Scripting & Computer Environments

Intro to Linux

IIIT-H

Aug 8, 2015

...Previously & Today...

Previously:

- The basics of computation: the hows and whys.
- Basics of operating systems
  - Types
  - Functions

Today:

- Linux Intro
- Basic commands

Brainstorm

1. Free and open source software (FOSS)? Advantages?
2. Linux? GNU/Linux? Kernel? Shell?
3. Internal & external commands you are familiar with?

Brainstorm

1. Free and open source software (FOSS)? Advantages?
2. Linux? GNU/Linux? Kernel? Shell?
3. Internal & external commands you are familiar with?

Brainstorm

1. Free and open source software (FOSS)? Advantages?
2. Linux? GNU/Linux? Kernel? Shell?
3. Internal & external commands you are familiar with?

Unix & Linux

- Unix is a multi-tasking, multi-user OS.
- A basis for many OSs:

Example

Berkeley Software Distribution - `BSD` (`NetBSD`, `OpenBSD`, and `FreeBSD`), Sun's `Solaris` (now `Oracle Solaris`) & the open source `OpenSolaris` (no more), `GNU/Linux` , `OS X`, `Android`, etc.

- Linux is a Unix-like open source OS (OSS).
- Linus Torvalds wrote the core component (the kernel).
- Technically, Linux refers to this core part.

Evolution of a Revolution

- 1969 - C developed at AT&T.
- 1973 - `UNIX` rewritten in C & the code shared (to UC, Berkeley too).
- By 1975 - AT&T started selling `UNIX` ($\sim$ half written by others).
- As a result, two versions: AT&T `Unix` and `BSD Unix`.
- In the 80s, companies wrote their own versions.

  e.g. `IBM's AIX`, Microsoft's `Xenix`, Sun's `SunOS` (later `SunSolaris`)
- 1983 - Richard Stallman started the `GNU (GNU's Not Unix)` project to distribute free Unix-like software.
- In the 90s, Linus wrote the kernel for his 386 system and shared it online.

MEMORABLE LINUX MILESTONES

CELEBRATING 20 YEARS OF LINUX

1991 — LINUS TORVALDS POSTS FAMOUS MESSAGE - *"HELLO EVERYBODY OUT THERE..."* - AND RELEASES FIRST LINUX CODE

1992 — LINUS LICENSES LINUX UNDER THE GPL, AN IMPORTANT DECISION THAT WILL CONTRIBUTE TO ITS SUCCESS IN THE COMING YEARS

1993 — SLACKWARE BECOMES FIRST WIDELY ADOPTED DISTRIBUTION

1996 — LINUS VISITS AQUARIUM, GETS BIT BY A PENGUIN AND CHOOSES IT AS LINUX MASCOT

1998 — TECH GIANTS BEGIN ANNOUNCING PLATFORM SUPPORT FOR LINUX

1999 — RED HAT GOES PUBLIC

2003 — IBM RUNS FAMOUS LINUX AD DURING THE SUPERBOWL

2005 — LINUS APPEARS ON THE COVER OF BUSINESSWEEK WITH A STORY THAT HAILS LINUX AS A BUSINESS SUCCESS

2007 — THE LINUX FOUNDATION IS FORMED TO PROMOTE PROTECT AND STANDARDIZE LINUX LINUS IS A FELLOW

2010 — THE LINUX-BASED ANDROID OS OUTSHIPS ALL OTHER SMARTPHONE OSES IN THE U.S. AND CLIMBS TO DOMINANCE

2011 — LINUX TURNS 20 AND POWERS THE WORLD'S SUPERCOMPUTERS, STOCK EXCHANGES, PHONES, ATMS, HEALTHCARE RECORDS, SMART GRIDS, THE LIST GOES ON

20 YEARS OF LINUX 1991 - 2011

THE LINUX FOUNDATION
http://www.linuxfoundation.org/

GNU

The goal of GNU (GNU's Not Unix):

> *"To create complete UNIX-compatible software systems entirely composed of free software."* *Richard Stallman*

- Unix-like but no unix code (hence GNU).
- The movement created many popular tools (`emacs`, `gcc`, `gdb`...).

GNU

The goal of GNU (GNU's Not Unix):

> *"To create complete UNIX-compatible software systems entirely composed of free software."* *Richard Stallman*

- Unix-like but no unix code (hence GNU).
- The movement created many popular tools (`emacs`, `gcc`, `gdb`...).

`GNU/Linux`

"There really is a Linux, and these people are using it, but it is just a part of the system they use. Linux is the kernel: the program in the system that allocates the machines resources to the other programs that you run. Linux is normally used in combination with the GNU operating system: the whole system is basically GNU with Linux added, or GNU/Linux." *Richard Stallman*

Some Licenses

- `Public domain` - authors waive all copyright.
  - can be used, reproduced/executed freely, without permission/payment.
- `Freeware` - Not public domain software or FOSS.
  - proprietary software (often) that you can use without paying a license cost (e.g. `Adobe Reader`, `Skype` ...)
- "`MIT/BSD`" `licences` - you can copy/redistribute/modify the software in any way as long as you:
  - respect the identity and rights of the author
  - agree not to sue the author over software quality
- `GNU GPL` - `Free Software Licenses` - `copyleft` philosophy
  - You can reproduce/modify/distribute but requires that any resulting copies or adaptations be bound by the same licensing agreement

Some Licenses

- `Public domain` - authors waive all copyright.
  - can be used, reproduced/executed freely, without permission/payment.
- `Freeware` - Not public domain software or FOSS.
  - proprietary software (often) that you can use without paying a license cost (e.g. `Adobe Reader, Skype` ...)
- "MIT/BSD" licenses - you can copy/redistribute/modify the software in any way as long as you:
  - respect the identity and rights of the author
  - agree not to sue the author over software quality
- GNU/GPL (GNU General Public License) - *copyleft* philosophy
  - You can reproduce/modify/distribute but requires that any resulting copies or adaptations be bound by the same licensing agreement.

Some Licenses

- `Public domain` - authors waive all copyright.
  - can be used, reproduced/executed freely, without permission/payment.
- `Freeware` - Not public domain software or FOSS.
  - proprietary software (often) that you can use without paying a license cost (e.g. `Adobe Reader, Skype` ...)
- `"MIT/BSD" licences` - you can copy/redistribute/modify the software in any way as long as you:
  - respect the identity and rights of the author
  - agree not to sue the author over software quality
- `GNU/GPL` (GNU General Public License) - *copyleft* philosophy
  - You can reproduce/modify/distribute but requires that any resulting copies or adaptations be bound by the same licensing agreement.

Some Licenses

- `Public domain` - authors waive all copyright.
  - can be used, reproduced/executed freely, without permission/payment.
- `Freeware` - Not public domain software or FOSS.
  - proprietary software (often) that you can use without paying a license cost (e.g. `Adobe Reader, Skype` ...)
- `"MIT/BSD" licences` - you can copy/redistribute/modify the software in any way as long as you:
  - respect the identity and rights of the author
  - agree not to sue the author over software quality
- `GNU/GPL (GNU General Public License)` - *copyleft* philosophy
  - You can reproduce/modify/distribute but requires that any resulting copies or adaptations be bound by the same licensing agreement.

Linux: The Whys

- Free! (as in 'free speech')
- Portable
- Prevalence, Scalability & Versatility
  - Most leading hosting companies' servers run Linux (source: here)
  - 95.2% of the top fastest supercomputers (source: here)
- Large community base
- Security

Some cons:

- Many distribution choices
- Lag in some software support (e.g. Photoshop, games)

Linux: The Whys

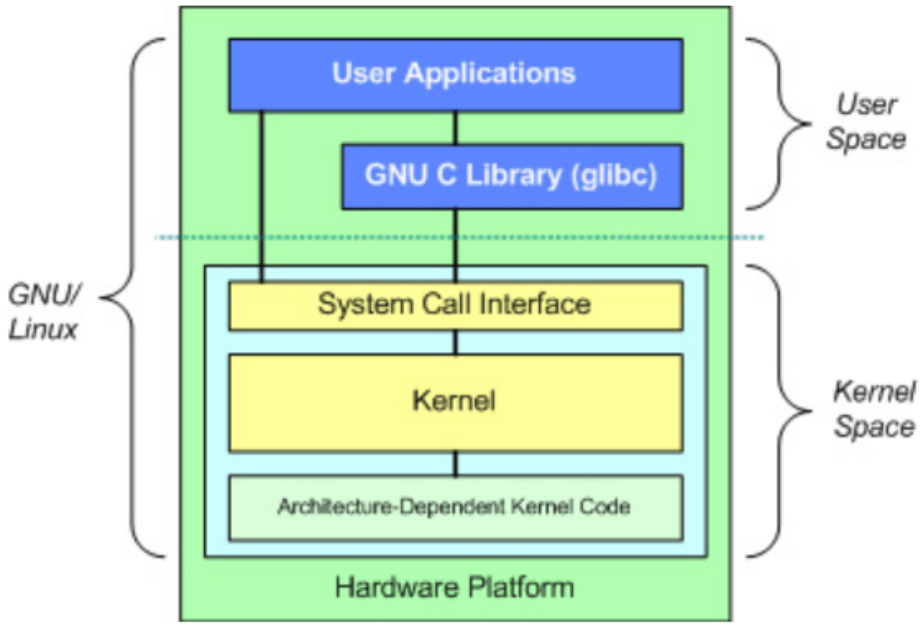
- Free! (as in 'free speech')
- Portable
- Prevalence, Scalability & Versatility
  - Most leading hosting companies' servers run Linux (source: here)
  - 95.2% of the top fastest supercomputers (source: here)
- Large community base
- Security

Some cons:

- Many distribution choices
- Lag in some software support (e.g. Photoshop, games)

Linux Architecture

The Standard OS Model

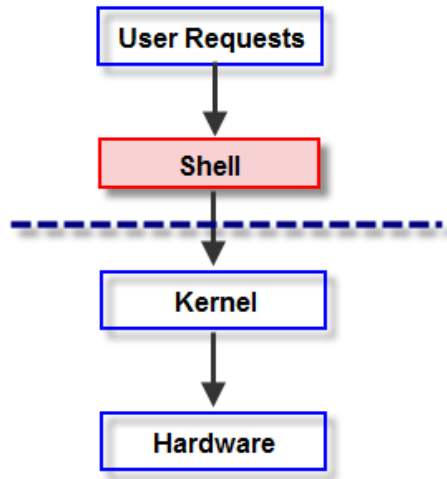


Basic Architecture of the GNU/Linux OS

Linux Architecture: The Shell

A program (a.k.a. command line interpreter) that allows the user to interact with the `UNIX/Linux` system.

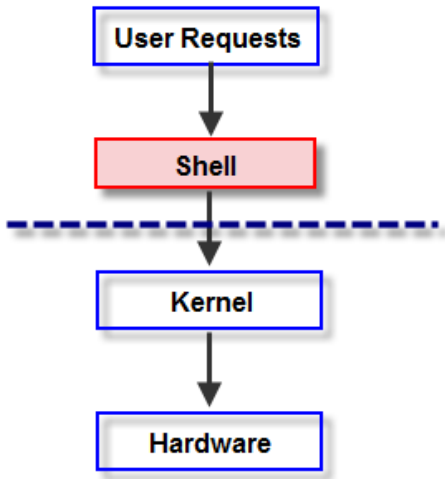
- Reads user's input.
- Parses it (evaluates special characters if any).
- Works with the kernel to execute the command.



Linux Architecture: The Shell

A program (a.k.a. command line interpreter) that allows the user to interact with the `UNIX/Linux` system.

- Reads user's input.
- Parses it (evaluates special characters if any).
- Works with the kernel to execute the command.



`Shell Script`

A regular text file that contains executable shell or `Linux` commands.

Examples

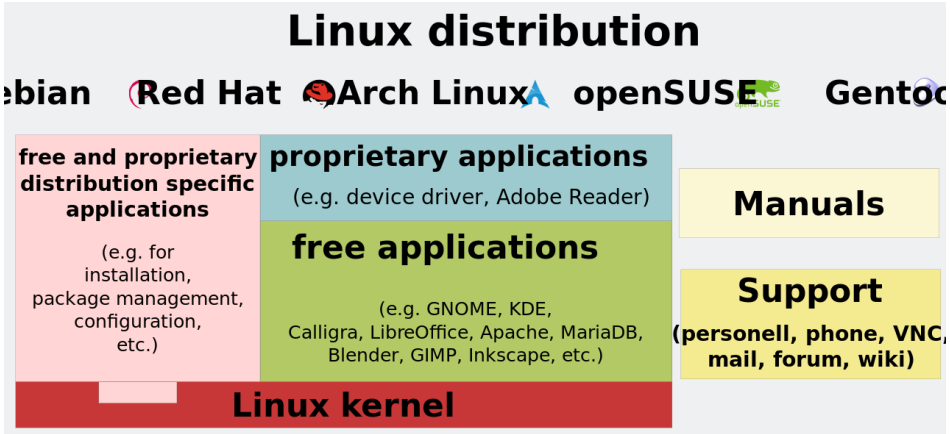
`Bourne shell (sh)`
`Bourne again shell (Bash)` ✓
`C shell (csh, tcsh)`
`Korn shell (ksh)`

Linux Distributions (Distros)

Distro = a collection of software, often open source, on top of a kernel.

- Different vendors distribute kernel + GNU + non-GNU components (e.g. desktop applications, server software, system management tools, documentation...)

- 300+ active distros; some more popular:



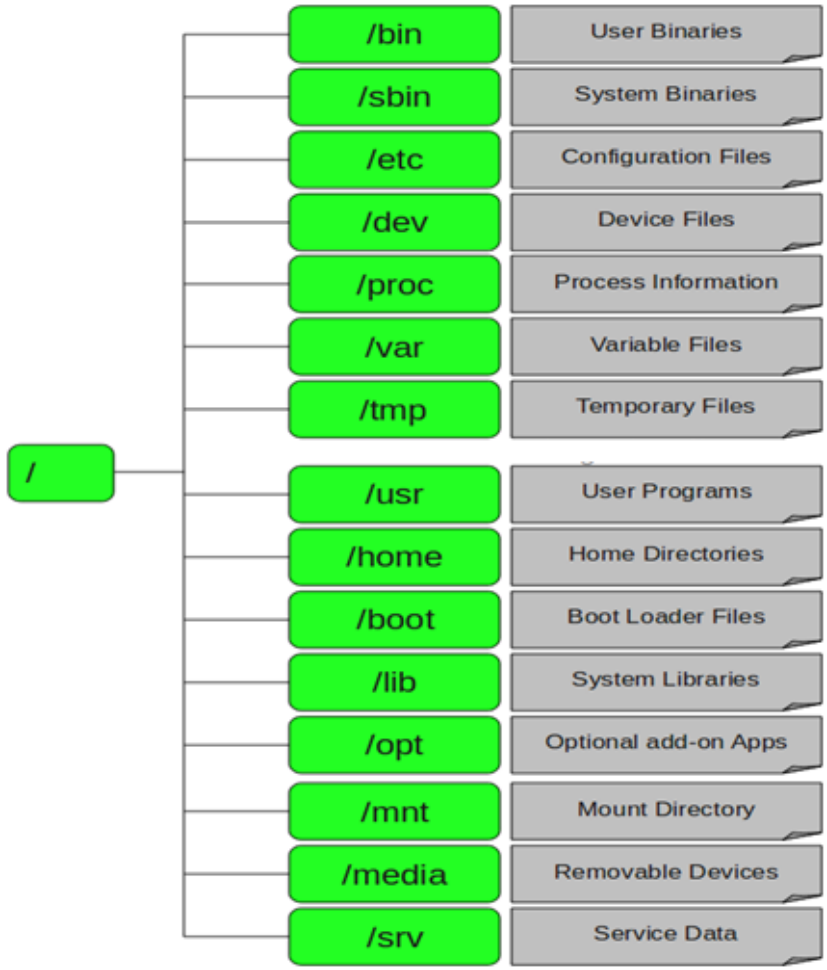
Getting Started: Installation

1. Disk partitioning & dual boot.
   - Download the preferred ISO file (CD image)
   - Burn it unto CD, boot from it and follow the wizard.
2. Run it within a virtual environment.

   e.g. `VMware, Oracle VM VirtualBox` ...
3. Other options for Windows users:
   - Linux Live CD
   - Linux on a USB drive (Live USB)
   - `cygwin` - a Linux-like environment for Windows, `Wubi`, etc

Linux: The File System

- In Linux, everything is a file!
- Hierarchical organization
- Absolute vs Relative paths
  - $\sim$ (tilde) - the home directory
  - . (a dot) - the current directory
  - .. (double dot) - the parent directory



The File System: File Types

1. `Regular/Ordinary file`: contains printable/non-printable stream of characters.
   - `Text files`: human-readable
     e.g. documentation, application settings, source code, logs
   - `Binary files`: executables, libraries, media files, ...
2. `Directory file`: maintains info about files it houses (e.g. name, inode number).
3. `Device/special file`: contains attributes of a device (e.g. printer, CD-ROM) used by the kernel.
4. Others
   - Links, Sockets, Named pipes

The File System: File Types

1. `Regular/Ordinary file`: contains printable/non-printable stream of characters.
   - `Text files`: human-readable
     e.g. documentation, application settings, source code, logs
   - `Binary files`: executables, libraries, media files, ...

2. `Directory file`: maintains info about files it houses (e.g. name, inode number).

3. `Device/Special file`: contains attributes of a device (e.g. printer, CD-ROM) used by the kernel.

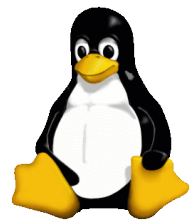
4. Others
   - `Links, Sockets, Named pipes ...`

The File System: File Types

1. `Regular/Ordinary file`: contains printable/non-printable stream of characters.
   - `Text files`: human-readable
     e.g. documentation, application settings, source code, logs
   - `Binary files`: executables, libraries, media files, ...
2. `Directory file`: maintains info about files it houses (e.g. name, inode number).
3. `Device/special file`: contains attributes of a device (e.g. printer, CD-ROM) used by the kernel.
4. Others
   - `Links, Sockets, Named pipes ...`

The File System: File Types

1. `Regular/Ordinary file`: contains printable/non-printable stream of characters.
   - `Text files`: human-readable
     e.g. documentation, application settings, source code, logs
   - `Binary files`: executables, libraries, media files, ...
2. `Directory file`: maintains info about files it houses (e.g. name, inode number).
3. `Device/special file`: contains attributes of a device (e.g. printer, CD-ROM) used by the kernel.
4. Others
   - `Links, Sockets, Named pipes ...`



Basic Commands

- Internal vs External commands?
- How does the shell locate commands?

General Syntax

```
<SomeCommand> [option 1] [option 2] ...[option n]
```

- Internal vs External commands?
- How does the shell locate commands?

General Syntax

```
<SomeCommand> [option 1] [option 2] ...[option n]
```

- Internal vs External commands?
- How does the shell locate commands?

General Syntax

```
<SomeCommand> [option 1] [option 2] ...[option n]
```

Print Working Directory

`pwd`

- Displays full path of the current directory.

- Internal vs External commands?
- How does the shell locate commands?

General Syntax

```
<SomeCommand> [option 1] [option 2] ...[option n]
```

Print Working Directory

`pwd`

- Displays full path of the current directory.

The List command

`ls [flags] [file]`

- Lists directory content
- Flags/options: -l, -a, -s, -S, -t ...

Basic Linux Commands (2)

Change Directory

`cd [dir]`

- Changes directory to `[dir]`.
- Defaults to user's home directory if `<dir>` not given.

Basic Linux Commands (2)

Change Directory

`cd [dir]`

- Changes directory to `[dir]`.
- Defaults to user's home directory if <dir> not given.

Know Your System

- `echo $SHELL`
- `uname [-a]`
- `whoami`
- `w(ho)`
- `ifconfig [-a]`
- `route`
- `df -h, du -h, free -m`

File & Directory Commands: Creation

Creating Files

`touch [flags] <file>` (easiest way)

- If the file exists, timestamp modified.
- If not, the file is created.

File & Directory Commands: Creation

Creating Files

`touch [flags] <file>` (easiest way)

- If the file exists, timestamp modified.
- If not, the file is created.

Creating Directories

`mkdir [flags] <dir name>`

- Creates a directory with the name `<dir name>`.

File & Directory Commands: Display

View + Concatenate

```
cat <file>
cat <file1> <file2> ...<file n>
od   [flags] <file>        (octal display)
    e.g.  od -bc /bin/ls
```

File & Directory Commands: Display

View + Concatenate

```
cat <file>
cat <file1> <file2> ...<file n>
od   [flags] <file>      (octal display)
    e.g.  od -bc /bin/ls
```

More & Less

`more <filename>`

Scrolls 1 page @ a time (space bar)

`less <filename>`

Scrolls up/down by pages/lines

File & Directory Commands: Display

View + Concatenate

```
cat <file>
cat <file1> <file2> ...<file n>
od   [flags] <file>        (octal display)
    e.g.  od -bc /bin/ls
```

More & Less

`more <filename>`

Scrolls 1 page @ a time (space bar)

`less <filename>`

Scrolls up/down by pages/lines

Head & Tail

`head -[numlines] <filename>`

`tail -[numlines] <filename>`

Copy

`cp [flags] <file> <destination>`

- Copies the file `<file>` to a location `<destination>`.
- Use `-r` flag to copy an entire directory.

Move

`mv [flags] <source> <destination>`

`mv [flags] <oldname> <newname>` (rename)

- Moves a file or directory from `<source>` to `<destination>`.
- Recurses for directories automatically (unlike `cp`).

File & Directory Commands: Movement

Copy

`cp [flags] <file> <destination>`

- Copies the file `<file>` to a location `<destination>`.
- Use `-r` flag to copy an entire directory.

Move

`mv [flags] <source> <destination>`
`mv [flags] <oldname> <newname>` (rename)

- Moves a file or directory from `<source>` to `<destination>`.
- Recurses for directories automatically (unlike `cp`).

File & Directory Commands: Deletion

"Remove" File

```
rm [flags] <file>
rm -i <filename>                (prompts - good idea!)
alias rm="rm -i"                is called aliasing
```

- Be cautious!
- Use wildcards (more on them later) to delete multiple files.

Remove Directory

```
rmdir [flags] <directory>       (empty directory)
rm -r <directory>               (directories + subdirectories)
```

Be extremely cautious!!

File & Directory Commands: Deletion

"Remove" File

```
rm [flags] <file>
rm -i <filename>                (prompts - good idea!)
alias rm="rm -i"                is called aliasing
```

- Be cautious!
- Use wildcards (more on them later) to delete multiple files.

Remove Directory

```
rmdir [flags] <directory>          (empty directory)
rm -r <directory>          (directories + subdirectories)
```

Be extremely cautious!!

Getting Help

The manual command

`man <section> <command>` section # = {1-8}

- Displays the manual page (manpage) of `<command>`.
- Use `/<keyword>` to do a keyword search in a manpage
- Make `man` your best friend!

Getting Help

The manual command

`man <section> <command>` section # = {1-8}

- Displays the manual page (manpage) of `<command>`.
- Use `/<keyword>` to do a keyword search in a manpage
- Make `man` your best friend!

`info` & `whatis`

`info <command>`

`whatis <command>`

Getting Help

The manual command

`man <section> <command>` section # = {1-8}

- Displays the manual page (manpage) of `<command>`.
- Use `/<keyword>` to do a keyword search in a manpage
- Make `man` your best friend!

`info` & `whatis`

`info <command>`

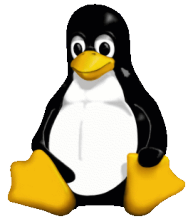
`whatis <command>`

Search/Locate commands

- `apropos <keyword>`

  Finds commands by keyword.
- `which <command>`
- `whereis <command>`

Next ...



Working with Files