Programming in Linux

Self learning with Linux

Team Emertxe



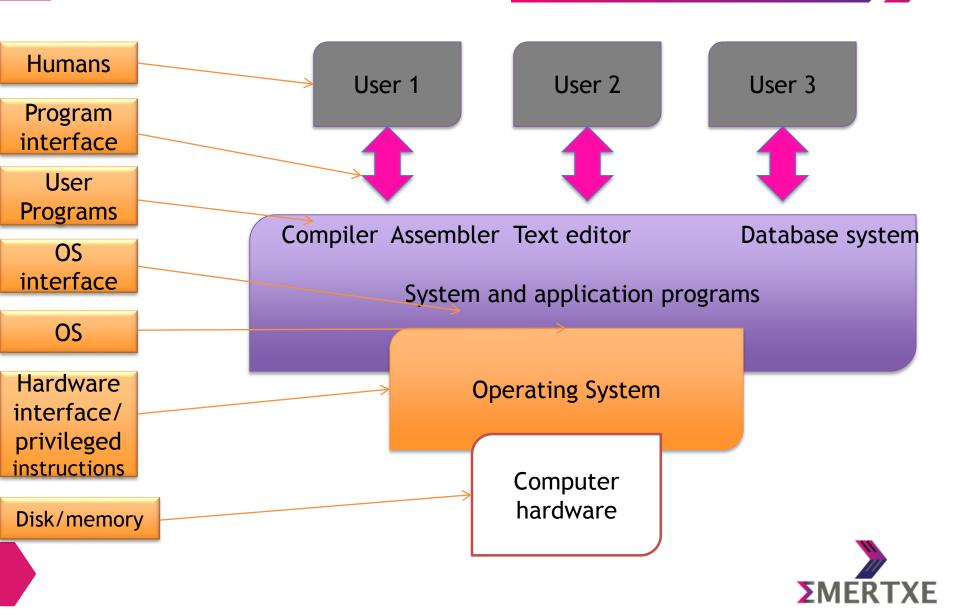
Linux Introduction

Let us ponder...

- ✓ What exactly is an Operating System (OS)?
- ✓ Why do we need OS?
- ✓ How would the OS would look like?
- ✓ Is it possible for a team of us (in the room) to create an OS of our own?
- ✓ Is it necessary to have an OS running in a Embedded System?
- ✓ Will the OS ever stop at all?



Operating System



What is Linux?

- ✓ Linux is a free and open source operating system that is causing a revolution in the computer world.
- ✓ Originally created by Linus Torvalds with the assistance of developers called community
- ✓ This operating system in only a few short years is beginning to dominate markets worldwide.



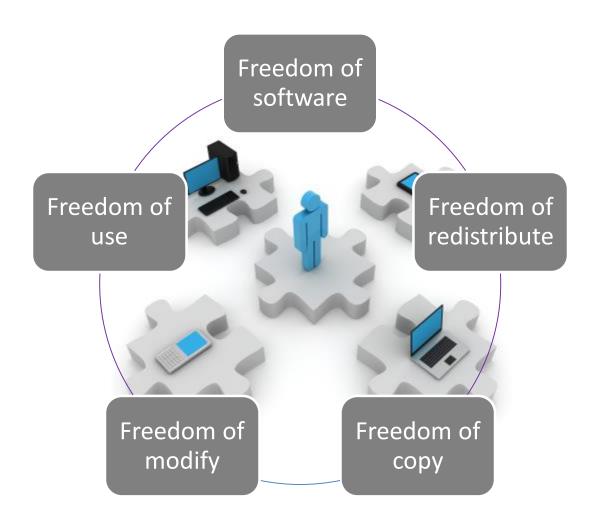


Why use Linux?

- ✓ Free & Open Source
- ✓ Reliability
- ✓ Secure
- ✓ Scalability



What is Open Source?





How it all started?

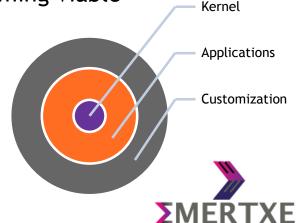
- ✓ With GNU (GNU is not UNIX)
- ✓ Richard Stallman made the initial announcement in 1983, Free Software Foundation (FSF) got formed during 1984
- ✓ Volunteer driven GNU started developing multiple projects, but making it as an operating system was always a challenge
- ✓ During 1991 a Finnish Engineer Linus Torvalds developed core OS functionality, called it as "Linux Kernel"
- ✓ Linux Kernel got licensed under GPL, which laid strong platform for the success of Open Source
- ✓ Rest is history!





How it evolved?

- ✓ Multiple Linux distributions started emerging around the Kernel
- ✓ Some applications became platform independent
- ✓ Community driven software development started picking up
- ✓ Initially seen as a "geek-phenomenon", eventually turned out to be an engineering marvel
- ✓ Centered around Internet
- ✓ Building a business around open source started becoming viable
- ✓ Redhat set the initial trend in the OS business



Where it stands now?

OS



CIOSCUD

Novell.

Databases





VoltDB

Server/Cloud









Enterprise









Consumer









Education







CMS



AUT@MATTIC



eCommerce









More details

Open Source SW vs. Freeware

OSS	Freeware
 ✓ Users have the right to access & modify the source codes ✓ In case original programmer disappeared, users & developer group of the S/W usually keep its support to the S/W. ✓ OSS usually has the strong users & developers group that manage and maintain the project 	 ✓ Freeware is usually distributed in a form of binary at 'Free of Charge', but does not open source codes itself. ✓ Developer of freeware could abandon development at any time and then final version will be the last version of the freeware. No enhancements will be made by others. ✓ Possibility of changing its licensing policy



GPL

- ✓ Basic rights under the GPL access to source code, right to make derivative works
- ✓ Reciprocity/Copy-left
- ✓ Purpose is to increase amount of publicly available software and ensure compatibility
- ✓ Licensees have right to modify, use or distribute software, and to access the source code



Problems with the GPL

- ✓ Linking to GPL programs
- √ No explicit patent grant
- ✓ Does no discuss trademark rights
- ✓ Does not discuss duration
- ✓ Silent on sub-licensing
- ✓ Relies exclusively on license law, not contract

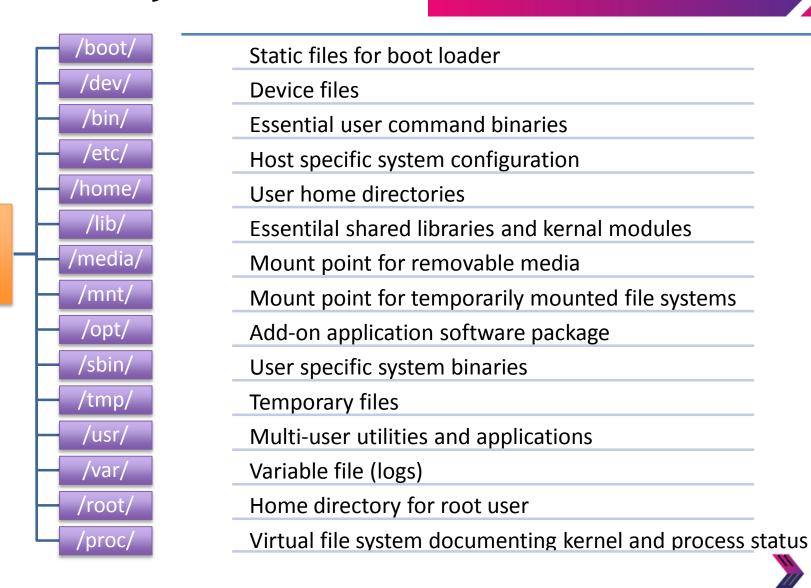


Linux: Properties

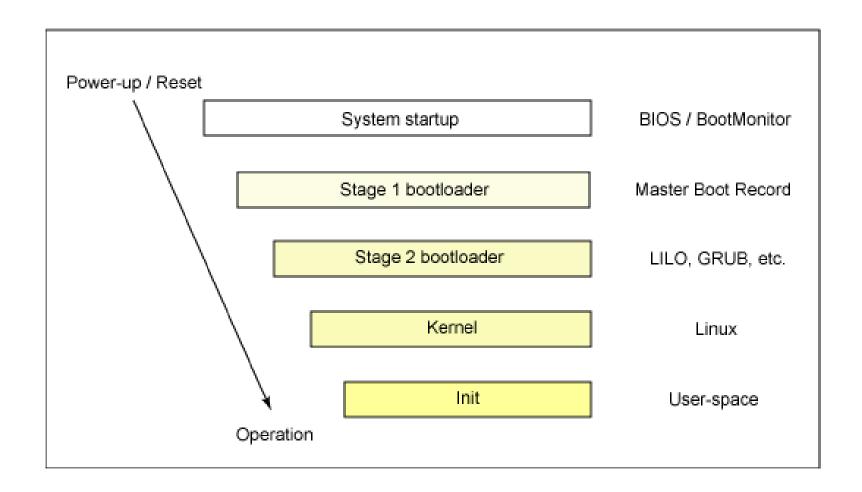
- ✓ Multitasking
- ✓ Multi-user
- ✓ Multiprocessing
- ✓ Protected Memory
- ✓ Hierarchical File System



Directory structure



x86 Booting



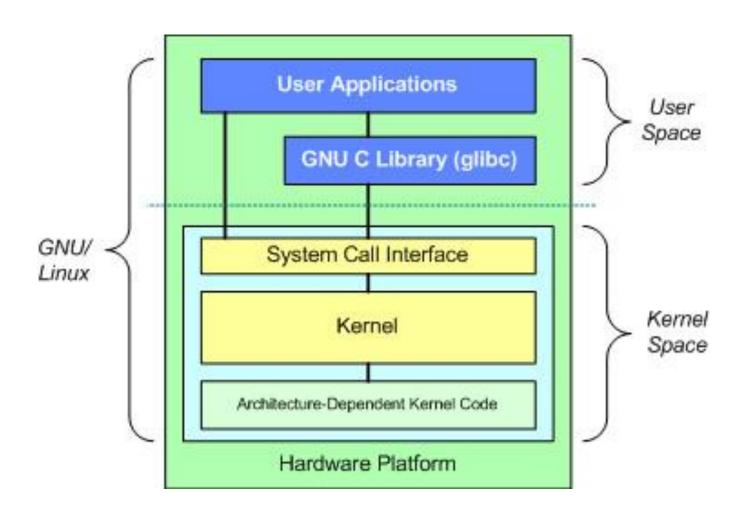


Kernel structure

- ✓ Hardware Controllers: This subsystem is comprised of all the possible physical devices in a Linux installation
- ✓ Linux Kernel: The kernel abstracts and mediates access to the hardware resources, including the CPU. A kernel is the core of the operating system
- ✓ O/S Services: These are services that are typically considered part of the operating system (e.g. shell)
- ✓ User Applications: The set of applications in use on a particular Linux system. (e.g. web-browser)



Kernel structure





Command Line Interface

Command Line Interface

- CLI
 - Textual mode
 - Executes requested command
- GUI
 - Mouse, keypad

```
File Edit View Scrollback Bookmarks Settings Help
        esd/common/.svn/text-base/lcd.c.svn-base:void lcd_setup(int lcd, int config)
        'esd/common/lcd.c:void lcd_setup(int lcd, int config)
                                                   /* lcd setup & logo display */
 inary file ./lib_arm/board.o matches inary file ./lib_arm/libarm.a matches
                                  :.svn-base:int lcd_setup(void);
:.svn-base: lcd_setup,
                                                                              /* lcd setup & logo display */
       rs/lcd_logo.c:int lcd_setup(void)
 inary file ./drivers/libdrivers.a matches
                             d_logo.c.svn-base:int lcd_setup(void)
 inary file ./u-boot matches
                                             lib_arm/libarm.a(board.o) (lcd_setup)
lcd_setup
 srinivas_reddy@srinivas-dt u-boot-1.1.6]$ vi lib
            lib_blackfin/ lib_i386/ lib_microblaze/ lib_nios/
lib_generic/ lib_m68k/ lib_mips/ lib_nios2.
 srinivas_reddy@srinivas-dt u-boot-1.1.6]$ vi lib_arm/board.c
 srinivas_reddy@srinivas-dt u-boot-1.1.6]$ ksnapshot
     Booklet: bash Booklet: vi u-boot-1.1.6: bash seenu: bash ibash
🔛 🔚 💹 🎉 🛃 🤡 👫 💹 weboot-1.16 : 😇 ODTs - Dolph: 🕙 Inbox - Thund: 🔞 File Ubuntu-8 : 📻 emertxe_slide 🗸 🖟 📜 📆 🔻 10:44 🖟
```





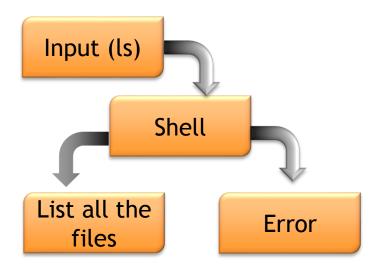
The Shell

- ✓ What is a shell?
- ✓ Different types of shells
 - Login-shell
 - Non-login shell
 - Sh
 - Bash
 - Ksh
 - Csh
- ✓ Hands-on:
 - echo \$0
 - cat /etc/shells



How Shell Invokes

The main task of a shell is providing a user environment





Bash Files

- ✓ Bash
 - Command interpreter
 - .bash_profile (During login)
 - .bashrc (New instance)
 - .bash_logout (Logout)
 - .bash_history (Command history)
- ✓ Hands-on:
 - Enter ls -a in your home directory
 - Display contents of all files mentioned above



Environment Variables

- ✓ Login-shell's responsibility is to set the non-login shell and it will set the environment variables
- ✓ Environment variables are set for every shell and generally at login time
- ✓ Environmental variables are set by the system.
- ✓ Environmental variables hold special values. For instance ,\$ echo \$SHELL
- ✓ Environmental variables are defined in /etc/profile, /etc/profile.d/ and ~/.bash_profile.
- ✓ When a login shell exits, bash reads ~/.bash_logout



The 'bash' variables & Friends

Name	Meaning	
env	Lists shell environment variable/value pairs	
<pre>export [var_name]</pre>	Exports/sets a shell variable	
HOME	Path of user's home directory	
PATH	Executable search path	
PWD	Present working directory	
PS1	Command prompt	
N=10	Assigning the variable. This a temporary variable effective only inside the current shell	
unset N	Unset the environment variable N	



Basic Shell Commands

Command	Meaning
ls	Lists all the files
pwd	Gives present working directory
cd	Change directory
man	Gives information about command
exit	Exits from the shell
which	Shows full path of command



Shell: Built-in Commands

- ✓ Built-in commands are contained with in the shell itself, means shell executes the command directly, without creating a new process
- ✓ Built-in commands: break,cd,exit,pwd,export,return,unset,alias,echo,print f,read,logout,help,man



VIsual editor

VIsual editor

- √ vi or vim
- ✓ To open a file
 - \$ vi <filename> or vim <filename>



VIsual editor...

- ✓ vi opens a file in command mode while starting
- ✓ The power of vi comes from its 3 modes
 - Escape mode (Command mode)
 Search mode
 File mode
 - Editing mode
 Insert mode
 Append mode
 Open mode
 Replace mode
 - Visual mode.



Cursor Movement

- ✓ You will clearly need to move the cursor around your file. You can move the cursor in command mode.
- ✓ vi has many different cursor movement commands. The four basic keys appear below

Character	Meaning
k	Move up one line
h	Move one character to the left
l	Move one character to the right
j	Move down one line

✓ Yes! Arrow keys also do work. But these makes typing faster



Basic vi commands

√ How to exit

Command	Meaning
: q	Quit without saving
:wq	Close the file with saving
:q!	Quit the file forcefully without saving

- ✓ Already looks too complicated?
- ✓ Try by yourself, let us write a C program
- ✓ Try out vimtutor. Go to shell and type vimtutor.



Escape mode or Command mode

✓ In command mode, characters you perform actions like moving the cursor, cutting or copying text, or searching for some particular text

Search mode

- vi can search the entire file for a given string of text. A string is a sequence of characters. vi searches forward with the slash (/) key and string to search.
- To cancel the search, press ESC . You can search again by typing n (forward) or N (backward).
- Also, when vi reaches the end of the text, it continues searching from the beginning. This feature is called wrap scan. Instead of (/), you may also use question (?). That would have direction reversed
- Now, try out. Start vi as usual and try a simple search. Type
 /<string> and press n and N a few times to see where the
 cursor goes.

Escape mode...

√ File mode

• Changing (Replacing) Text

Command	Meaning
:%s/first/sec	Replaces the first by second every where in the file
%s/old/new/gc	For all lines in a file, find string "old" and replace with string "new" for each instance on a line
:e filename	Open another file without closing the current
set all	Display all settings of your session
:r filename	Reads file named filename in place



Editing Modes...

Command	Mode name	Insertion point
a	Append	Just after the current character
A	Append	End of the current line
i	Insert	Just before the current character
1	Insert	Beginning of the current line
0	Open	New line below the current line
0	Open	New line above the current line



Editing Text

- ✓ Deleting Text Sometimes you will want to delete some of the text you are editing.
- ✓ To do so, first move the cursor so that it covers the first character of the group you want to delete, then type the desired command from the table below.

Command	Meaning
dd	For deleting a line
ndd	For deleting a n lines
X	To delete a single character
shift + d	Delete contents of line after cursor
dw	Delete word
ndw	Delete 'n' words



Some Useful Shortcuts

Command	Meaning
shift-g	Go to last line in file
shift-j	Joining the two lines
•	Repeat the previous command executed
ctrl+a	Increment number under the cursor
ctrl+x	Decrements numbers under the cursor



Visual Mode

√ Visual Mode

Visual mode helps to visually select some text, may be seen as a sub mode of the command mode to switch from the command mode to the visual mode type one of

- ctrl+v Go's to visual block mode.
- Only v for visual mode
- d or y Delete or Yank selected text
- I or A Insert or Append text in all lines (visual block only)



File related commands

File redirection

- ✓ Out put redirection (>)
- ✓ Redirecting to append (>>)
- \checkmark Redirecting the error (2>)

```
eg:$ls > /tmp/outputfile
```

eg: \$ls -l >> /tmp/outputfile

eg: \$ls 2> /tmp/outputfile



Piping

- ✓ A pipe is a form of redirection that is used in Linux operating systems to send the output of one program to another program for further processing.
- ✓ A pipe is designated in commands by the vertical bar character

```
eg: $ ls -al /bin | less
```



File Related commands

- ✓ Every thing is viewed as a file in Linux. Even a Directory is a file.
- ✓ Basic Shell Command Set

Command	Meaning
df	Disk free space
du	Disk usage
cp <source/> <dest></dest>	Copy file from one to another
mv <source/> <dest></dest>	Rename a file
rm <file></file>	Remove a file
stat	File related statistics (i-node)
ln	Linking between files (-s option for soft link)



Continued

Command	Meaning
mkdir <dir_name></dir_name>	Make directory
rmdir <dir_name></dir_name>	Remove a particular directory
touch	Change file timestamps
WC	Counts the number of lines in a file
cat	Display contents of the file in standard output
more	Display contents, navigate forward
head	Display first 10 lines of the file (-n to change)
tail	Display last 10 lines of the file (-n to change)
sort	Sort lines of text files



File Detailed Listing

```
aayush@aayush-laptop:~/Documents/try$ ls -l
total 4
brw------ 1 root root 7, 0 2010-09-12 02:02 block_file
crw------ 1 root root 108, 0 2010-09-12 02:02 character_file
drwxr-xr-x 2 aayush aayush 4096 2011-03-17 03:56 directory_file
lrwxrwxrwx 1 aayush aayush 12 2011-03-21 21:03 link_file -> regular_file
prw-r--r-- 1 root root 0 2011-03-17 04:51 namedpipe_file
-rw-r--r-- 1 aayush aayush 0 2011-03-17 04:36 regular_file
srwxr-xr-x 1 aayush aayush 0 2011-03-17 04:32 socket_file
```

permissions

Owner & group

File size

Created time & Date

Filename



Linux file types

1st column

- -
- d
- (
- b
- •
- S
- \bullet = or p

Meaning

- Plain text
- Directory
- Character driver
- Block driver
- Link file
- Socket file
- FIFO file



File permissions

```
√ r or 4 -r--r--
                        Read
✓ w or 2 --w--w-
                        Write
\checkmark x or 1 ---x--x Execute
     rwx rwx rwx
     421 421 421
     user group others
Changing the File Permissions
$ chmod - Change file permessions
$ chown - Change file owner
$ chmod [ ug+r, 746 ] file.txt
$ chown -R user:group [filename | dir ]
```



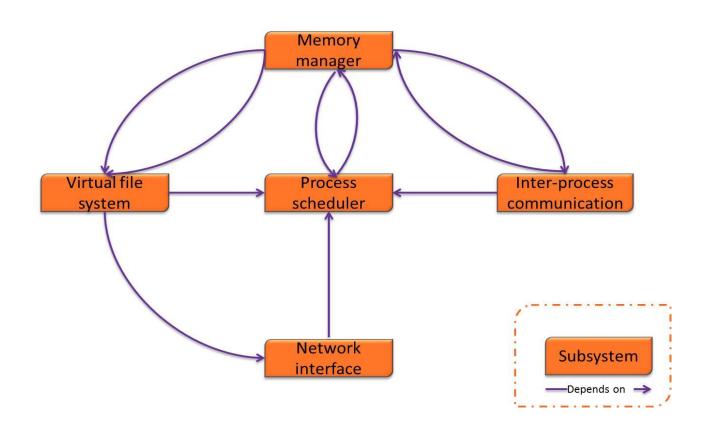
Linux kernel: Subsystems

Subsystems

- √ The Process Scheduler (SCHED)
- ✓ The Memory Manager (MM)
- ✓ The Virtual File System (VFS)
- ✓ The Network Interface (NET)
- √ The Inter-Process Communication (IPC)

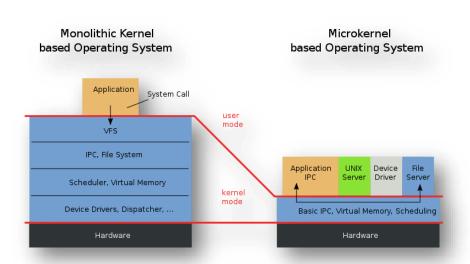


Kernel Structure





Kernel design



- Most older operating systems are monolithic, that is, the whole operating system is a single executable file that runs in 'kernel mode'
- ✓ This binary contains the process management, memory management, file system and the rest (Ex: UNIX)
- ✓ The alternative is a microkernel-based system, in which most of the OS runs as separate processes, mostly outside the kernel
- ✓ They communicate by message passing.
 The kernel's job is to handle the message passing, interrupt handling, low-level process management, and possibly the I/O (Ex: Mach)



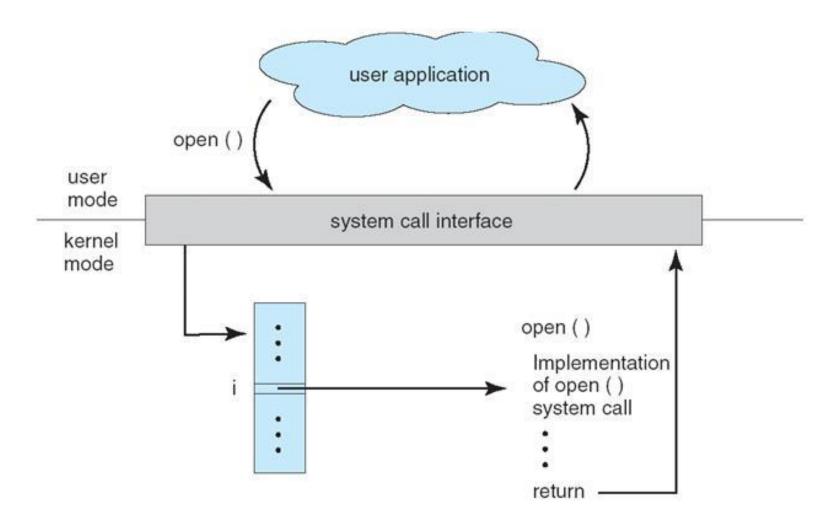
System Calls

System calls

- ✓ A set of interfaces to interact with hardware devices such as the CPU, disks, and printers.
- ✓ Advantages:
 - Freeing users from studying low-level programming
 - It greatly increases system security
 - These interfaces make programs more portable

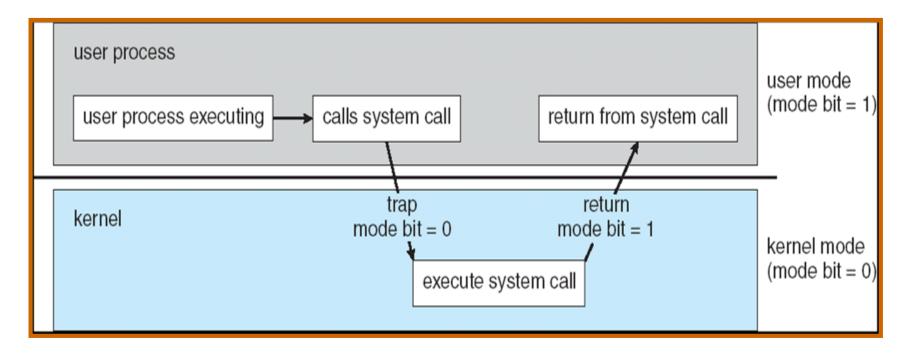


System call





System Call...



Logically the system call and regular interrupt follow the same flow of steps. The source (I/O device v/s user program) is very different for both of them. Since system call is generated by user program they are called as 'Soft interrupts' or 'traps'



Stay connected

About us: Emertxe is India's one of the top IT finishing schools & self learning kits provider. Our primary focus is on Embedded with diversification focus on Java, Oracle and Android areas

Emertxe Information Technologies,
No-1, 9th Cross, 5th Main,
Jayamahal Extension,
Bangalore, Karnataka 560046
T: +91 80 6562 9666
E: training@emertxe.com



https://www.facebook.com/Emertxe



https://twitter.com/EmertxeTweet



https://www.slideshare.net/EmertxeSlides



THANK YOU