming, several programs are in memory concurrently; the system switches among the programs for efficient processing, and minimal idle time.
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - 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

Operating-System Operations

- An operating system is **interrupt driven**
- **Interrupt** driven by hardware
- **exception** or **trap** is a software-generated interrupt caused either
  - by an error (ex: Division by zero or invalid memory access)
  - Or by a request from a user program for operating system service →**System call**
- Other process problems include infinite loop, processes modifying each other or the operating system
- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **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

Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a passive entity, process is an active entity.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- 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

Memory Management

- All data in memory before and after processing
- All instructions in memory in order to execute
- Memory management determines what is in memory when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Primitives to manipulate files and dirs
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - 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
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights

Special-Purpose OS

- **Real-Time**

  Often used in a dedicated application. The system reads information from sensors and must respond within a fixed amount of time to ensure correct performance.

- **Embedded**
  - Special purpose systems
    - Cell phones, wireless routers, TV's, space vehicles, etc.
  - Requirements
    - High reliability, difficult or impossible to upgrade after deployed, run in hostile environments, self diagnosis and repair, unattended operation.

Summary

Introduction to Operating System

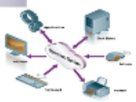
What OS do you use?

- Linux
- Windows
- Mac
- Unix
- Solaris

Roles of OS

Managing multi tasks

Managing resources



Syllabus

- Introduction to Operating Systems
- Process Management
- Concurrent Management
- Critical Section Problem
- Dead Lock
- Memory Mangement
- Cache Memory
- Secondary Storage Memories
- Buffer Cache
- File Representation
- File System Architecture
- Introduction to Distribution Systems
- Conclusion

Online Resources

P.K Biswas: IIT KGP

Kishore Ramachandran, Georgia Tech

Prior Knowledge

Basics

- Data Structure
- Object Oriented Programming
- Programming C and Java

Evaluation Scheme

- Mid sems: 30%
- End sems: 30%
- Programming assignments: 20% (4-5)
- Quiz: 10% (2)
- Class participation: 10%

Attendance Policy

Attendance will be taken everyday and missing class can be expected to significantly reduce your chances of success. There will be no repetition.

Missing Exams

- If you miss a exam due to an unexcused absence, you will receive a grade of 0 for that quiz/exam.
- The make-up exam may be **SIGNIFICANTLY MORE DIFFICULT** than the original exam.

Office Hours

- Wednesday 2:00-5:00 PM

What is an Operating System?

What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware

Operating system goals:

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

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
  - 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

Operating System Definition

- OS is a resource allocator
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a control program
  - Controls execution of programs to prevent errors and improper use of the computer

Operating System Definition (Cont)

- No universally accepted definition
- "Everything a vendor ships when you order an operating system" is good approximation
  - But varies wildly
- "The one program running at all times on the computer" is the kernel. Everything else is either a system program (ships with the operating system) or an application program

Computer Startup

- bootstrap program is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as firmware
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Operating System Structure

- Multiprogramming needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - In multiprogramming, several programs are in memory concurrently; the system switches among the programs for efficient processing, and minimal idle time.
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - 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

Operating-System Operations

- An operating system is interrupt driven
- Interrupt driven by hardware
- exception or trap is a software-generated interrupt caused either
  - by an error (ex: Division by zero or invalid memory access)
  - Or by a request from a user program for operating system service →System call
- Other process problems include infinite loop, processes modifying each other or the operating system
- Dual-mode operation allows OS to protect itself and other system components
  - User mode and kernel mode
  - 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

Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a passive entity, process is an active entity.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one program counter specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- 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

Memory Management

- All data in memory before and after processing
- All instructions in memory in order to execute
- Memory management determines what is in memory when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - file
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Primitives to manipulate files and dirs
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - 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
- Systems generally first distinguish among users, to determine who can do what
  - User identities (user IDs, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (group ID) allows set of users to be defined and controls managed, then also associated with each process, file
  - Privilege escalation allows user to change to effective ID with more rights

Special-Purpose OS

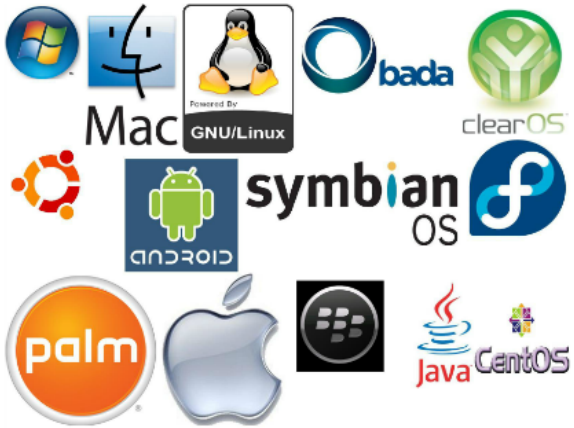
- Real-Time

  Often used in a dedicated application. The system reads information from sensors and must respond within a fixed amount of time to ensure correct performance.

- Embedded
  - Special purpose systems
    - Cell phones, wireless routers, TV' s, space vehicles, etc.
  - Requirements
    - High reliability, difficult or impossible to upgrade after deployed, run in hostile environments, self diagnosis and repair, unattended operation.

Summary

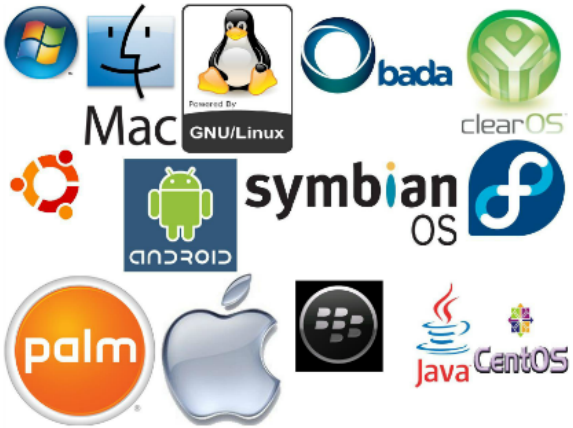
What OS do you use?



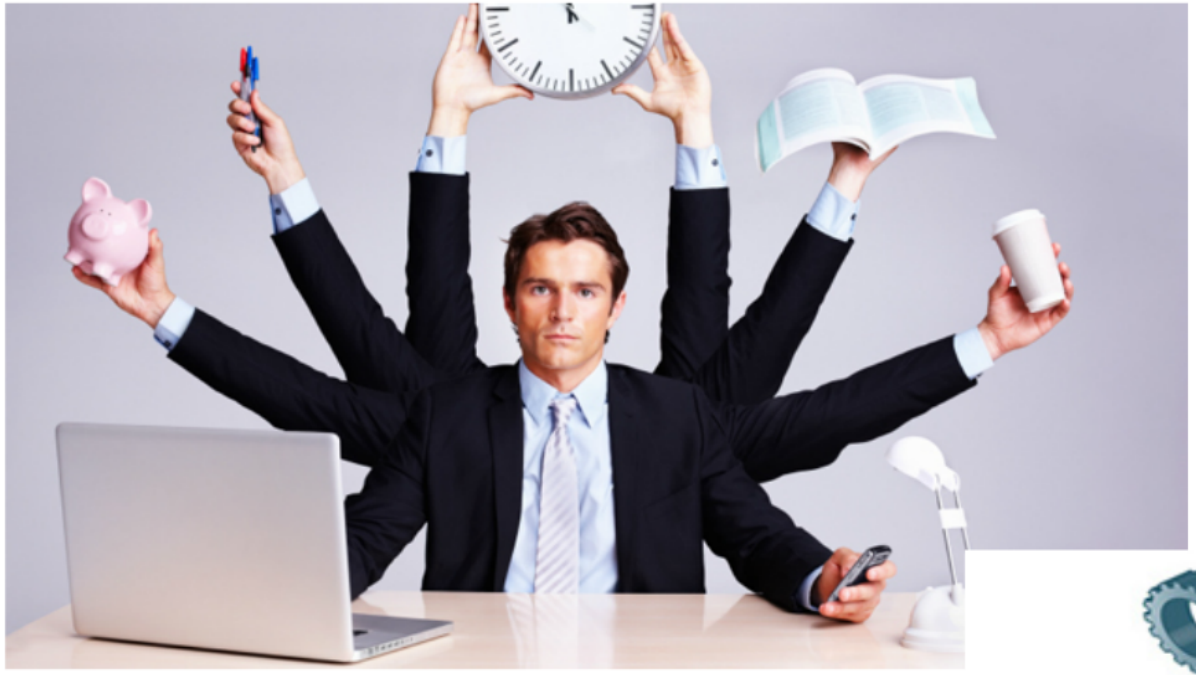
- Linux
- Windows
- Mac
- Unix
- Solaris

S

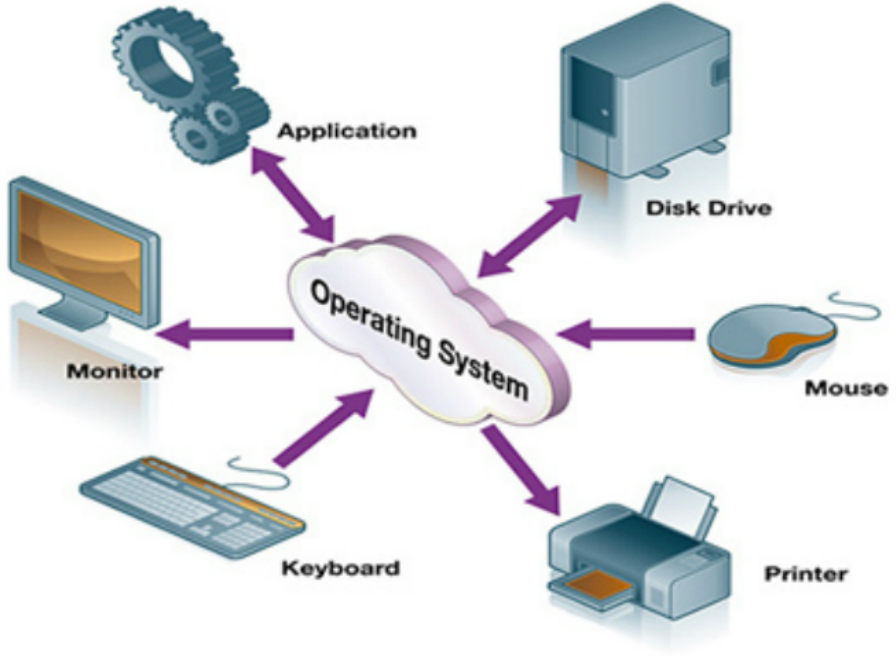
Roles of OS



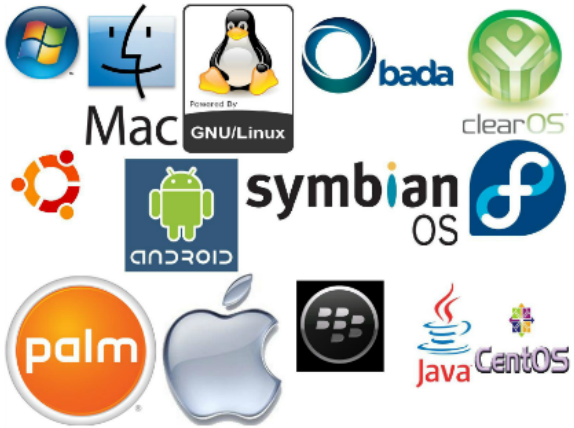
Managing multi tasks



Managing resources



Syllabus



- Introduction to Operating Systems
- Process Management
- Concurrent Management
- Critical Section Problem
- Dead Lock
- Memory Mangement
- Cache Memory
- Secondary Storage Memories
- Buffer Cache
- File Representation
- File System Architecture
- Introduction to Distribution Systems
- Conclusion

Online Resources

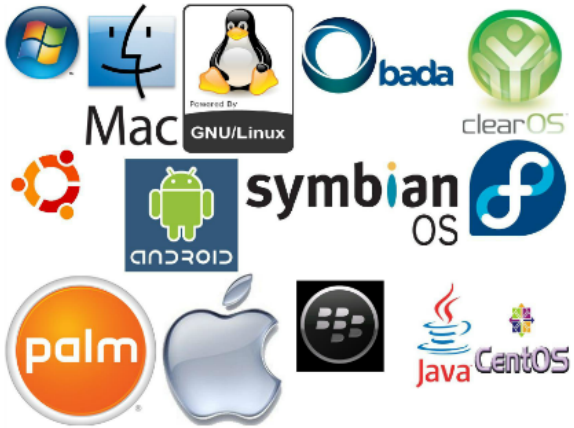
P.K.Biswas: IIT KGP

https://www.youtube.com/playlist?list=PLLDC70psjvq5hIT0kfr1sirNuees0NIbG

Kishore Ramachandran, Georgia Tech

https://www.udacity.com/wiki/ud156

Prior Knowledge

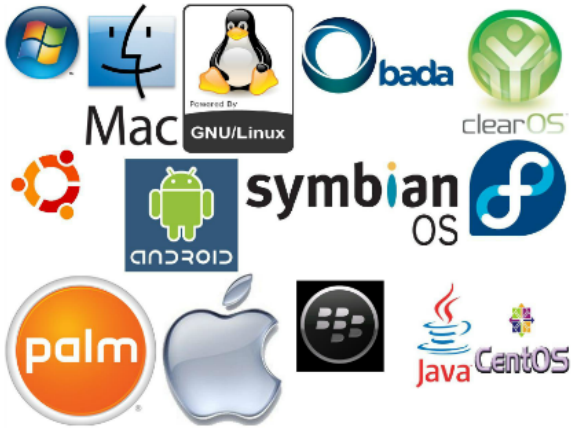


Basics

- Data Structure
- Object Oriented Programming
- Programming C and Java

Evaluation Scheme

- Mid sems: 30%
- End sems: 30%
- Programming assignments: 20% (4-5)
- Quiz: 10% (2)
- Class participation: 10%



Attendance Policy

Attendance will be taken everyday and missing class can be expected to significantly reduce your chances of success. There will be no repetition.

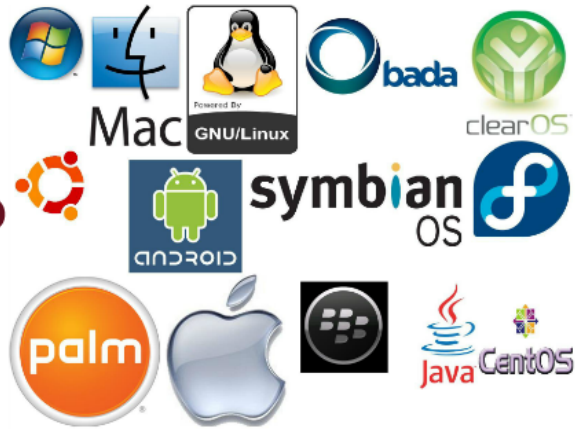
Missing Exams

- If you miss a exam due to an unexcused absence, you will receive a grade of 0 for that quiz/exam.
- The make-up exam may be **SIGNIFICANTLY MORE DIFFICULT** than the original exam.

Office Hours

- Wednesday 2:00-5:00 PM

What is an Operating System?

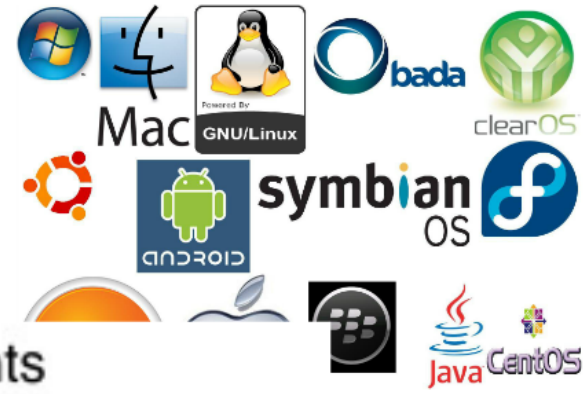


- A program that acts as an intermediary between a user of a computer and the computer hardware

Operating system goals:

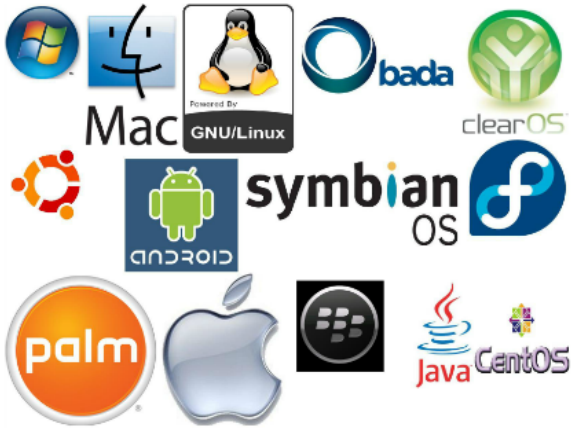
- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

Computer System Structure

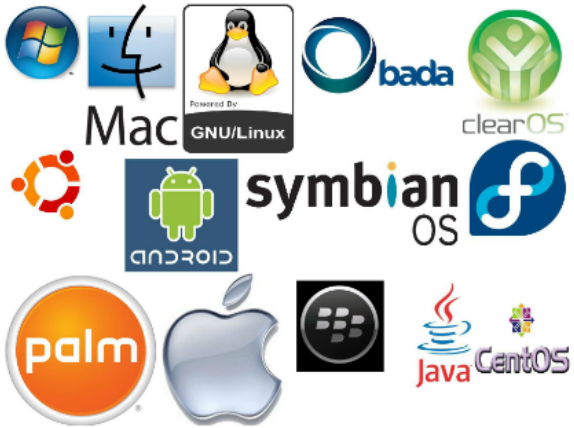


- Computer system can be divided into four components
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system 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

Operating System Definition

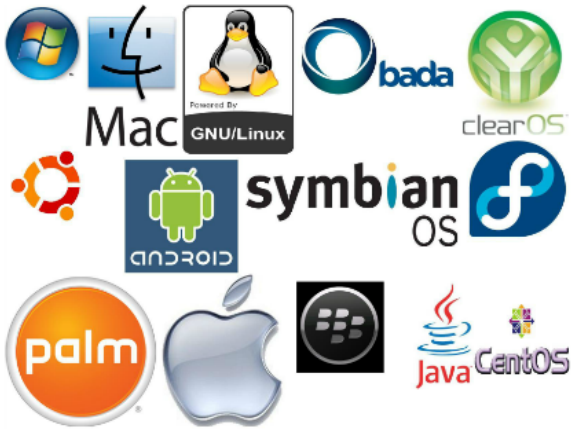


- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer



Operating System Definition (Cont)

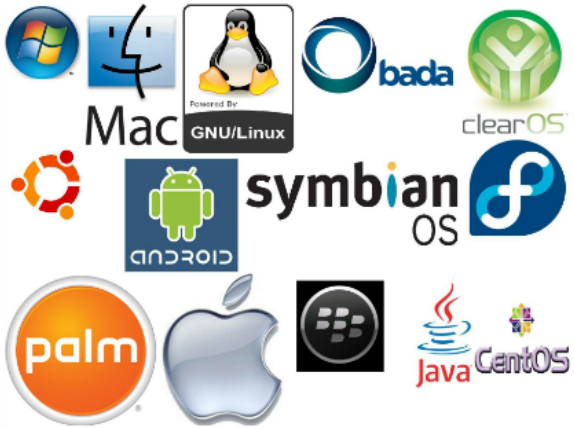
- No universally accepted definition
- "Everything a vendor ships when you order an operating system" is good approximation
  - But varies wildly
- "The one program running at all times on the computer" is the **kernel**. Everything else is either a system program (ships with the operating system) or an application program



Computer Startup

- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Operating System Structure

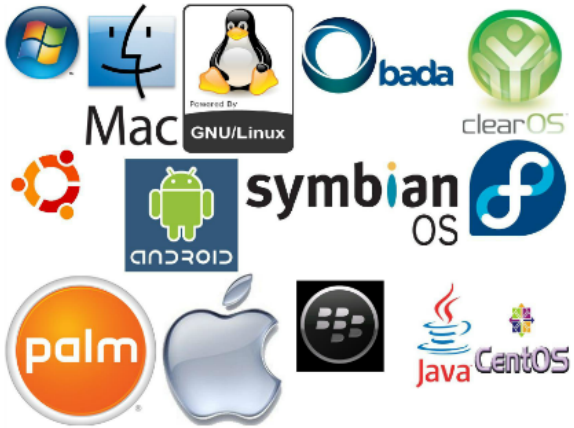


- **Multiprogramming** needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - In multiprogramming, several programs are in memory concurrently; the system switches among the programs for efficient processing, and minimal idle time.
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - 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

Operating-System Operations

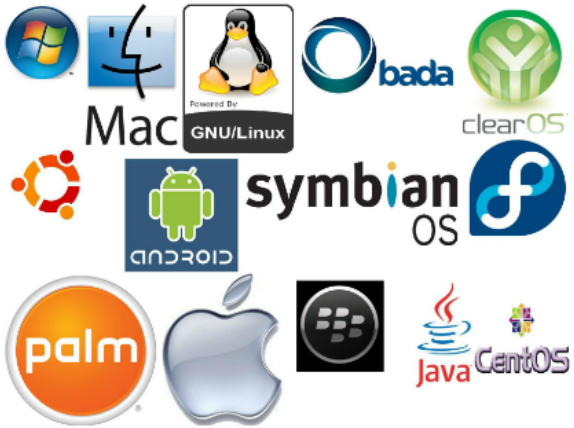
- An operating system is **interrupt driven**
- **Interrupt** driven by hardware
- **exception** or **trap** is a software-generated interrupt caused either
  - by an error (ex: Division by zero or invalid memory access)
  - Or by a request from a user program for operating system service →**System call**
- Other process problems include infinite loop, processes modifying each other or the operating system
- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **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

Process Management



- A process is a program in execution. It is a unit of work within the system. Program is a *passive entity*, process is an *active entity*.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- 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

Memory Management

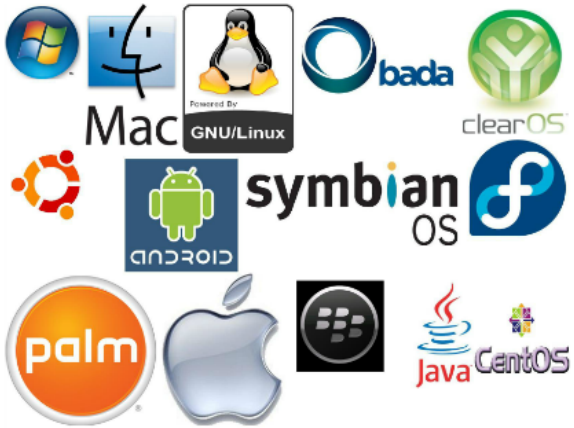


- All data in memory before and after processing
- All instructions in memory in order to execute
- Memory management determines what is in memory when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Primitives to manipulate files and dirs
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

I/O Subsystem

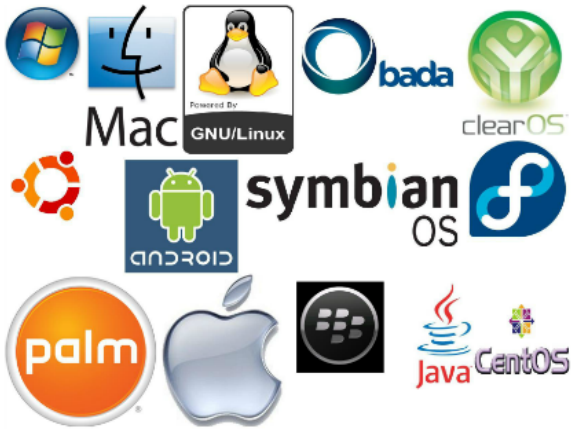


- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - 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
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights

Special-Purpose OS

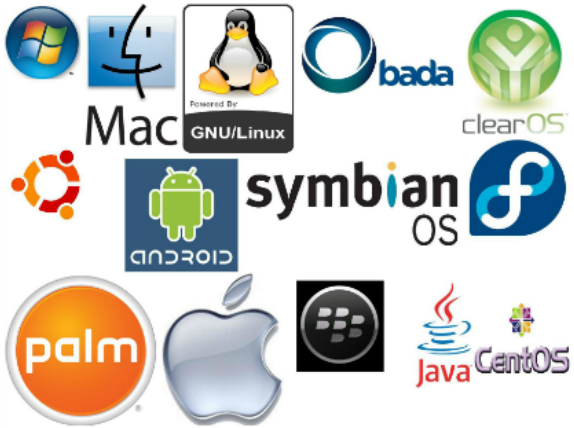


- **Real-Time**

  Often used in a dedicated application. The system reads information from sensors and must respond within a fixed amount of time to ensure correct performance.

- **Embedded**
  - Special purpose systems
    - Cell phones, wireless routers, TV's, space vehicles, etc.
  - Requirements
    - High reliability, difficult or impossible to upgrade after deployed, run in hostile environments, self-diagnosis and repair, unattended operation.

Summary


Mac
GNU/Linux
bada
clearOS
symbian
OS
android
palm
Java
CentOS