CSE308 Operating Systems

Operating System Structure

Dr S.Rajarajan

SoC

SASTRA

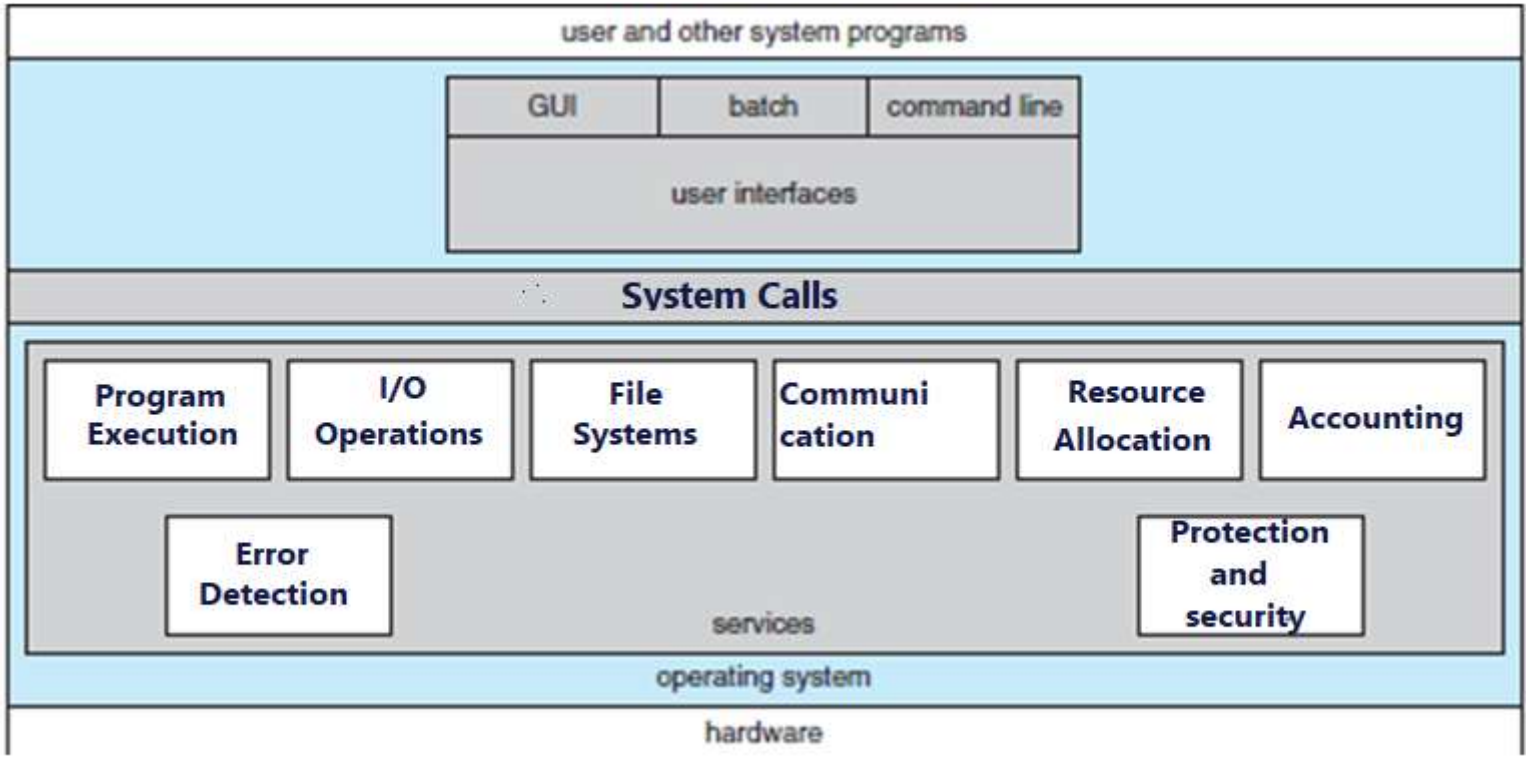
Topics

- Operating System Services
- User-Operating System Interface
- System Calls
- System Programs
- Operating System Design and Implementation
- Operating System Structure
- Virtual Machines
- Operating System Debugging
- Operating System Generation
- System Boot

OPERATING SYSTEM SERVICES

Operating System Services

- OS provide an **environment for execution of programs** and **offers services** to programs and users e.g fork, pipe, I/O
- Specific services provided **differ from one OS to another**, but we can identify **common classes**
- These operating system services are provided for the **convenience of the programmer**, to make the programming task easier.



User Services

- **User interface** - Almost all operating systems have a user interface (UI).
  - Varies between **Command-Line (CLI)**, **Graphics User Interface (GUI)**, **Batch**
- **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**
- **File-system manipulation** - Programs need to **read and write files** and **directories**, **create and delete them**, **search them**, list file Information, permission management.

Operating System Services (Cont.)

- **Communications(IPC)** – Processes may **exchange information**, on the **same computer** or between computers **over a network**
  - Communications may be via **shared memory** or through **message passing** (packets moved by the OS)
- **Error detection** – OS needs to be constantly aware of possible errors
  - May occur in **the CPU and memory hardware, in I/O devices, in user program**
  - For each type of error, OS should take the **appropriate action** to ensure correct and consistent computing
  - **Debugging facilities** can greatly enhance the user's and programmer's abilities to efficiently use the system

Resource Sharing Services

- **Resource allocation -** When multiple users or multiple jobs running concurrently, resources must be allocated to each of them
  - Many **types of resources** - Some (such as CPU cycles, main memory, and file storage) may have special allocation code, others (such as I/O devices) may have general request and release code
- **Accounting -** To keep track of which users use how much and what kinds of computer resources

Resource Sharing Services(cont.)

- **Protection and security -** The owners of information stored in a multiuser or networked computer system may want to control use of that information, **concurrent processes should not interfere** with each other
  - **Protection** involves ensuring that all **access to system resources is controlled**
  - **Security** of the system **from outsiders** requires user **authentication**, extends to defending external I/O devices from **invalid access attempts**

OPERATING SYSTEM INTERFACES

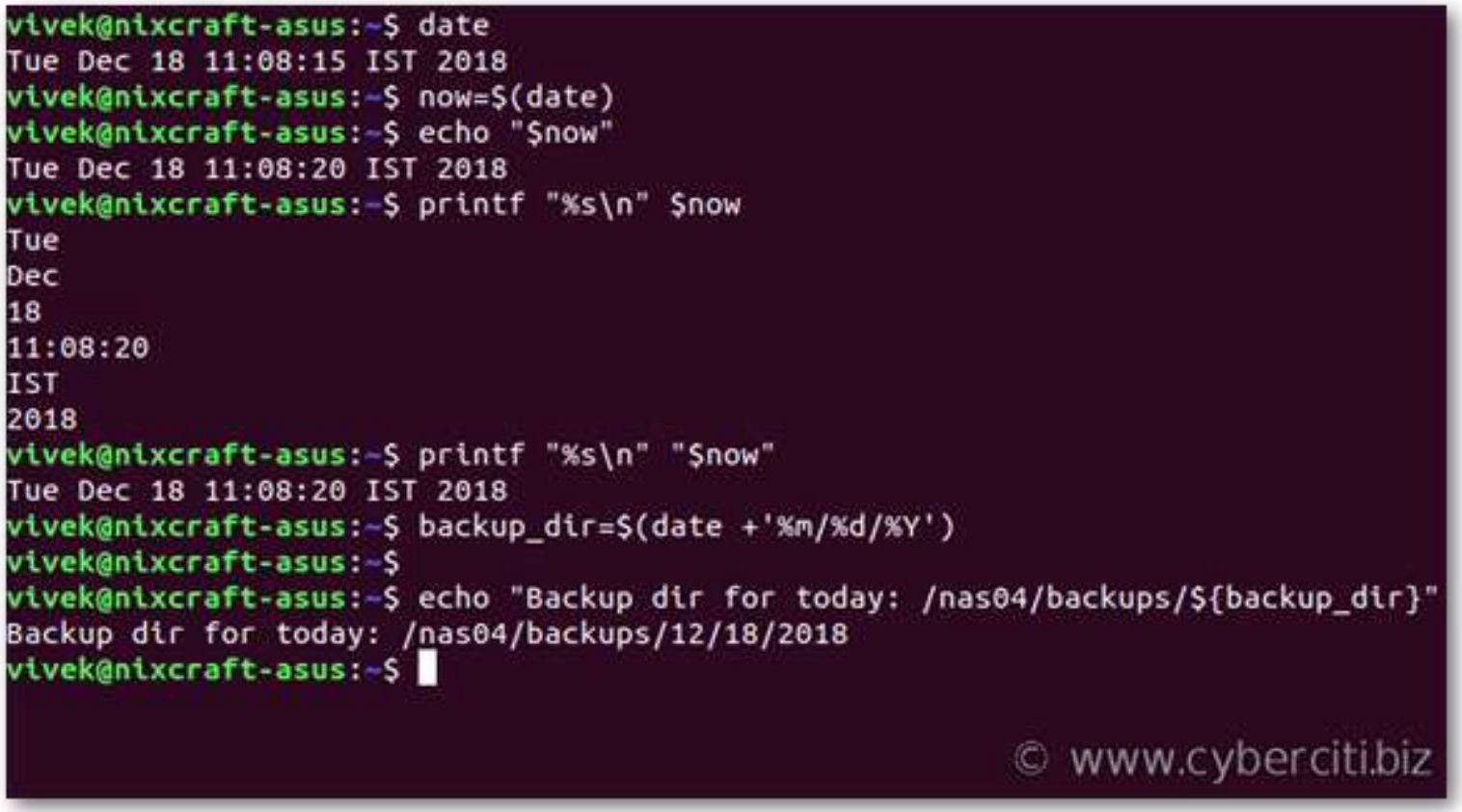
Operating System User Interface - CLI

- Command Line Interface (CLI) or **command interpreter** allows **direct command entry**
    - Sometimes implemented in **kernel**, sometimes by **systems program**
- On systems with multiple command interpreters to choose from, the
- Interpreters are known as **shells**
- **For example, on UNIX and Linux systems, a** user may choose among several different shells, including the ***Bourne shell, C shell, Bourne-Again shell, Korn shell, and others.***
- The **main function** of the command interpreter is to **get and execute the next user-specified command**.
- Many of the commands given at this level **manipulate files**: create, delete, list, print, copy, execute, and so on.

CLI (cont.)

- These commands can be **implemented in two general ways**.
- **Two approaches** of implementation
  - In one approach, the **command interpreter itself contains the code** to execute the command.
  - For example, a command to **delete a file** may cause the command interpreter to **jump to a section of its code** that sets up the parameters and makes the appropriate system call.
  - An alternative approach—used **by UNIX**, among other operating systems — **implements most commands through system programs**.
  - In this case, the command interpreter does not understand the command in any way; it merely uses the command to **identify a file to be loaded** into memory and executed.
  - Thus, the UNIX command to delete a file **rm file.txt** would search for **a file called rm**, load the file into memory, and execute it with the **parameter file.txt**

Bash Shell Command Interpreter



User Operating System Interface - GUI

- User-friendly **desktop** metaphor interface
  - Usually **mouse, keyboard, and monitor**
  - **Icons** represent files, programs, actions, etc
  - Various mouse buttons over objects in the interface cause various actions (provide information, options, execute function, open directory (known as a **folder**)
  - Invented at **Xerox PARC**
- Many systems now include **both CLI and GUI** interfaces
  - Microsoft Windows is GUI with CLI “command” shell
  - Apple Mac OS X as “Aqua” GUI interface with UNIX kernel underneath and shells available
  - Solaris is CLI with optional GUI interfaces (Java Desktop, KDE)

iPad touch screen

Choice of Interface

- The choice of whether to use a command-line or GUI interface is mostly one of personal preference.
- **System administrators** and **power users** who have deep knowledge of a system frequently use the **command-line interface**.
- For them, it is more efficient, giving them faster access to the activities they need to perform.
- Further, command line interfaces usually make **repetitive tasks easier - scripts**
- For example, if a frequent task requires a set of command-line steps, those steps can be recorded into a file, and that file can be run just like a program.
- The is interpreted by the command-line interface.
- These **shell scripts are** very common on systems that are command-line oriented, such as UNIX and Linux

Choice of Interface(Cont.)

- In contrast, most Windows users are happy to use the Windows **GUI environment**.
- The user interface can vary from system to system and even from user to user within a system.
- It typically is substantially removed from the actual system structure.
- The design of a useful and **friendly user** interface is therefore not a direct function of the operating system