

So far we have briefly covered

1. OS definition
2. OS evolution
3. The computer architecture managed by OS

OS Services

OS services?

Provisions of OS supports for :

- Convenience of the User and
- Efficiency of computer system operation

OS Services

Services for the user's convenience:

- Program Execution
- I/O Operation
- File Manipulation
- Communication
- Error Detection

OS Services: Program Execution

Program Execution

OS provides for Loading, Running, and Ending of User programs.

OS Services: I/O Operation

I/O operation

OS provides for actual execution of I/O operations.

OS Services: File Manipulation

File Manipulation

OS provides for:

reading from, writing into,
creating, and deleting files.

OS Services: Communication

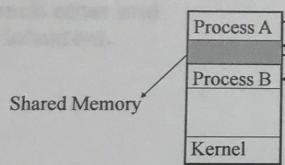
Communication

OS provides for exchange of information
between two processes which may take place in
one of the following two methods:

- a. Shared memory
- b. Message passing

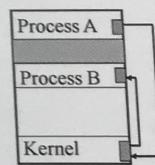
OS Services: Communication

(a) Shared Memory



OS Services: Communication

(b) Message Passing



OS Services: Error Detection

Error Detection

OS provides for
detecting and handling of both hardware
and software errors

OS Services

Services for the efficiency:

- Resource Allocation
- Accounting
- Protection

Changing the contents of any of the above
registers is a privileged instruction.

OS Services: Resource Allocation

Resource Allocation

- Special allocation category

Allocation of resources that need customized allocation routines (e.g., allocation of CPU, allocation of Memory, etc.)

- General allocation category

Allocation of resources that need general allocation routines (e.g., I/O devices)

OS Services: Accounting

Accounting

OS provides for collecting information that are used for:

Billing and Reconfiguration

OS Services: Protection

OS provides for protecting:
processes from each other and
The system from intruders.

Protections

protection

I/O Protection

Memory Protection

CPU Protection

Protections

I/O Protection

Dual mode

Monitor mode

User Mode

mode-bit

Protection

Memory Protection

Fence Register

Lower and upper bound registers

Base and Limit Registers

Changing the contents of any of the above registers is a privileged instructions.

System Calls: process Control

OS Services

How to invoke OS Services:

- *System Calls*
- *System Programs*

System Calls: process Control

OS Services

What is a system call?

It is a small program by which a process asks for a service from the OS

∴ It is an interface between a process and OS

System Calls

Example:

The system calls in completing the process of reading from a file.

get name of the file
open file
read data
close file

System Calls

The number of system calls are different for different operating systems and they are in hundreds:

<i>Windows 7</i>	<i>Over 600</i>
<i>FreeBSD</i>	<i>Over 500</i>
<i>NetBSD</i>	<i>Over 400</i>
<i>Linux</i>	<i>Over 300</i>

System Calls

Most of the system calls are hidden from the programmer by the compiler and by the runtime support routines.

Example

A Write statement is compiled to a runtime support routine that in turn is responsible for invoking right system calls to execute a write statement.

System Calls

System Calls Categories:

Process Control
File management
Device management
Information maintenance
Communications

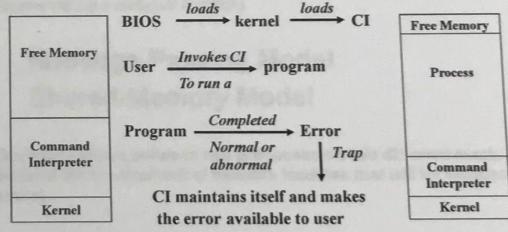
System Calls: process Control

Process Control

create process,
load, execute
halt execution, (end, abort)
terminate process
get process attributes, set process attributes
wait for time, wait event, signal event
allocate and free memory

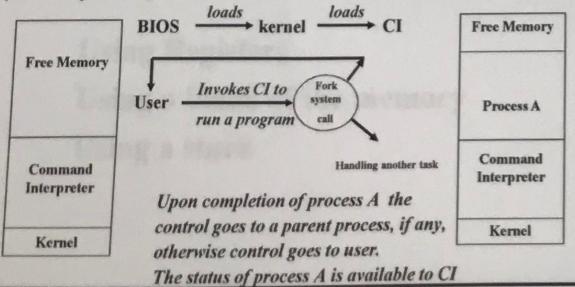
System Calls: process Control

Process controller behavior in a single-tasking system (MS-DOS)



System Calls: process Control

Process controller behavior in a multi-tasking system (Berkeley Unix)



System Calls: File Manipulation

File Manipulation

create file, delete file
open, close
read, write, reposition
get file attributes, set file attributes

System Calls: Device Management

Device Management

request device, release device
read, write, reposition
get device attributes, set device attributes
logically attach or detach devices.

System Calls: Information maintenance

Information maintenance

get time or date
set time or date
get system data, set system data
get process, file, or device attributes
set process, file, or device attributes

System Calls: Communications

Communications

create, delete communication connections
send, receive messages
transfer status information
attach and detach remote devices

System Calls: Communications

Communication between two Processes

(Processes may be on the same machine or different machines connected by a network facility.)

Message-Passing Model
Shared-Memory Model

Communications between two processes on two different machines demand the involvement of network facilities that will be discussed in future)

A reminder

Parameter passing into the OS

Using Registers

Using a block of the memory

Using a stack

OS Services

Accessibility:

System calls for UNIX are callable by C or C++ programs.

System calls for Modern Microsoft Windows platform are part of the Win32 API (Application Programmer Interface) and they are callable by all the compilers written for Microsoft windows.

System Programs

What is a System Program?

It is a large program by which a user asks for a service from the OS

∴ It is an interface between a user and OS

Example:

A Word Processor

System Programs

What are the differences between a “System call” and a “System Program”?

1. System call is a much smaller program than system program
2. Several system calls are involved in one system program
3. System call provides an interface between a process and OS whereas a system program provides an interface between user and OS