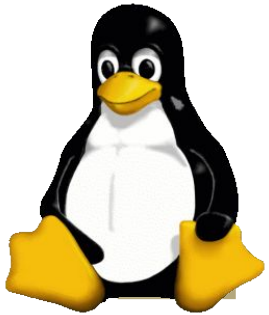


Linux - The Beginning

Maruthi S. Inukonda

09th Jan 2019

Agenda

- Introduction
- Architecture

Introduction

Brief history

Linux

- Powering 90% of today's Internet.
- Initially developed by Linus Torvalds in 1991
- Written in C and assembly language
- Open sourced under GNU Public License (GPL).
- Actively maintained by Linux community
- About 14000 passionate developers in the Linux community.
- About 1000 code changes per day.

Linus Torvalds

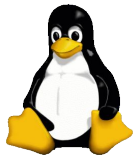
What is Linux? (1/2)

Linux is
- Multi-User
- Multi-Tasking
- Multi-Processing
- Multi-Tenant

Unix like operating system

Linux is not
- Distributed
- Cloud

operating system



What is Linux? (2/2)

- Free and secure
- Can run on 30 different architectures
  x86, ARM, ppc, …
- Can run on any size computer.
  Mobiles, PC, servers, embedded, supercomputers, …
- Being open source allows OS research & development

Linux, Fork, Distribution

Linux (more formally GNU/Linux)

- Kernel code from kernel.org
- Tools and libraries code from gnu.org
- Akin to river water

Linux fork

- Forks from Linux mainline
- Merges into mainline periodically
- Akin to tributaries

```
$ uname -o
GNU/Linux
```

Linux distribution (aka Distro)

- Tested, packaged open source software
- Optional GUI from gnome.org, kde.org, ...
- Akin to purified bottled water.

Linux distributions (1/4)

- Community supported & free
  - Cent OS
  - Fedora Core
  - Ubuntu
  - Debian
  - Open SuSe
  - Slackware
  - Arch Linux
  - many more ...

Linux distributions (2/4)

- Company supported & commercial
  - Red Hat Enterprise Linux (RHEL)
  - Ubuntu Server
  - SuSE Linux Enterprise Server (SLES)
  - Oracle unbreakable Enterprise Linux (OEL)
  - few more ...

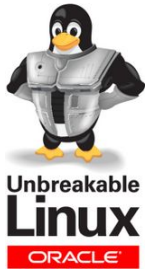
Linux distributions (3/4)

- For desktop/laptop
  - Ubuntu desktop
  - Fedora Core
  - Open SuSe
- For server
  - Ubuntu server
  - CentOS
  - RHEL/Debian/SLES/OEL
- For kernel development
  - Slackware
  - ArchLinux

Note: Command line administration of Linux is applicable for servers and desktop/laptops. This workshop focuses only on command line usage/administration.

Linux distributions (4/4)

- deb based
  - Ubuntu server/desktop
  - Debian
- rpm based
  - Fedora Core/OpenSuse
  - RHEL/CentOS/SLES/OEL

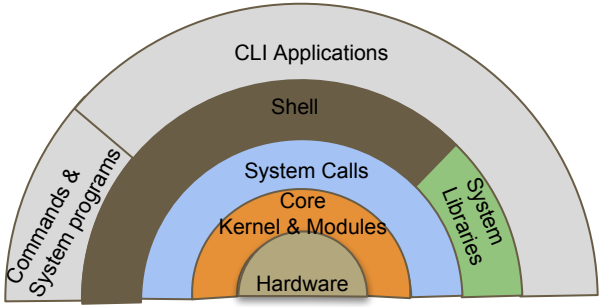



```
$ cat /etc/os-release
NAME="Ubuntu"
VERSION="16.04.5 LTS (Xenial Xerus)"
...

$ cat /etc/redhat-release
...
```

Architecture

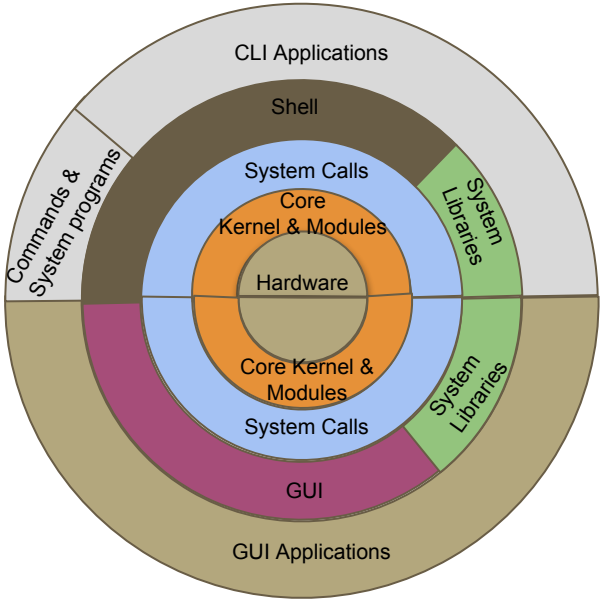
Linux Architecture - Command Line Interface (CLI)

- Hardware
  - ➢ CPU, Memory, Disk, Graphics, Network, etc
- Core Kernel & Modules
  - ➢ Process, Memory, File, Network subsystems, Device drivers
- System Calls
  - ➢ read, write, fork, exec, clone, etc
- System Libraries
  - ➢ libc, libpthread, etc
- Commands & System programs
  - ➢ cd, ls, mkdir, top, vi, gcc, etc
- Command Line Interface (CLI) (Shell)
  - ➢ bash, sh, etc
- Command line applications
  - ➢ pine, git, gdb, etc



Linux Architecture - Graphical User Interface (GUI)

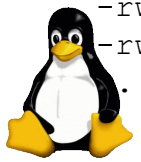
- Hardware
  - ➢ CPU, Memory, Disk, Graphics, Network, etc
- Core Kernel & Modules
  - ➢ Process, Memory, File, Network subsystems, Device drivers
- System calls
  - ➢ read, write, fork, exec, clone, etc
- System Libraries
  - ➢ libc, libpthread, etc
- Commands & System programs
  - ➢ cd, ls, mkdir, top, vi, gcc, etc
- Command Line Interface (CLI) (Shell)
  - ➢ bash, sh, etc
- Graphical User Interface (GUI)
  - ➢ X-Windows (Gnome, KDE, etc)
- Applications
  - ➢ Browser, eMail client, office suite, etc



Core Kernel

- Boot loader is the first program loaded by firmware (BIOS or UEFI)
- Core Kernel (in `/boot/vmlinuz-<version>`) is a program loaded by boot loader (grub)
- Kernel always runs in privileged mode in kernel space.
- To find kernel version use `uname -r`

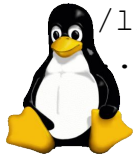
```
$ uname -r
4.4.0-31-generic

$ ls -lR /boot/grub
-r--r--r-- 1 root root     8432 Nov 14  2016 grub.cfg
drwxr-xr-x 2 root root    12288 Nov 14  2016 i386-pc
...

$ ls -lR /boot
-rw-r--r-- 1 root root   189558 Jul 13  2016 config-4.4.0-31-generic
-rw-r--r-- 1 root root 35907255 Nov 14  2016 initrd.img-4.4.0-31-generic
-rw------- 1 root root  3866473 Jul 13  2016 System.map-4.4.0-31-generic
-rw-r--r-- 1 root root  7047520 Nov 14  2016 vmlinuz-4.4.0-31-generic
...
```



Kernel Modules

- Loadable modules having device drivers loaded by hotplug of devices.
- Kernel modules also run in privileged mode in kernel space.
- To list loaded kernel modules, use `lsmod`
- To find all modules in /lib/modules/<kernel-version>/kernel/drivers/

```
$ lsmod
Module                  Size  Used by
...
drm                   401408  6 drm_kms_helper,i915
...

$ modinfo drm
filename:       /lib/modules/4.4.0-31-generic/kernel/drivers/gpu/drm/drm.ko
license:        GPL and additional rights
description:    DRM shared core routines
author:         ...

$ ls /lib/modules/`uname -r`/kernel/drivers/gpu/drm.ko
/lib/modules/4.4.0-31-generic/kernel/drivers/gpu/drm.ko
...
```



Shell

- First process after login
- Interprets and launches commands keyed-in at the command prompt.
- Commonly used shells are
  - bash
  - sh
- Types of shells
  - root vs non-root shell (# vs $ prompt)
  - login vs non-login shell (-bash vs bash)
- Interpreter for shell scripts

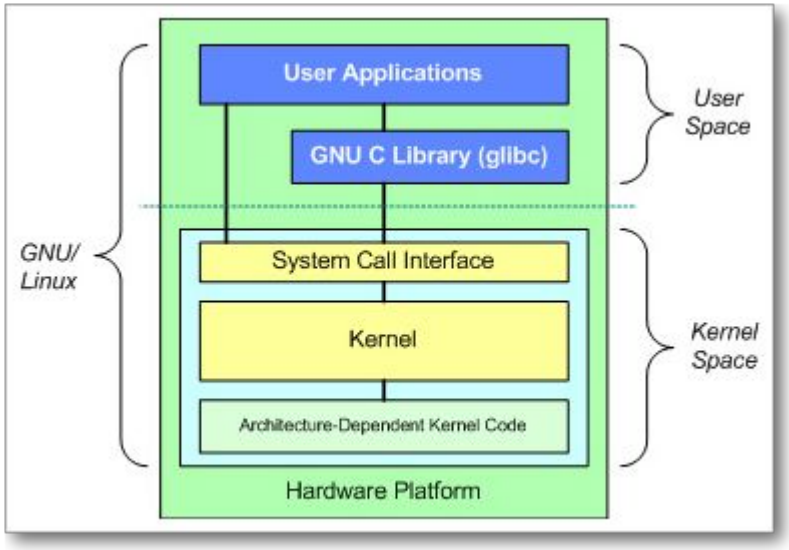
```
maruthisi@godavari:~$ _
```

```
root@godavari:~# _
```

System calls

- Entry points into the kernel.
- C language APIs.
- About 400 system calls
  - open(), read(), write(), close(), ioctl()
  - fork(), wait(), clone()
  - socket(), connect(), accept(), shutdown()
  - mmap(), munmap(), fadvise()
  - ...
- Using system calls in your program directly makes it
  - portable across Unices.
  - non-portable across Windows/Linux.

```
$ man syscalls

$ uname -o
GNU/Linux
```



Commands

- Programs that are keyed-in by user and launched by shell
- Types of commands
  - Internal (builtin) commands
  - External commands
- Internal commands
  - Implemented by the shell itself.

    echo, fg, bg, cd, ...
- External commands
  - Implemented by programs in /bin or /sbin directory.

    ls, mkdir, top, df, ...

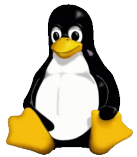
```
$ type cd
cd is a shell builtin

$ type mkdir
mkdir is /bin/mkdir
```

```
$ man ls
```

System Libraries

- Reusable routines packaged as `.a` or `.so`
- Every command loads its dependent libraries at launch time.
- Types of libraries
  - Archive libraries (.a)
  - Shared object (.so)
- Types of linking
  - Compile time (Static) (only with .a) - Deprecated.
  - Load time - (only with .so)
  - Run time (Dynamic) - (only with .so)
- To know the dependent libraries use `ldd path_to_program`
- Using standardized library function calls in your program makes it portable.

```
$ ldd /bin/ls
    linux-vdso.so.1 =>  (0x00007ffc1d7eb000)
    ...
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fce2111b000)
    libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007fce20a8a000)
```



References

- Linux manual pages
- www.wikipedia.org
- www.vger.kernel.org/vger-lists.html
- www.kernel.org
- www.gnu.org
- www.gnome.org, www.kde.org
- Courtesy Google images

Q & A