Linux Internals

Day 1

Team Emertxe

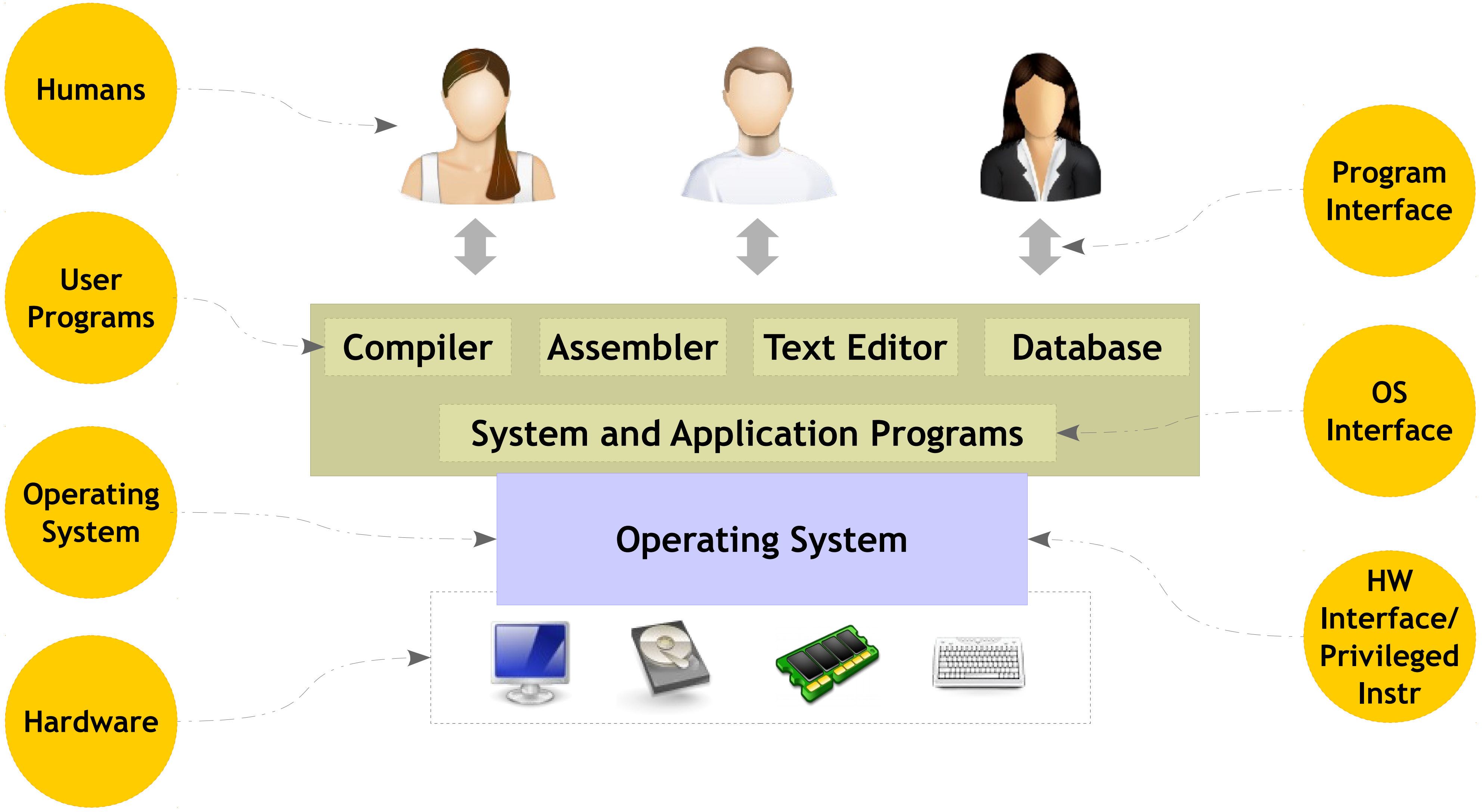
Introduction

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?

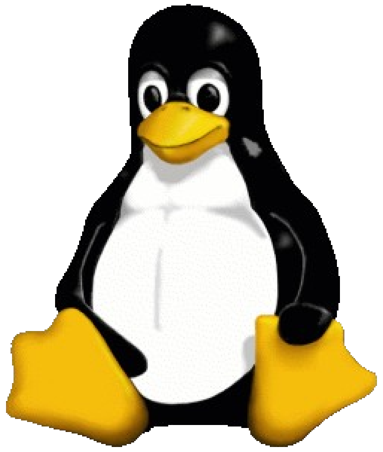
Introduction

Operating System



Introduction
What is Linux?

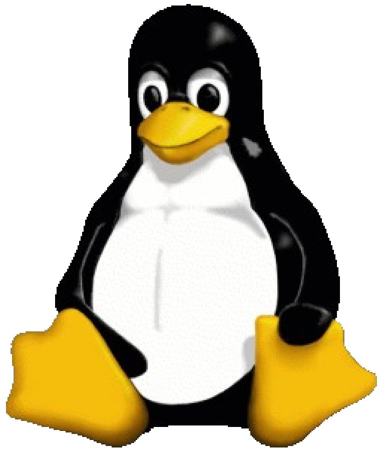
- Linux is a free and open source operating system that is causing a revolution in the computer world
- Originally created by Linus Torvalds with the assistance of developers called community
- This operating system in only a few short years is beginning to dominate markets worldwide



Introduction

Why use Linux?

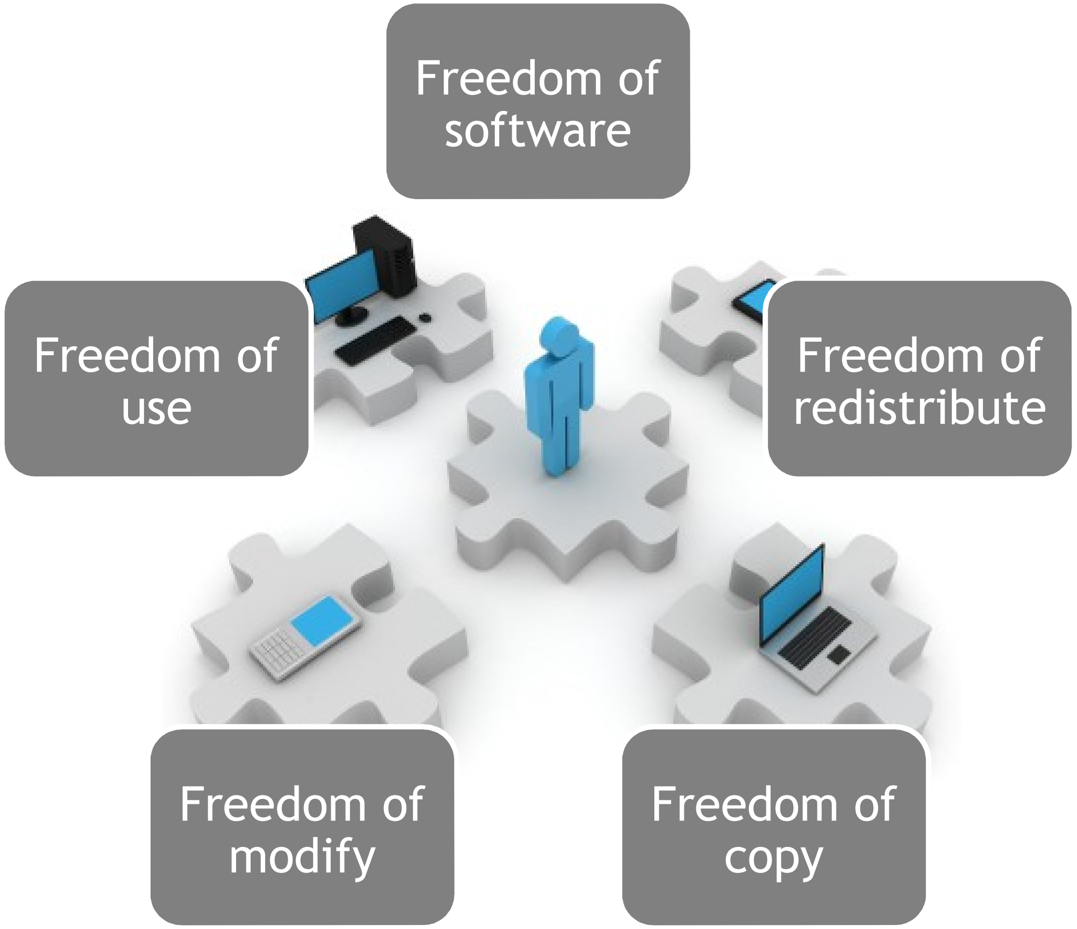
- Free & Open Source –GPL license, no cost
- Reliability –Build systems with 99.999% upstream
- Secure –Monolithic kernel offering high security
- Scalability –From mobile phone to stock market servers



EMERTXE

Introduction

What is Open Source?



Introduction

Open Source - How it all started?

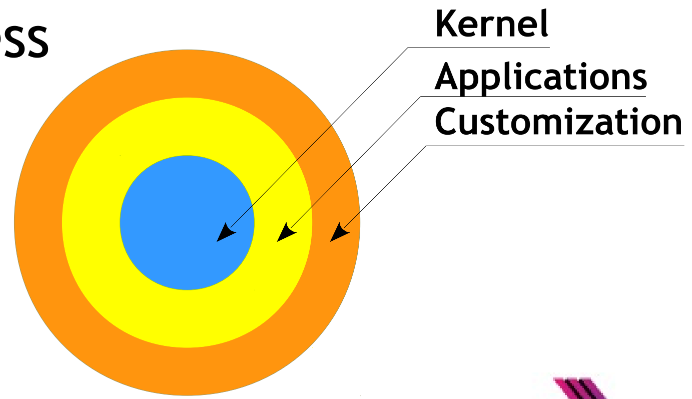
- With GNU (GNU is not UNIX)
- Richard Stallman made the initial announcement in 1983, Free Software Foundation (FSF) got formed during 1984
- Volunteer driven GNU started developing multiple projects, but making it as an operating system was always a challenge
- During 1991 a Finnish Engineer Linus 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!

ΣMERTXE

Introduction

Open Source - 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



Introduction

Open Source - Where it stands now?

OS
- CANONICAL
- redhat
- android
- Novell

Databases
- EnterpriseDB
  The Enterprise PostgreSQL Company
- MySQL
- Hortonworks
- VoltDB

Server/Cloud
- openNMS
- openstack
  CLOUD SOFTWARE
- cloudstack
  open source cloud computing
- OPSCODE

Enterprise
- pentaho
  POWERFUL ANALYTICS MADE EASY
- Alfresco
- SUGARCRM.
- XWIKI

Consumer
- Apache OpenOffice™
- LibreOffice®

Education
- canvas
  BY INSTRUCTURE
- docebo®
- moodle

CMS
- Joomla!®
- AUTOMATTIC
- MediaWiki

eCommerce
- Magento
  eCommerce Platform for Growth
- spree
- opencart

Introduction
Open Source 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

Introduction

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

Introduction
GPL - Issues

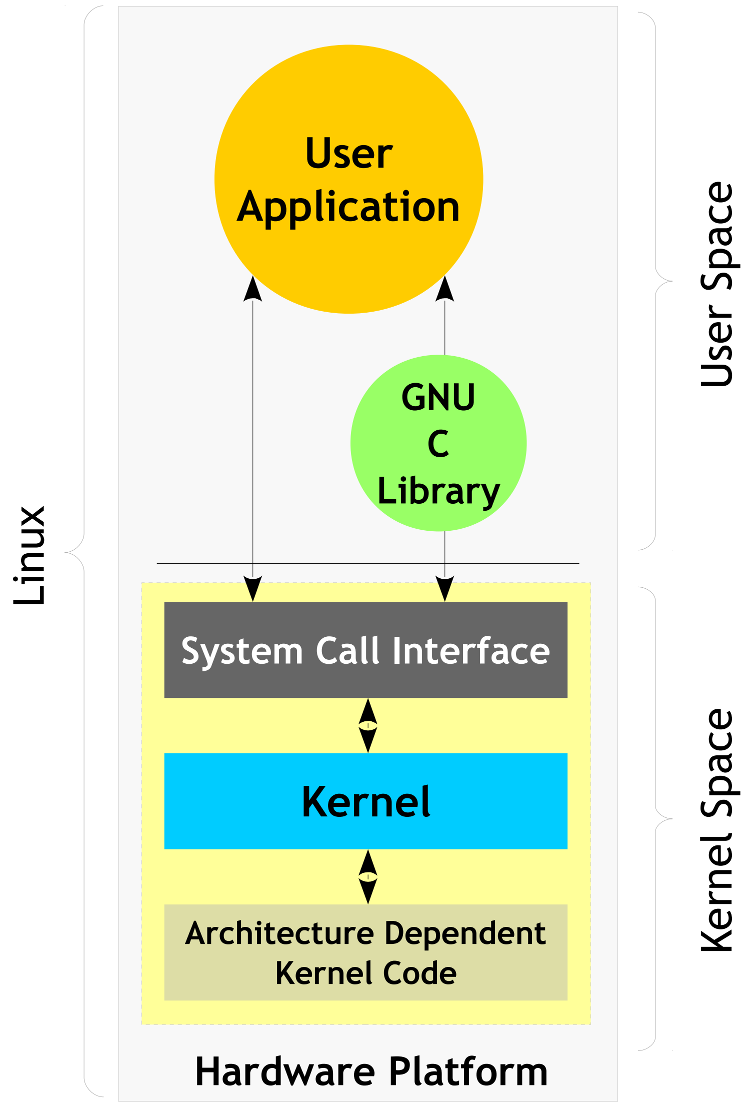
- Linking to GPL programs
- No explicit patent grant
- Does no discuss trademark rights
- Does not discuss duration
- Silent on sub-licensing
- Relies exclusively on license law, not contract

Introduction
Linux Properties

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

Introduction

Linux Components



- **Hardware Controllers:** This subsystem is comprised of all the possible physical devices in a Linux installation - CPU, memory hardware, hard disks
- **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. windowing system, command shell)
- **User Applications:** The set of applications in use on a particular Linux system (e.g. web browser)

Introduction

Linux Directory Structure

/

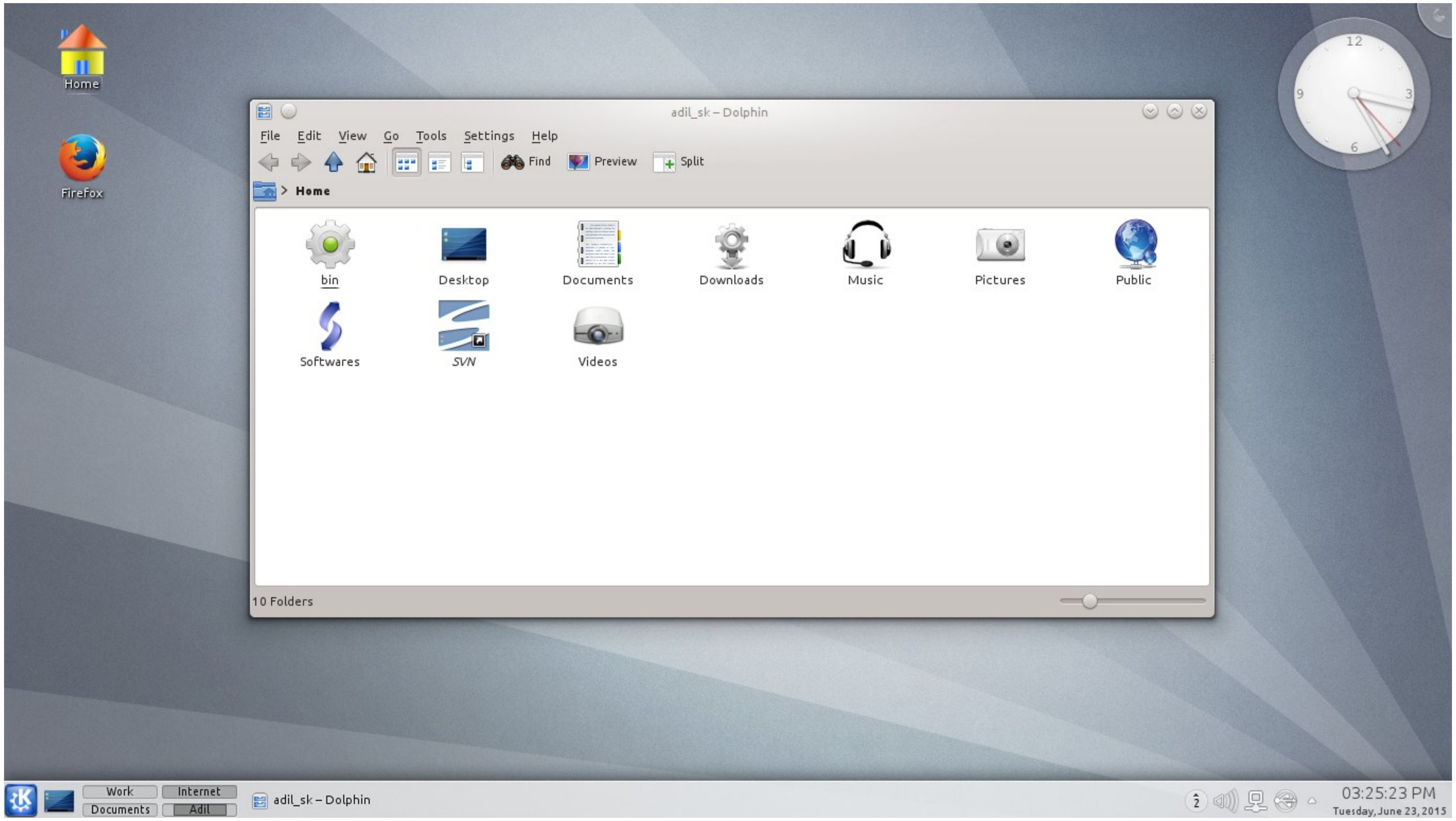
/home	User home directories
/root	Root user's home directory
/bin	Essential user command binaries
/sbin	Essential super user command binaries
/lib	Essential shared libraries and kernel modules
/usr	Multi user utilities and application
/opt	Add-on application software packages
/etc	Host specific configuration
/dev	Device files
/media	Mount point for removable media
/mnt	Mount point for temporarily mounted file systems
/boot	Static boot-able images
/tmp	Temporary files
/var	Variable files (Logs)
/proc	Virtual FS documenting kernel and process status

User Interfaces

User Interfaces

GUI

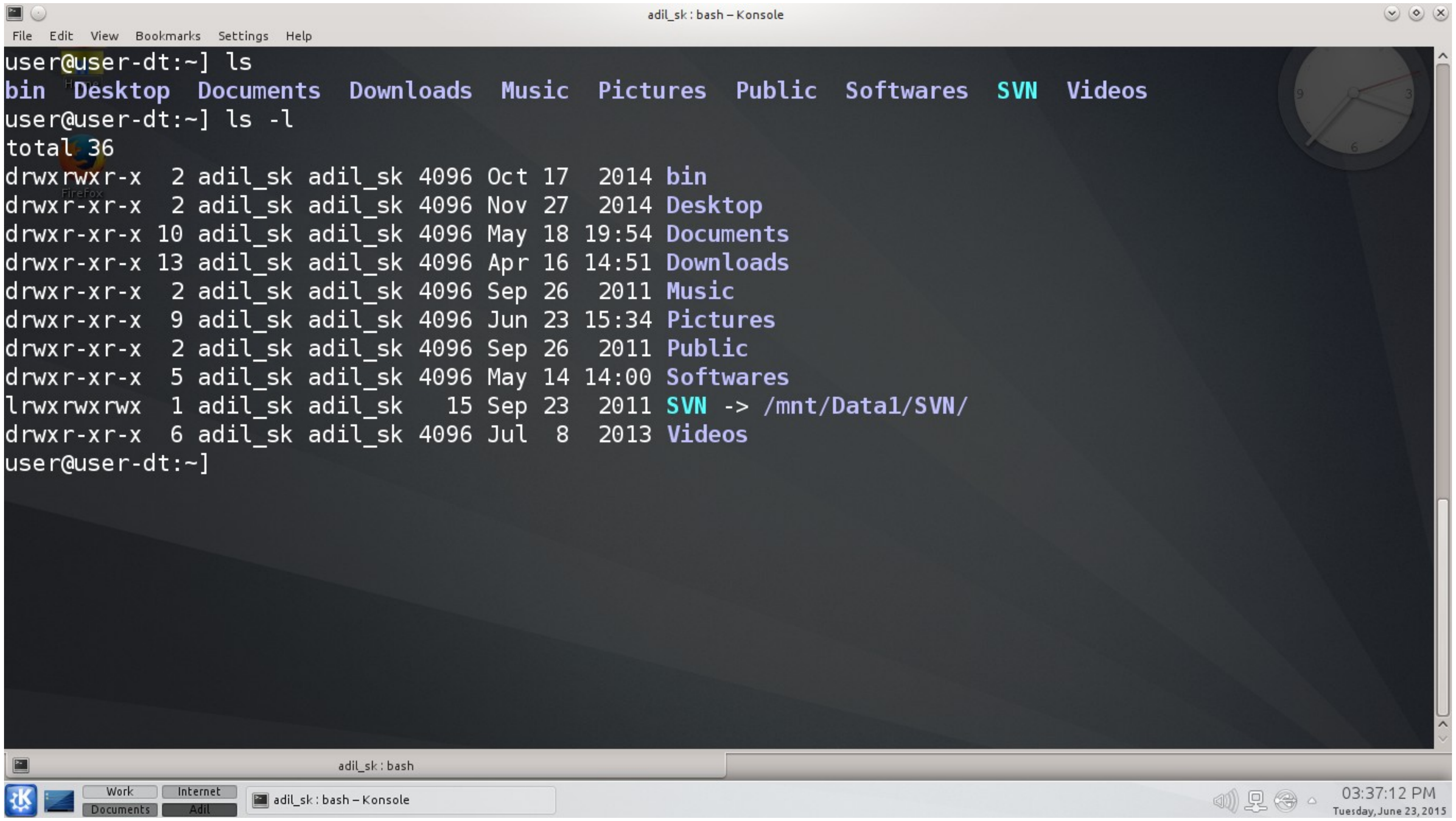
- Can use Mouse, Keyboard, Touch Screens



ΣMERTXE

User Interfaces
CLI

- Textual mode used to execute requested commands



```
user@user-dt:~] ls
bin  Desktop  Documents  Downloads  Music  Pictures  Public  Softwares  SVN  Videos
user@user-dt:~] ls -l
total 36
drwxrwxr-x  2 adil_sk adil_sk 4096 Oct 17  2014 bin
drwxr-xr-x  2 adil_sk adil_sk 4096 Nov 27  2014 Desktop
drwxr-xr-x 10 adil_sk adil_sk 4096 May 18 19:54 Documents
drwxr-xr-x 13 adil_sk adil_sk 4096 Apr 16 14:51 Downloads
drwxr-xr-x  2 adil_sk adil_sk 4096 Sep 26  2011 Music
drwxr-xr-x  9 adil_sk adil_sk 4096 Jun 23 15:34 Pictures
drwxr-xr-x  2 adil_sk adil_sk 4096 Sep 26  2011 Public
drwxr-xr-x  5 adil_sk adil_sk 4096 May 14 14:00 Softwares
lrwxrwxrwx  1 adil_sk adil_sk   15 Sep 23  2011 SVN -> /mnt/Data1/SVN/
drwxr-xr-x  6 adil_sk adil_sk 4096 Jul  8  2013 Videos
user@user-dt:~]
```

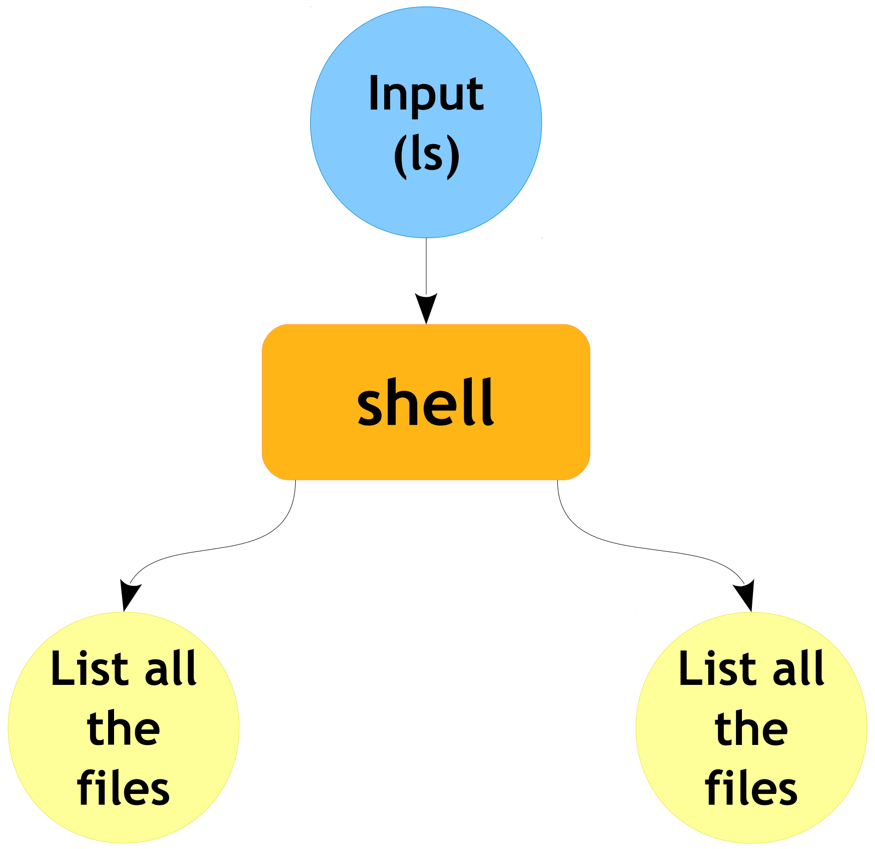
User Interfaces
The Shell

- What is a Shell
- Types of shells
  - Login
  - Non Login
    - sh
    - bash
    - ksh
    - csh
- Hands on:
  - cat /etc/shells
  - echo $0

User Interfaces

The Shell - Invocation

- The main task of a shell is providing a user environment



User Interfaces

The Shell - bash

- Bash – The 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

User Interfaces

The Shell – Environmental 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

User Interfaces

The Shell – Environmental Variables

- env - lists shell environment variable/value pairs
- export [var_name] - exports/sets a shell variable
  - HOME - path to 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

Basic Shell Commands
Types

- An executable program like all those files can have in /usr/bin.
- A command built into the shell itself. bash provides a number of commands internally called shell built-ins The cd command, for example, is a shell built-in
- A shell function. These are miniature shell scripts incorporated into the environment.
- An alias. Commands that you can define yourselves, built from other commands.
- To know the type, try

```
$ type <command>
```

Basic Shell Commands

Information

- ls - list's all the files
- pwd - gives present working directory
- man - gives information about command
- info <topic> - information pages on <topic>
- which - shows full path of command
- df - disk free
- du - disk usage
- stat - File and Inode information
- uname - print system information

Basic Shell Commands

User Specific

- All Accesses into a Linux System are through a User
- User related Shell Command Set
  - useradd - create user
  - userdel - delete user
  - su - [username] - start new shell as different user
  - finger - user information lookup
  - passwd - change or create user password
  - who, w - to find out who is logged in
  - whoami - who are you

Basic Shell Commands
Remote Access

- ssh - (secured login) is a program for logging into a remote machine and for executing commands on a remote machine.

  Example: ssh username@ipaddress

- scp - (secured copy) copies files between hosts on a network.

  Example: scp filename username@ipaddress:/path/

Basic Shell Commands
File System related

- Every thing is viewed as a file in Linux. Even a directory is a file.
  - mount - Mounting filesystem
  - find, locate - Search for files
  - cd - change directory.
  - cp - copy
  - mv - move, rename
  - rm – remove

Basic Shell Commands

File System related

- mkdir - make directory
- rmdir - remove directory
- cat, less, head, tail - used to view text files
- touch - create and update files
- wc - counts the number of lines in a file

Basic Shell Commands

Archiving

- gzip - This will compress folder or file
- gunzip - This will uncompress
- tar - Archiving files

Basic Shell Commands

Filters

- Filters are the programs, which read some input, perform the transformation on it and gives the output. Some commonly used filters are as follow
  - tail - Print the last 10 lines of each FILE to standard output.
  - sort - Sort lines of text files
  - tr - Translate, squeeze, and/or delete characters from standard input, writing to standard output.
  - wc - Print newline, word, and byte counts for each file

Basic Shell Commands
Pattern Matching

- Grep is pattern matching tool used to search the name input file. Basically its used for lines matching a pattern

  Example:

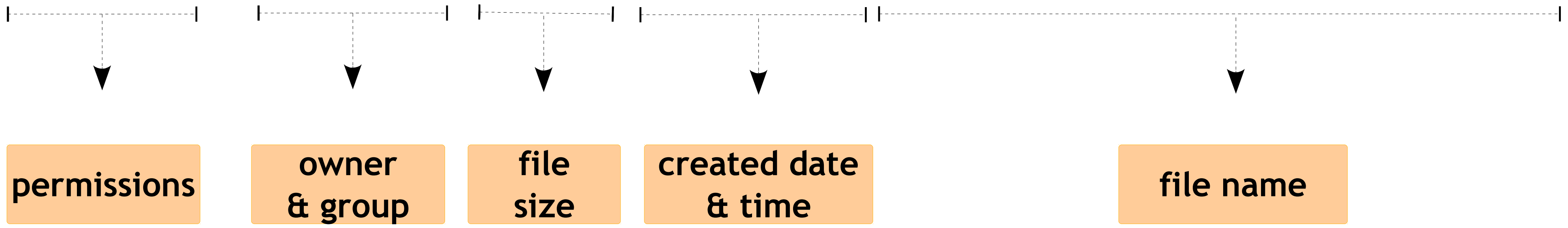
  ```
  $ ls | grep *.c
  ```

  This will list the files from the current directory with .c extension

Files

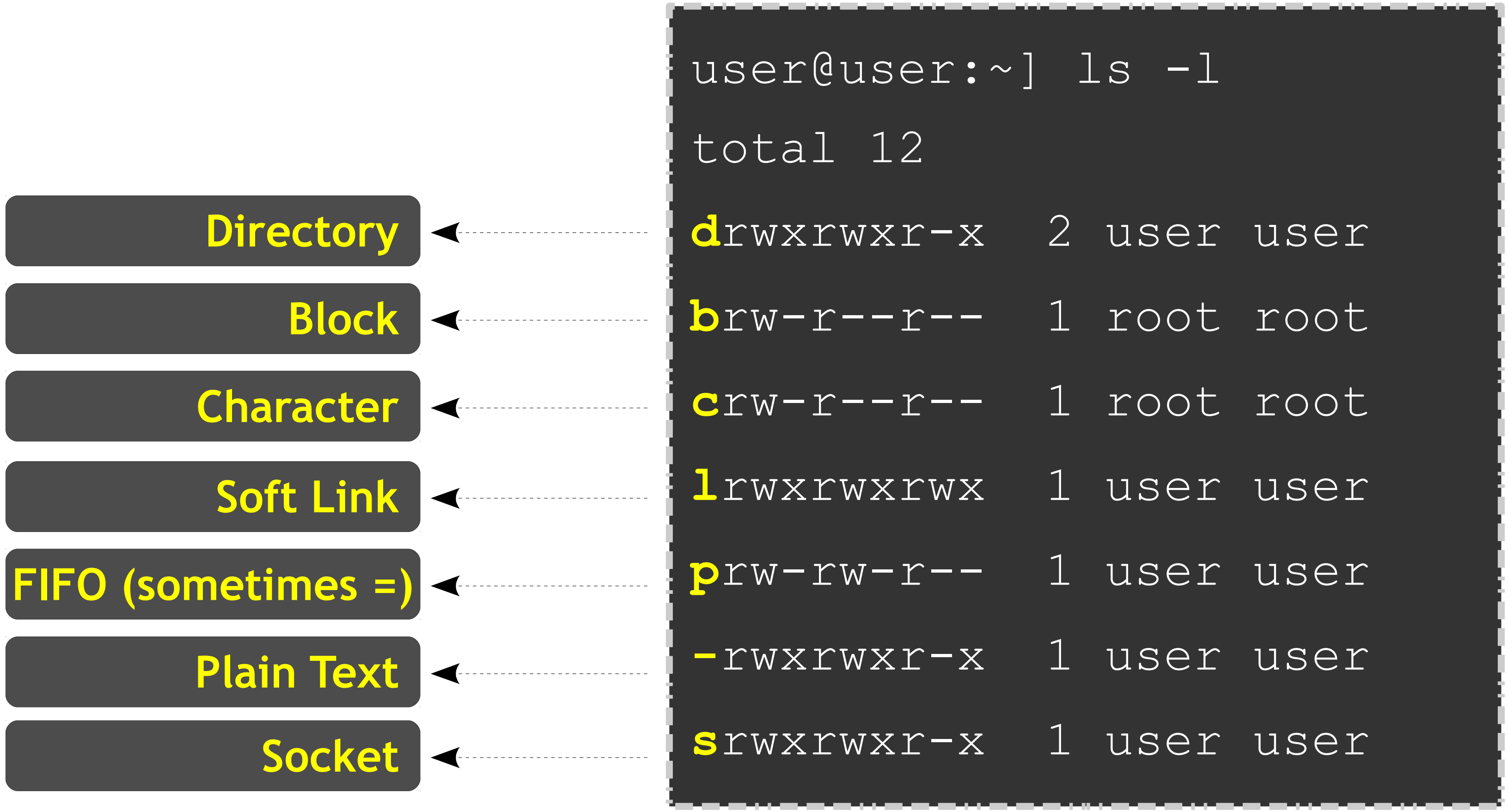
Listing

```
user@user:~] ls -l
total 12
drwxrwxr-x  2 user user     4096 Jun 23 16:48 A-Direcory
brw-r--r--  1 root root     7, 0 Jun 23 16:55 block_file
crw-r--r--  1 root root   108, 0 Jun 23 16:49 character_file
lrwxrwxrwx  1 user user       12 Jun 23 16:50 link_to_regular_file -> regular_file
prw-rw-r--  1 user user        0 Jun 23 16:50 named_pipe
-rw-rw-r--  1 user user        0 Jun 23 16:48 regular_file
-rwxrwxr-x  1 user user     7639 Jun 23 16:54 server
srwxrwxr-x  1 user user        0 Jun 23 16:55 server_socket
```



Files

Types



```
user@user:~] ls -l
total 12
drwxrwxr-x  2 user user
brw-r--r--  1 root root
crw-r--r--  1 root root
lrwxrwxrwx  1 user user
prw-rw-r--  1 user user
-rwxrwxr-x  1 user user
srwxrwxr-x  1 user user
```

Files

Permissions

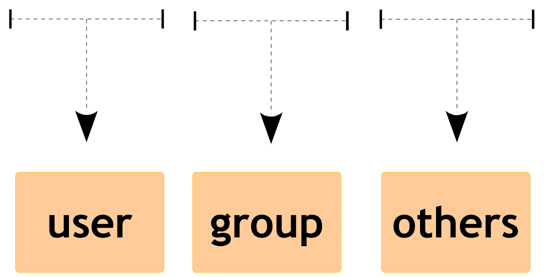
```
user@user:~] ls -l
total 1
-rwxrwxr-x  2 user user
```

	Value used to Set
➤ Execute	001 - 1
➤ Write	010 - 2
➤ Read	100 - 4

Files

Permissions

```
user@user:~] ls -l
total 1
-rwxrwxrwx  2 user user
```



Files

Permission Modification

- chmod – Change file permissions
- chown – Change file owner
- chmod [ug+r, 746] file.txt
- chown -R user:group [filename | dir]

I/O Redirection
Output

- Output redirection (>)
- Redirecting to append (>>)
- Redirecting the error (2>)

Examples:

```
$ ls > /tmp/output_file
```

Redirects the output of ls to output_file

```
$ ls -l >> /tmp/output_file
```

Appends the output of ls -l to output_file

```
$ ls 2> /tmp/output_file
```

Redirects the error output of ls to output_file

ΣMERTXE

I/O Redirection

Pipe

- 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

  Example:

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

Visual Editor - vi

Visual Editor - vi

- Screen-oriented text editor originally created for the Unix operating system
- The name vi is derived from the shortest unambiguous abbreviation for the ex command visual
- Improved version is called as vim
- To open a file

```
$ vi <filename>
```

or

```
$ vim <filename>
```

Visual Editor - vi

- vi opens a file in command mode to start mode.
- 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

Visual Editor – vi

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
  - k move up one line
  - h line move one character to the left
  - l line move one character to the right
  - j move down one line
- Yes! Arrow keys also do work. But these makes typing faster

Visual Editor – vi

Basic Commands

- Open a file

  `$ vi <file_name>`

- How to exit

:q	-> Close with out saving.
:wq	-> Close the file with saving.
:q!	-> Close the file forcefully with out saving

- Already looks too complicated?
- Try by yourself, let us write a C program

Visual Editor – vi

Escape / 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.

Visual Editor – vi

Escape / Command Mode

- File mode
  - Changing (Replacing) Text

    :%s/first/sec - Replaces the first by second every where in the file

    :%s/orange/apple/gc - For all lines in a file, find string "orange" and replace with string "apple" for each instance on a line
  - File Interactions (edit and read)

    :e filename - open another file without closing the current

    :r filename - reads file named filename in place
  - Editor Settings

    :set all - display all settings of your session

Visual Editor – vi

Escape / Command Mode – Useful Shortcuts

Command	Operation
G	Go to last line of the file
gg	Go to first line of the file
.	Repeat the previous command
Ctrl a	Increment number under the cursor by 1
Ctrl x	Decrements numbers under the cursor by 1
J	Joining the two adjacent lines
(n)gg	Move cursor to **n**th line

ΣMERTXE

Visual Editor – vi

Editing Mode

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
I	**Insert**	beginning of the current line
o	**Open**	new line below the current line
O	**Open**	new line above the current line

Visual Editor – vi

Editing Mode – 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	Operation
dd	For deleting a line
(n)dd	For deleting a n lines
x	To delete a single character
D	Delete contents of line after cursor
dw	Delete word
(n)dw	Delete n words

Visual Editor – vi

Visual Mode – Editing Text

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

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

Branch Office:
Emertxe Information Technologies,
No-1, 9th Cross, 5th Main,
Jayamahal Extension,
Bangalore, Karnataka 560046

Corporate Headquarters:
Emertxe Information Technologies,
83, Farah Towers, 1st Floor,
MG Road,
Bangalore, Karnataka - 560001
T: +91 809 555 7333 (M), +91 80 41289576 (L)
E: training@emertxe.com

https://www.facebook.com/Emertxe

https://twitter.com/EmertxeTweet

https://www.slideshare.net/EmertxeSlides

Thank You