



CS-3103 : Operating Systems : Sec-A (NB) :
Operating-System Structures

- Goals: Provide a way to understand an operating systems
 - Services
 - Interface
 - System Components
- The type of system desired is the basis for choices among various algorithms and strategies!

OS Services Helpful to the User

■ **User interface** - Almost all operating systems have a **user interface (UI)**. This interface can take several forms:

- **Command-Line (CLI)** -- uses text commands and a method for entering them (say, a keyboard for typing in commands in a specific format with specific options).
- **Graphics User Interface (GUI)** -- the interface is a window system with a pointing device to direct I/O, choose from menus, and make selections and a keyboard to enter text..
- **Batch Interface** -- commands and directives to control those commands are entered into files, and those files are executed

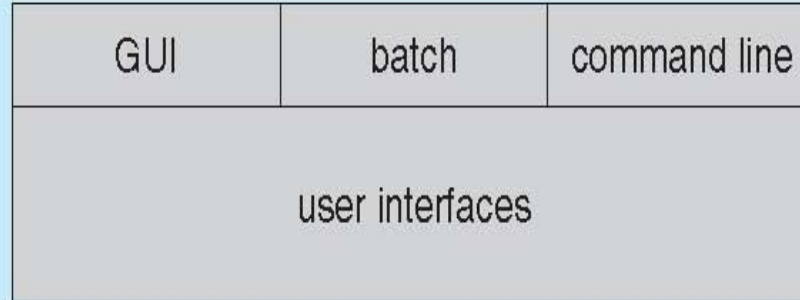
Some systems provide two or all three of these variations.

■ **Program execution** - The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error)

■ **I/O operations** - A running program may require I/O, which may involve a file or an I/O device

A View of Operating System Services

user and other system programs



system calls

program
execution

I/O
operations

file
systems

communication

resource
allocation

accounting

error
detection

protection
and
security

services

operating system

hardware

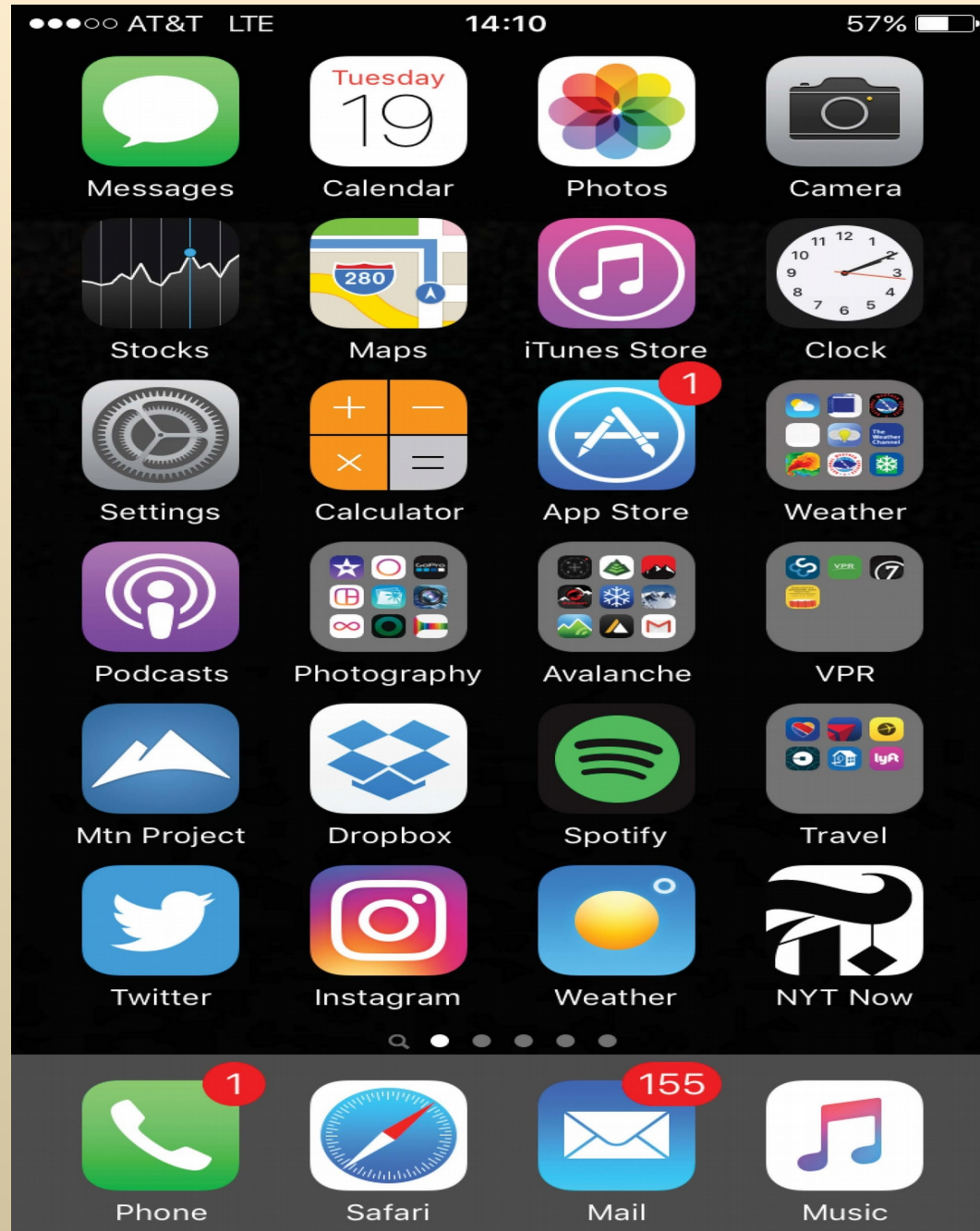
Command Interpreters (CLI)

CLI allows users to directly enter commands to be performed by the operating system.

- Some operating systems include the command interpreter in the kernel.
- Some operating systems, such as Windows and UNIX, treat the command interpreter as a special program that is running when a job is initiated or when a user first logs on.
- On systems with multiple command interpreters to choose from, the interpreters are known as **shells**.
- The main function of the command interpreter is to get and execute the next user-specified command.
- Sometimes commands built-in, sometimes just names of programs
 - If the latter, adding new features doesn't require shell modification

Touchscreen Interfaces

- Touchscreen devices require new interfaces
 - Mouse not possible or not desired
 - Actions and selection based on gestures
 - Virtual keyboard for text entry
 - Voice commands.



System Calls

- Programming interface to the services provided by the OS
- Typically written in a high-level language (C or C++)
- Mostly accessed by programs via a high-level **Application Programming Interface (API)** rather than direct system call
- Three most common APIs are:
 - Win32 API for Windows,
 - POSIX API for POSIX-based systems (including virtually all versions of UNIX, Linux, and Mac OS X),
 - Java API for the Java virtual machine (JVM)

Examples of Windows and Unix System Calls

	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

Android

- Developed by Open Handset Alliance (mostly Google)
 - Open Source
- Similar stack to iOS
- Based on Linux kernel but modified
 - Provides process, memory, device-driver management
 - Adds power management
- Runtime environment includes core set of libraries and Dalvik virtual machine
 - Apps developed in Java plus Android API
 - ▶ Java class files compiled to Java bytecode then translated to executable then runs in Dalvik VM
- Libraries include frameworks for web browser (webkit), database (SQLite), multimedia, smaller libc

Android Architecture

Applications

Application Framework

Libraries

SQLite

OpenGL

surface
manager

media
framework

webkit

libc

Android runtime

Core Libraries

Dalvik
virtual machine

Linux kernel

Performance Tuning: Example, “top” program or Windows Task Manager

Task Manager					
File Options View					
Processes Performance App history Startup Users Details Services					
Name	Status	27% CPU	62% Memory	0% Disk	0% Network
Apps (6)					
▶ Adobe Acrobat Reader DC (32 bit)		0%	18.4 MB	0 MB/s	0 Mbps
▶ Google Chrome (2)		0%	61.1 MB	0 MB/s	0 Mbps
▶ Microsoft Office PowerPoint (32...		0%	2.0 MB	0 MB/s	0 Mbps
▶ Paint		0%	0.1 MB	0 MB/s	0 Mbps
▶ Task Manager		1.2%	7.8 MB	0.1 MB/s	0 Mbps
▶ Windows Explorer		0%	22.5 MB	0 MB/s	0 Mbps
Background processes (34)					
▶ Adobe Acrobat Update Service (...)		0%	0.1 MB	0 MB/s	0 Mbps
Adobe RdrCEF (32 bit)		0%	0.5 MB	0 MB/s	0 Mbps
Adobe RdrCEF (32 bit)		0%	11.3 MB	0 MB/s	0 Mbps
Adobe RdrCEF (32 bit)		0%	2.5 MB	0 MB/s	0 Mbps
▶ Adobe® Flash® Player Feedbac...		0%	1.0 MB	0 MB/s	0 Mbps

System Components – Process Management

- Process Management
 - Process: An Active Entity
 - Physical and Logical Resources
 - Memory, I/O buffers, data, etc.
 - Data Structures Representing Current Activities:

Program
(code)

+

Program Counter
Stack

Data Section

CPU Registers

....

And More

System Components – Process Management

- Services
 - Process creation and deletion
 - Process suspension and resumption
 - Process synchronization
 - Process communication
 - Deadlock handling

System Components – Main-Memory Management

- Memory: a large array of words or bytes, where each has its own address
- OS must keep several programs in memory to improve CPU utilization and user response time
- Management algorithms depend on the hardware support
- Services
 - Memory usage and availability
 - Decision of memory assignment
 - Memory allocation and deallocation

System Components – File Management

- Goal:
 - A uniform logical view of information storage
 - Each medium controlled by a device
 - Magnetic tapes, magnetic disks, optical disks, etc.
- OS provides a logical storage unit: File
 - Formats:
 - Free form or being formatted rigidly.
 - General Views:
 - A sequence of bits, bytes, lines, records

System Components – File Management

- Services
 - File creation and deletion
 - Directory creation and deletion
 - Primitives for file and directory manipulation
 - Mapping of files onto secondary storage
 - File Backup

System Components – I/O System Management

- Goal:
 - Hide the peculiarities of specific hardware devices from users
- Components of an I/O System
 - A buffering, caching, and spooling system
 - A general device-driver interface
 - Drivers

System Components – Secondary-Storage Management

- Goal:
 - On-line storage medium for programs & data
 - Backup of main memory
- Services for Disk Management
 - Free-space management
 - Storage allocation, e.g., continuous allocation
 - Disk scheduling, e.g., FCFS

System Components – Networking

- Issues
 - Resources sharing
 - Routing & connection strategies
 - Contention and security
- Network access is usually generalized as a form of file access
 - World-Wide-Web over file-transfer protocol (ftp), network file-system (NFS), and hypertext transfer protocol (http)

System Components – Protection System

- Goal
 - Resources are only allowed to be accessed by authorized processes.
- Protected Resources
 - Files, CPU, memory space, etc.
- Services
 - Detection & controlling mechanisms
 - Specification mechanisms
- Remark: Reliability!