CS341: Operating System

Operating System Service & Structure

Lect05 : 12th Aug 2014

Dr. A. Sahu

Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

Outline

- Operating system service
  - System Call
- Types of computing Environment
- Possibility of exploring open-source OS
- Structure & Components of OS
  - One should know breadth knowledge about OS before going to each topic in depth.

Operating System Services

- OS provide
  - An environment for execution of programs
  - And services to programs and users
- Services
  - One set of OS services provides functions that are ***helpful to the user***
  - Another set of OS functions exists for ***ensuring the efficient operation of the system*** itself via resource sharing

OS Services: provides functions that are helpful to the user

- **User interface** - Almost all OSs 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
  - 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

OS Services: provides functions that are helpful to the user

- **File-system manipulation** - The file system is of particular interest.
  - Programs need to read and write files and directories, create and delete them, search them, list file Information, permission management.
- **Communications** – 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)

OS Services: provides functions that are helpful to the user

- **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

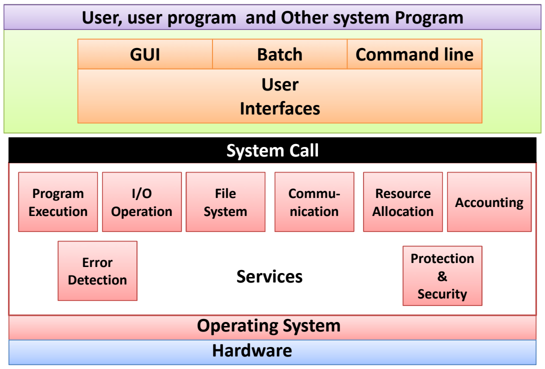
OS Services: provides functions that ensure efficient operation of System

- **Resource sharing and allocation –**
  - When multiple users or multiple jobs running concurrently, resources must be allocated to each of them
  - Many types of resources - CPU cycles, main memory, file storage, I/O devices.
- **Accounting -** To keep track of
  - Which users use how much and what kinds of computer resources

OS Services: provides functions that ensure efficient operation of System

- **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

A View of OS Services



User, user program and Other system Program

GUI

Batch

Command line

User Interfaces

System Call

Program Execution

I/O Operation

File System

Commu-nication

Resource Allocation

Accounting

Error Detection

Services

Protection & Security

Operating System

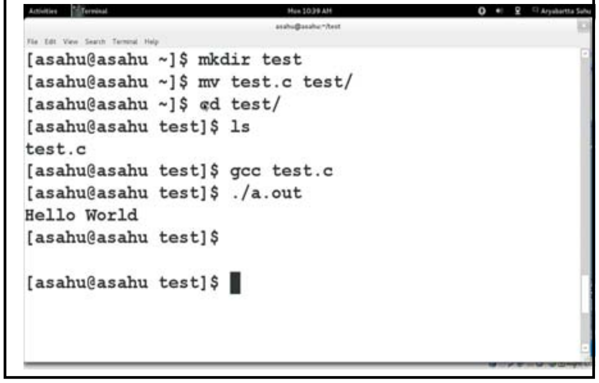
Hardware

OS User Interface - CLI

CLI or command interpreter allows direct command entry

- Sometimes implemented in kernel, sometimes by systems program
- Sometimes multiple flavors implemented – shells
- Primarily fetches a command from user and executes it
- Sometimes commands built-in, sometimes just names of programs
  - If the latter, adding new features doesn't require shell modification

Linux Shell: Command Interpreter



```
[asahu@asahu ~]$ mkdir test
[asahu@asahu ~]$ mv test.c test/
[asahu@asahu ~]$ cd test/
[asahu@asahu test]$ ls
test.c
[asahu@asahu test]$ gcc test.c
[asahu@asahu test]$ ./a.out
Hello World
[asahu@asahu test]$

[asahu@asahu test]$ █
```

OS User Interface - GUI

- Graphical User Interfaces
- 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**

OS User Interface - GUI

- Many systems now include both CLI and GUI interfaces
  - Microsoft **Windows** is GUI with CLI "command" shell
  - Apple Mac OS X is "**Aqua**" GUI interface with UNIX kernel underneath and shells available
  - Unix and Linux have CLI with optional GUI interfaces (CDE, **KDE, GNOME**)

Mac Book GUI

14

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.
- **Android Kit Kat**

System Calls

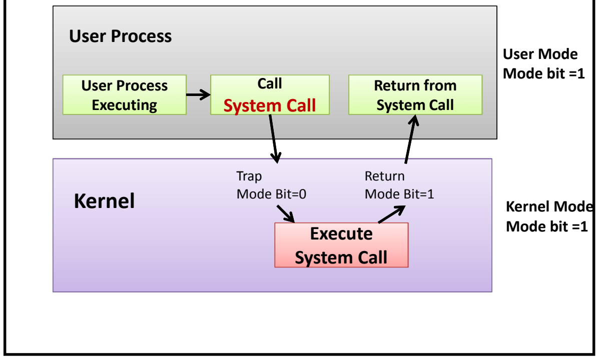
- Programming interface to the services provided by the OS
  - Typically written in a high-level language (C or C++)
- Mostly accessed by programs
  - Via a high-level **Application Programming Interface (API)**
  - rather than direct system call use
- Three most common APIs are
  - **Win32 API** for 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)

Note that the system-call names used throughout this course are generic

Operating-System Operations Modes

- **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
- Increasingly CPUs support multi-mode operations
  - i.e. **virtual machine manager** (**VMM**) mode for guest **VMs**

Transition from User to Kernel Mode

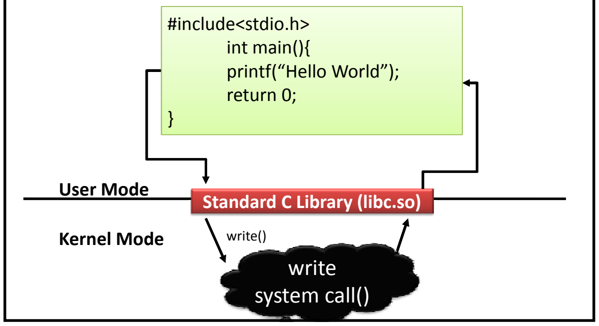


Installing Linux on 8085 or 8086

- Is it possible to install linux on top of 8085 or 8086 based system ?
- **No**
- Because it don't support mode bit
  - Kernel Mode or user mode bit
  - i386,i586,i686……in short ix86 support mode bit

Standard C Library Example

C program invoking printf() library call, which calls write() system call



```
#include<stdio.h>
        int main(){
        printf("Hello World");
        return 0;
}
```