Scripting & Computer Environments

Linux Fundamentals

IIIT-H

Aug 3, 2013

...Previously & Today...

Previously:

- The whats, whys and hows of computation
- Basics of operating system
  - Types
  - Functions

Today:

- The `Linux` OS

Unix & Linux

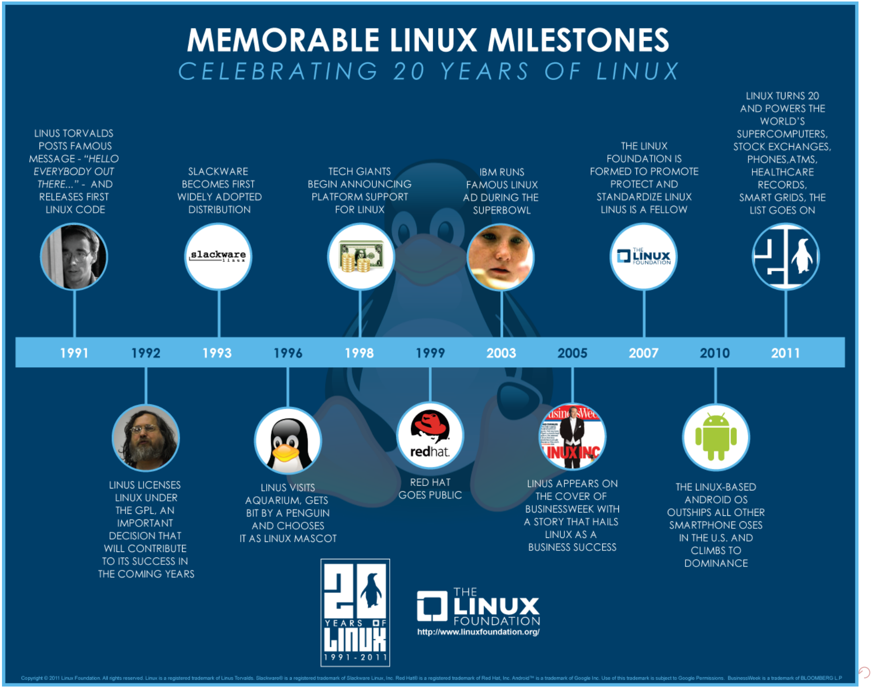
- Unix is a multi-tasking, multi-user OS
- A basis for many OSs:

> **Example**
>
> Berkeley Software Distribution - BSD (NetBSD, OpenBSD, FreeBSD), Sun's Solaris - recently Open Solaris, GNU/Linux, OS X, Android, etc

- Linux is a Unix-like open-source OS (OSS).
- Linus Trovalds 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 rewrote in C & the code shared (to UC, Berkeley too).
- By 1975 - AT&T started selling UNIX (~ 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, SunOS` (later `SunSolaris`), etc
- 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

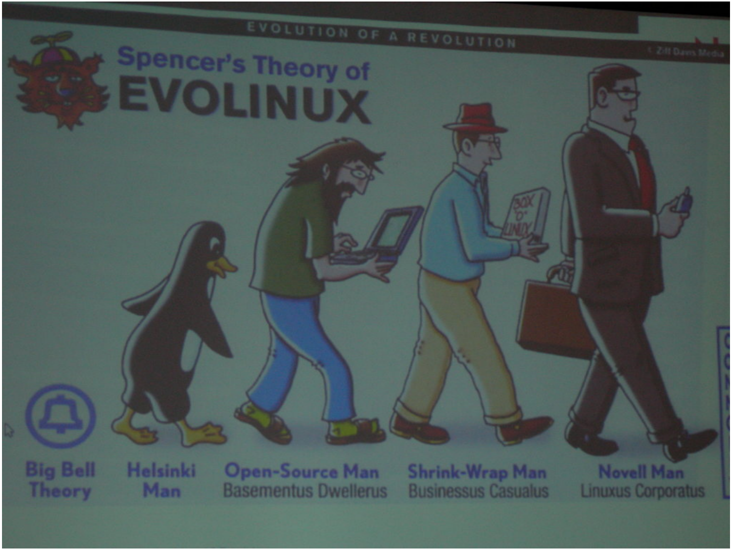
2005 — LINUS APPEARS ON THE COVER OF BUSINESSWEEK WITH A STORY THAT HAILS LINUX AS A BUSINESS SUCCESS

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

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



source: Ohio LinuxFest 2005

The OSS Philosophy: 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

The OSS Philosophy: Licensing

1. `Public domain` - authors waive all copyright.
   - Can be used, reproduced/executed freely, without permission/fee.

2. `Freeware` - Not public domain software or FOSS.
   - Proprietary software that you can use without paying a license cost (e.g. `Adobe Reader, Skype` ...)

3. `"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

4. `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.

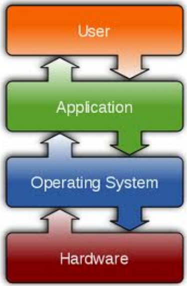
Why Linux?

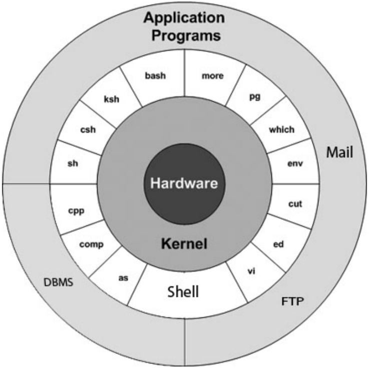
- Free! Trustworthy??
- Portable
- Prevalence,Scalability & versatility
  - Most leading hosting companies' servers run Linux (source: here).
  - 95.2% of the top fastest supercomputers (source: here)
- Large community base
- Security

Some cons:

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

Linux Architecture





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.

Example

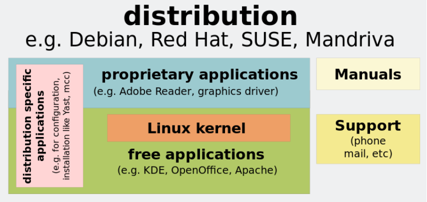
```
Bourne shell (sh)
Bourne again shell (bash)
C shell (csh, tcsh)
Korn shell (ksh)
```

Popular Distributions (Distros)

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+ distros; some more popular:

debian
fedora
gentoo linux
KNOPPIX
Mandriva
MEPIS
Novell
redhat
slackware
SUSE
ubuntu
xandros



distribution

e.g. Debian, Red Hat, SUSE, Mandriva

distribution specific applications
(e.g. for configuration, installation like Yast, mcc)

proprietary applications
(e.g. Adobe Reader, graphics driver)

Linux kernel

free applications
(e.g. KDE, OpenOffice, Apache)

Manuals

Support
(phone mail, etc)

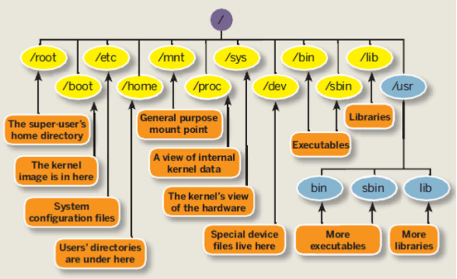
Getting Started: Installation

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, VBox` ...
3. Other options for Windows users:
   - Linux live cd
   - `cygwin`: a Linux-like environment for Windows
   - Linux on a USB drive

The Linux File System

- In Linux everything is a file!
- Hierarchical organization


/
/root
/etc
/mnt
/sys
/bin
/lib
/boot
/home
/proc
/dev
/sbin
/usr
The super-user's home directory
The kernel image is in here
System configuration files
Users' directories are under here
General purpose mount point
A view of internal kernel data
The kernel's view of the hardware
Special device files live here
Executables
Libraries
bin
sbin
lib
More executables
More libraries

- Absolute path vs Relative path
  - ∼ (tilde) - the home directory
  - . (full stop) - the current directory
  - .. (double full stop) - the parent directory

The Linux File System: File Types

1. Text files: human-readable
   e.g. documentation, application settings, source code, logs

2. Binary files: executables, libraries, media files, ...

- Regular/Ordinary file: contains printable/non-printable stream of characters.

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

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

Basic Commands

General Syntax

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

Print Working Directory

```
pwd
```

- Displays full path of current directory

The List command

```
ls [flags] [file]
```

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

Basic Commands

General Syntax

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

Print Working Directory

```
pwd
```

- Displays full path of current directory

The List command

```
ls [flags] [file]
```

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

Basic Commands

General Syntax

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

Print Working Directory

```
pwd
```

- Displays full path of current directory

The List command

```
ls [flags] [file]
```

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

Basic 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`

Basic 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.

Creating Directories

`mkdir [flags] <dir name>`

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

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: Displaying

View + Concatenate

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

More and Less Commands

```
more <filename>
```

Scrolls 1 page @ a time (space bar)

```
less <filename>
```

Scrolls up/down by pages/lines

Head and Tail Commands

```
head -[numlines] <filename>
```

The first/last `[numlines]` of the file

```
tail -[numlines] <filename>
```

File & Directory Commands: Displaying

View + Concatenate

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

More and Less Commands

```
more <filename>
```

Scrolls 1 page @ a time (space bar)

```
less <filename>
```

Scrolls up/down by pages/lines

Head and Tail Commands

```
head -[numlines] <filename>
```

The first/last `[numlines]` of the file

```
tail -[numlines] <filename>
```

File & Directory Commands: Displaying

View + Concatenate

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

More and Less Commands

`more <filename>`

Scrolls 1 page @ a time (space bar)

`less <filename>`

Scrolls up/down by pages/lines

Head and Tail Commands

`head -[numlines] <filename>`

The first/last `[numlines]` of the file

`tail -[numlines] <filename>`

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>                (renames)
```

- Moves a file/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>                (renames)
```

- Moves a file/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 about them later) to delete multiple files.

Remove Directory

```
rmdir [flags] <directory>                     (on 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 about them later) to delete multiple files.

Remove Directory

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

- Be extremely cautious!

Getting Help

The `manual` command

```
man <command>
```

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

The `info` command

```
info <command>
```

The `whatis` command

```
whatis <command>
```

The `apropos` command

```
apropos <keyword>
```

- Finds all commands containing the keyword.

Getting Help

The manual command

```
man <command>
```

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

The info command

```
info <command>
```

The whatis command

```
whatis <command>
```

The apropos command

```
apropos <keyword>
```

- Finds all commands containing the keyword.

Getting Help

The `man`ual command

```
man <command>
```

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

The `info` command

```
info <command>
```

The `whatis` command

```
whatis <command>
```

The `apropos` command

```
apropos <keyword>
```

- Finds all commands containing the keyword.