RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

BCS-501 Operating System

Lecturer-04

Manisha Verma
Assistant Professor
Computer Science & Engineering

System Calls

System call is a pogramming interface to the services provided by the OS or it will take permisson from OS after that it will perfom action.

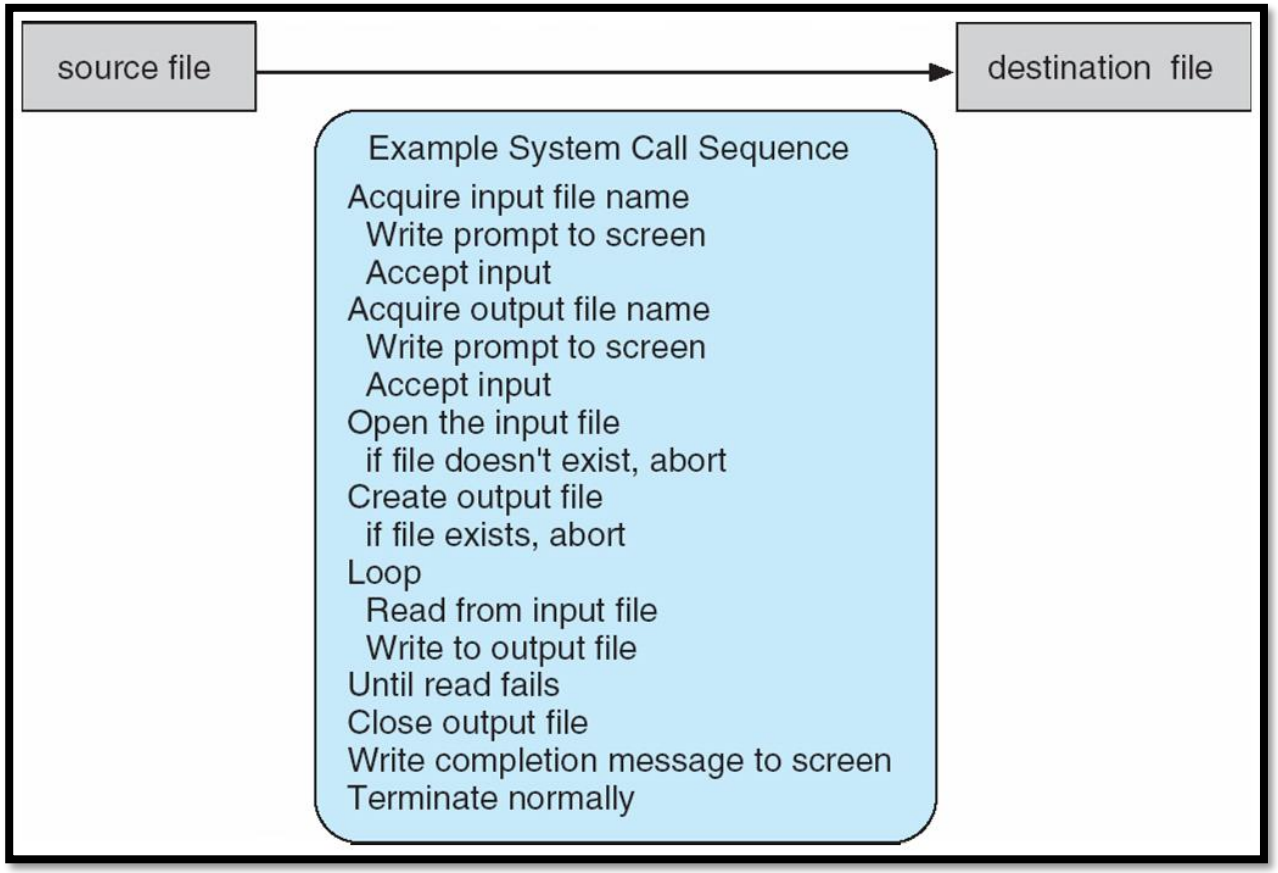
Typically written in a high-level language (C or C++)

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), and Java API for the Java virtual machine .

System Calls

System call sequence to copy the contents of one file to another file



EXAMPLE OF STANDARD API

As an example of a standard API, consider the `read()` function that is available in UNIX and Linux systems. The API for this function is obtained from the `man` page by invoking the command

```
man read
```

on the command line. A description of this API appears below:

```
#include <unistd.h>

ssize_t     read(int fd, void *buf, size_t count)
```

return value — function name — parameters

A program that uses the `read()` function must include the `unistd.h` header file, as this file defines the `ssize_t` and `size_t` data types (among other things). The parameters passed to `read()` are as follows:

- `int fd`—the file descriptor to be read
- `void *buf`—a buffer where the data will be read into
- `size_t count`—the maximum number of bytes to be read into the buffer

On a successful read, the number of bytes read is returned. A return value of 0 indicates end of file. If an error occurs, `read()` returns −1.

System Calls

Typically, a number associated with each system call

System-call interface maintains a table indexed according to these numbers

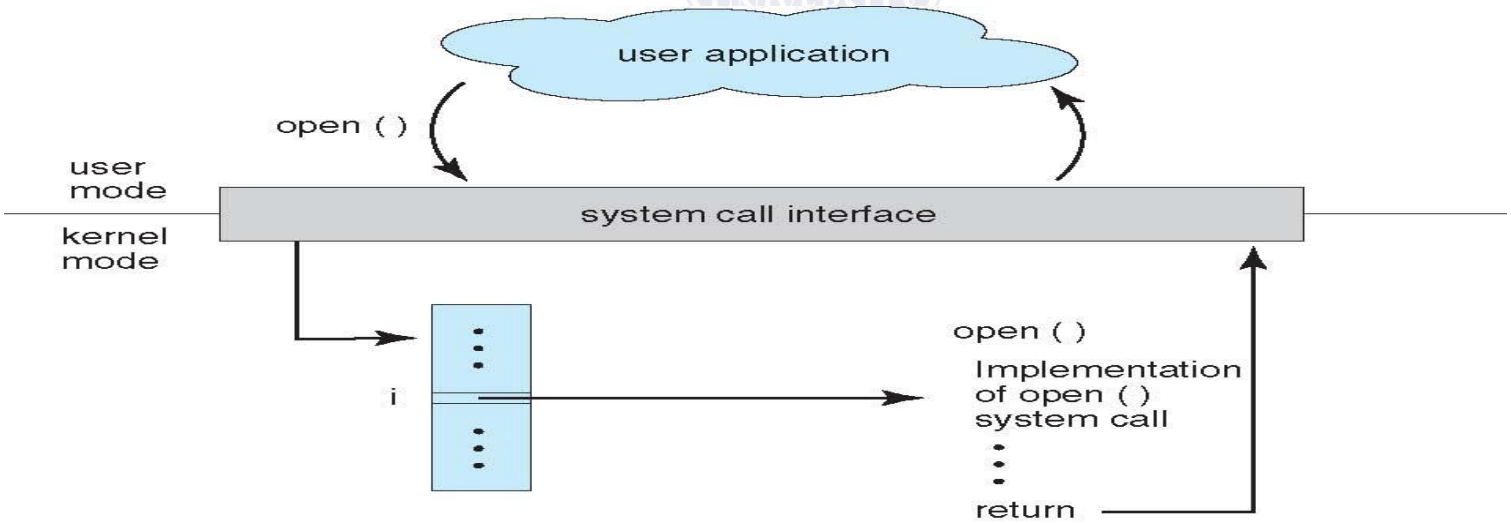
The system call interface invokes the intended system call in OS kernel and returns status of the system call and any return values.

The caller need know nothing about how the system call is implemented

Just needs to obey API and understand what OS will do as a result call

Most details of OS interface hidden from programmer by API

Managed by run-time support library (set of functions built into libraries included with compiler)



System call parameter

Often, more information is required than simply identity of desired system call

Exact type and amount of information vary according to OS and call

Three general methods used to pass parameters to the OS

Simplest: pass the parameters in registers

In some cases, may be more parameters than registers

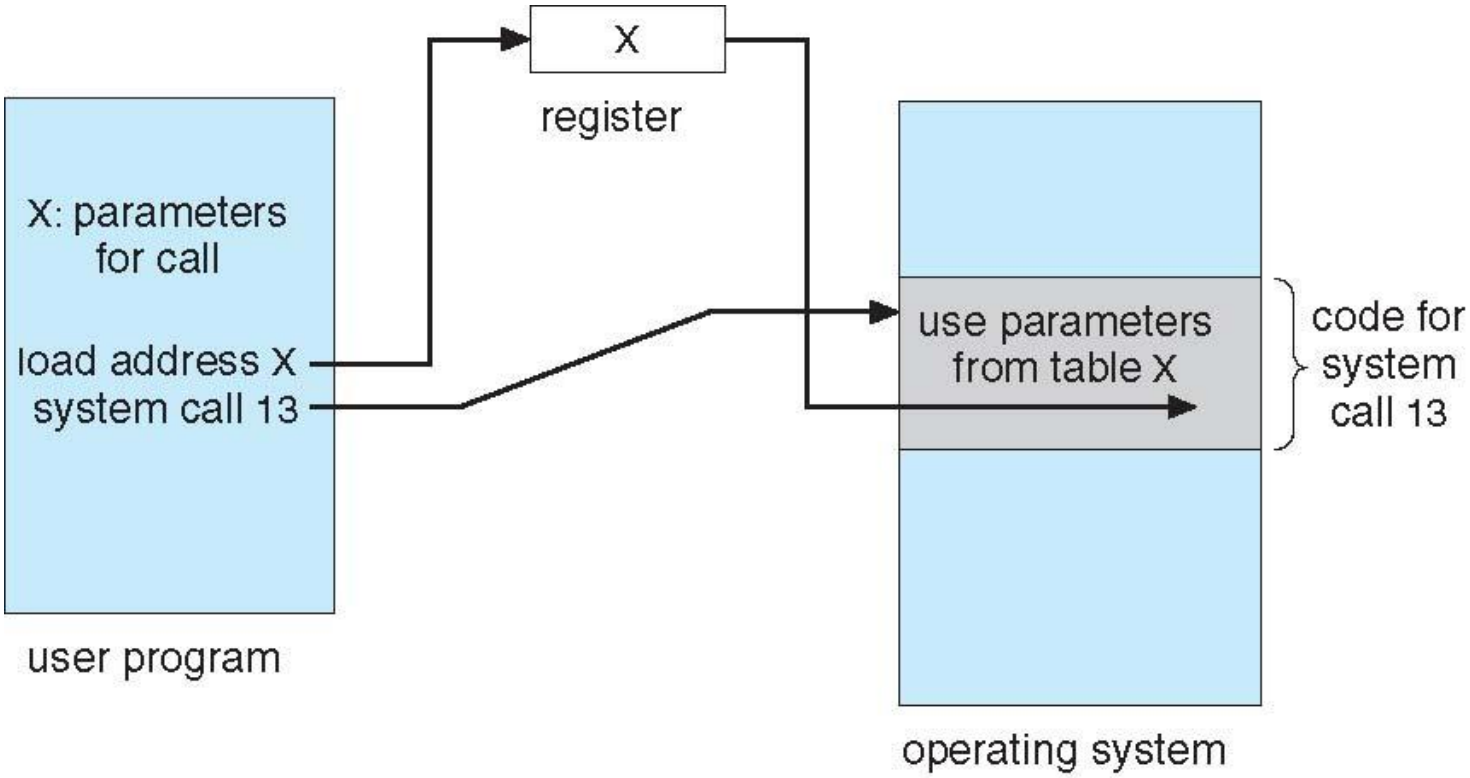
Parameters stored in a block, or table, in memory, and address of block passed as a parameter in a register

This approach taken by Linux and Solaris

Parameters placed, or pushed, onto the stack by the program and popped off the stack by the operating system

Block and stack methods do not limit the number or length of parameters being passed.

parameter passing value



Types

Information maintenance
- get time or date, set time or date
- get system data, set system data
- get and set process, file, or device attributes

Communications
- create, delete communication connection
- send, receive messages if message passing model to host name or process name from client to server
- Shared-memory model create and gain access to memory regions
- transfer status information
- attach and detach remote devices

File management
- create file, delete file
- open, close file
- read, write, reposition
- get and set file attributes

Device management
- request device, release device
- read, write, reposition
- get device attributes, set device attributes
- logically attach or detach devices

Information maintenance--

- get time or date, set time or date
- get system data, set system data
- get and set process, file, or device attributes
- Communications
- create, delete communication connection
- send, receive messages if message passing model to host name or process name

• client to server---

- Shared-memory model create and gain access to memory regions
- transfer status information
- attach and detach remote devices

Protection-----

- Control access to resources
- Get and set permissions
- Allow and deny user access

MCQ

The system call interface invokes the intended system call in OS kernel and returns status of the system call and……..
1.any return values
2. any backvalues
3. Any null values
4. any novalues

communication connection is a…….
1.Send messages
2. receive messages
3. if message passing model to host name or process name
4. All of these

System-call interface maintains a…….
1.table indexed
2.numbers
3.OS kernel and returns status of the system call
4.All of these

Shared-memory model create and gain access to ….
1.memory regions
2. transfer status information
3.attach and detach remote devices
4.None of these

Programming interface to the services provided by the…….
1.OS
2.Memory
3.Virtual memory
4.cpu