

A Taste of Gnu/Linux Usage and Installation

G. Sivakumar

Computer Science and Engineering
IIT Bombay
siva@iitb.ac.in

June 26, 2005

Outline

- GNU/Linux Development
- Unix Philosophy and Fundamentals
- Linux Quickstart (Bird's Eye View)



What is Linux?

- **Linux is a full-featured, 32-bit multi-user/multi-tasking OS.**
- Linux adheres to the common (POSIX) standards for UNIX .
- Native TCP/IP support.
- A mature X Windows GUI interface.
- Complete development environment. C, C++, Java, editors, version control systems.
- Open Source.

Pronounced as Lin-u~~ck~~s.



What is Linux?

- Linux is a full-featured, 32-bit multi-user/multi-tasking OS.
- Linux adheres to the common (POSIX) standards for UNIX .
- Native TCP/IP support.
- A mature X Windows GUI interface.
- Complete development environment. C, C++, Java, editors, version control systems.
- Open Source.

Pronounced as Lin-u-cks.



What is Linux?

- Linux is a full-featured, 32-bit multi-user/multi-tasking OS.
- Linux adheres to the common (POSIX) standards for UNIX .
- **Native TCP/IP support.**
- A mature X Windows GUI interface.
- Complete development environment. C, C++, Java, editors, version control systems.
- Open Source.

Pronounced as Lin-u-cks.



What is Linux?

- Linux is a full-featured, 32-bit multi-user/multi-tasking OS.
- Linux adheres to the common (POSIX) standards for UNIX .
- Native TCP/IP support.
- A mature X Windows GUI interface.
- Complete development environment. C, C++, Java, editors, version control systems.
- Open Source.

Pronounced as Lin-u-cks.



What is Linux?

- Linux is a full-featured, 32-bit multi-user/multi-tasking OS.
- Linux adheres to the common (POSIX) standards for UNIX .
- Native TCP/IP support.
- A mature X Windows GUI interface.
- Complete development environment. C, C++, Java, editors, version control systems.
- Open Source.

Pronounced as Lin-u-cks.



What is Linux?

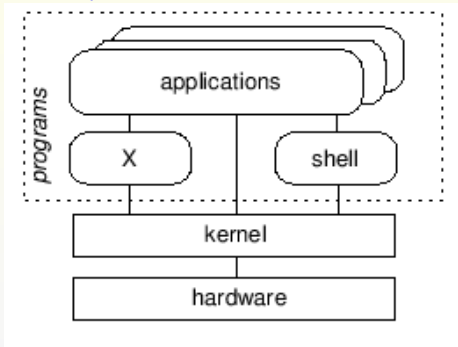
- Linux is a full-featured, 32-bit multi-user/multi-tasking OS.
- Linux adheres to the common (POSIX) standards for UNIX .
- Native TCP/IP support.
- A mature X Windows GUI interface.
- Complete development environment. C, C++, Java, editors, version control systems.
- **Open Source.**

Pronounced as Lin-u~~ck~~s.



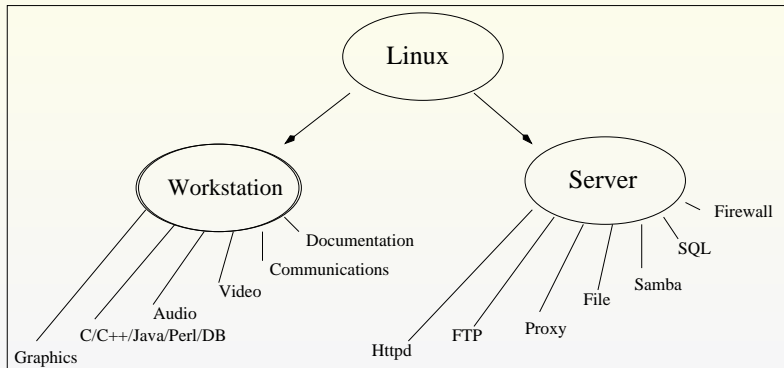
What is GNU/Linux?

Linux (**Linus Torvalds**) is only the **kernel**.



- The shell and the window environment are programs
- Programs access hardware via the kernel





Embedded Systems also!



The Linux Market

Summary of 2004 IDC Linux Market Study

In December of 2004, OSDL received the results of a global Linux market share and forecast study conducted by research firm IDC. This study predicts the overall market revenue for desktops, servers, and packaged software running on Linux will exceed \$35 billion by 2008. Other key findings were:

- Linux is no longer a niche phenomenon. The overall Linux marketplace revenues for server and PC hardware and packaged software on Linux is expected to reach \$35.7 billion by 2008.
- Packaged software is the fastest growing market segment within the Linux marketplace in terms of revenue, growing 44% annually to over \$14 billion in 2008.
- Free Linux deployments are attractive, but the reality is more commercial and government organizations will move toward paid, supported copies.
- Non-traditional server deployments with Linux as secondary or non-primary OS and redeployments



Who owns GNU/Linux?

- Analogy with Internet
- GNU/Linux Community (Richard Stallman, GPL, www.fsf.org)
- Cathedral vs. Bazaar Model (Eric Raymond)
- Open Software Development Labs (www.osdl.org)
 - Data Centre Linux (DCL)
 - Carrier Grade Linux (CGL)
 - Desktop Linux (DTL)
- Linux is taken seriously now!

First they will ignore you. Then they will ridicule you. Then they will fight you. Then you will win!

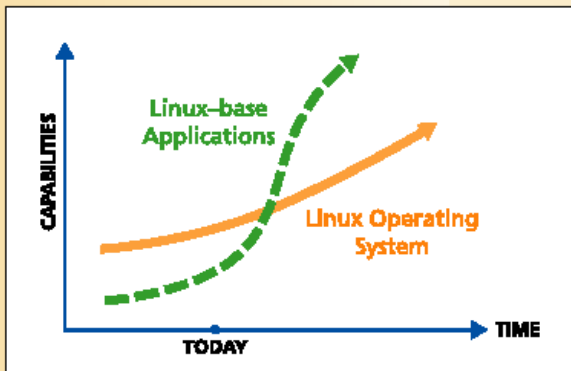


OSDL and the DTL Working Group

OSDL is a non-profit organization founded in 2000 with the principal objective of accelerating the growth and adoption of Linux in the enterprise. OSDL is sponsored and sustained by a range of IT vendor and communications industry vendors, along with a growing number of Linux end users. OSDL supports and drives the business-hardening of Linux and promotes the use of Linux and open source software around the world. The DTL working group was established in February, 2003, to enhance Linux for use on the enterprise desktop and promote its use.

community
ested part
solutions w
identified s

Currently,
Linux-based
application
capabilities
lag that of the
operating
system.



The Desktop Linux Working Group

The OSDL Desktop Linux Working Group is focused on accelerating the use of Linux on the enterprise desktop. The DTL initiatives' goal is to make the Linux Desktop in the enterprise commonplace. To achieve this goal, the DTL initiative has established three primary objectives:

- Identify a set of usage models which cover a large proportion of actual use within the enterprise; create a document specifying what capabilities are required of the Linux Desktop to successfully address each of those usage models.
- Identify mismatches and gaps between existing desktop implementations and the usage models, then work with the open source community to address those problems.
- Identify and remove barriers that ISVs face in porting to and developing on Linux, enabling a more robust market for Linux-based desktop applications.

Linux is already a technically capable operating system environment today. Like any OS, feature/function enhancements occur regularly as users ask for them and developers incorporate them. However to achieve more widespread adoption and reach its true potential for desktop productivity, Linux must attract a critical mass of client application developers. Therefore, the first two objectives above lead directly to the third. Without an



OSDL Engineering Project Participation

In 2004, the OSDL itself expended considerable resources in participating in and contributing to a wide range of open source development projects. To give credit where it is exceptionally due, following is a list of those technical areas, projects and direct contributions by OSDL engineering staffers:

Clustering (CGL/DCL)

- Mark Haverkamp (TIPC, Event Service)
- Daniel McNeil (foundational services - RHAT/Sistina)

Fast Reboot (CGL/DCL)

- Randy Dunlap (kexec)

Multipath IO (CGL/DCL)

- Dave Olein (dm-based multipath IO)

Security (CGL/DCL)

- Chris Wright (Security SIG, profiles, LSM, DigSig, virtualization projects)

OSDL Kernel Conduit (DCL)

- Leann Ogasawara (AIO patches, openais?)

Async disk IO (DCL)

- Daniel McNeil (patch set, performance)
- Judith Lebzelter (AIO stress test, performance)

Network performance (DCL)

- Steve Hemminger (bridge, TCP, NFS, IP route tools, etc.)

Logical Volume Management (DCL)

- Dave Olein (gap analysis)
- Mark Wong (performance testing)

Persistent Device Naming with UDEV (DCL)

- Mark Haverkamp (testing)
- Mary Meredith (documentation)
- Leann Ogasawara (kernel janitor enabled drivers to support sysfs required for udev)

Large Numbers of Disks - 4096 (DCL)

- Dave Olein (integration testing)

Stability/Maintenance (CGL/DCL community)

- Randy Dunlap and Leanne Ogasawara (janitor, bug fixes, etc.)
- Cliff White (STP/PLM for Linus/Andrewtrees)
- Judith Lebzelter (PLM cross compile tests for Linus/Andrewtrees)
- Leann Ogasawara (Kernel Janitors)

Database test (DCL)

- Mark Wong (PostgreSQL performance, features testing)

File system and IO scheduler test (global)

- Judith Lebzelter (iozone/tiobench perf.)
- Mary Meredith and Mark Wong (database file I/O, raw I/O performance)
- Dave Olein (I/O scheduler analysis)

Test Development Framework STP/PLM (global)

- Bryce Harrington (STP project leader, maintenance)
- Cliff White (STP maintenance)
- Judith Lebzelter (PLM project leader, maintenance)

Tinderbox Project (DTL community)

- Bryce Harrington (x.org client, samba client)
- Cliff White (kernel client and toolkit)



- **Multi-user**
 - A user needs an account to use a computer
 - Each user must log in
 - Complete separation of different users' files and configuration settings
- **Small components**
 - Each component should perform a single task
 - Multiple components can be combined and chained together for more complex tasks
 - An individual component can be substituted for another, without affecting other components



- Multi-user
 - A user needs an account to use a computer
 - Each user must log in
 - Complete separation of different users' files and configuration settings
- Small components
 - Each component should perform a single task
 - Multiple components can be combined and chained together for more complex tasks
 - An individual component can be substituted for another, without affecting other components



- Multi-user
 - A user needs an account to use a computer
 - Each user must log in
 - Complete separation of different users' files and configuration settings
- Small components
 - Each component should perform a single task
 - Multiple components can be combined and chained together for more complex tasks
 - An individual component can be substituted for another, without affecting other components



- Multi-user
 - A user needs an account to use a computer
 - Each user must log in
 - Complete separation of different users' files and configuration settings
- Small components
 - Each component should perform a single task
 - Multiple components can be combined and chained together for more complex tasks
 - An individual component can be substituted for another, without affecting other components



- Multi-user
 - A user needs an account to use a computer
 - Each user must log in
 - Complete separation of different users' files and configuration settings
- Small components
 - Each component should perform a single task
 - Multiple components can be combined and chained together for more complex tasks
 - An individual component can be substituted for another, without affecting other components



- Multi-user
 - A user needs an account to use a computer
 - Each user must log in
 - Complete separation of different users' files and configuration settings
- Small components
 - Each component should perform a single task
 - Multiple components can be combined and chained together for more complex tasks
 - An individual component can be substituted for another, without affecting other components



- Multi-user
 - A user needs an account to use a computer
 - Each user must log in
 - Complete separation of different users' files and configuration settings
- Small components
 - Each component should perform a single task
 - Multiple components can be combined and chained together for more complex tasks
 - An individual component can be substituted for another, without affecting other components



- Multi-user
 - A user needs an account to use a computer
 - Each user must log in
 - Complete separation of different users' files and configuration settings
- Small components
 - Each component should perform a single task
 - Multiple components can be combined and chained together for more complex tasks
 - An individual component can be substituted for another, without affecting other components



Using a Linux System

- Login prompt displayed
 - When Linux first loads after booting the computer
 - After another user has logged out
- Need to enter a username and password
- The login prompt may be graphical or simple text
- If text, logging in will present a shell
- If graphical, logging in will present a desktop



- A shell provides an interface between the user and the operating system kernel
- Either a command interpreter or a graphical user interface
- Traditional Unix shells are command-line interfaces (CLIs)
- Usually started automatically when you log in or open a terminal
- Linux's most popular command interpreter is called **bash**
 - The Bourne-Again Shell
 - More sophisticated than the original sh by Steve Bourne
 - Can be run as sh, as a replacement for the original Unix shell
- Gives you a prompt and waits for a command to be entered



Try the following..

- Shell (ls, cat, grep, wc, cut, seq, man ...)
- Files (df, /etc/fstab, du, permissions, ...)
- Processes (ps, top, pstree, kill, ...)
- System Files (/etc/passwd, /proc/cpuinfo, lspci, lsub, ...)
- Demo of various tools
 - Editors (Emacs, vim, gedit, ...)
 - Compilers (gcc, perl, ruby, python, java, ...)
 - Multimedia tools (xmms, realplay, mplayer, ...)
 - Games (xboard, gnuchess, ...)
 - Office/productivity- OpenOffice, evolution, firefox
 - Word processing, presentations (latex, beamer, ...)
 - Many others ...



Becoming a Power User of Linux

- Shell Commands & Utilities (Basic and advanced, online docs)
- Shell Programming (pipes and filters to combine tasks)
- File Systems (organization, access control)
- Processes (starting, controlling)
- Networking (TCP/IP, LAN, WAN)
- System Administration
 - System and User Management (cron, log files, quotas, ...)
 - Optimizing System Performance
 - Securing the System
- ...



Linux Distributions

Many different (packaging) Linux distributions

- Operating system kernel (2.6.11 is latest)
- An installation program
- Various applications.

<http://www.distrowatch.com/>

Commercial Distributions

- Red Hat, Mandrake and Suse.1 ...
- Does not restrict your freedoms though!
- You pay for the packaging and support.

Completely free distributions (download from www.linuxiso.org)

- Fedora Core
- Debian
- Slackware.



Entire Linux package is on a *single bootable CD-ROM*.

Advantages

- No disk partitioning.
- Does not modify any files on disk
- Can try everything without any risk!

Disadvantage

- Configurations not saved to disk.
- Applications can't write to NTFS partitions!
- Not useful for long term!

First was **Knoppix** (www.knopper.net). Many now (Ubuntu, Mepis, Slax, ...)



Most *complex* task in a Linux Installation

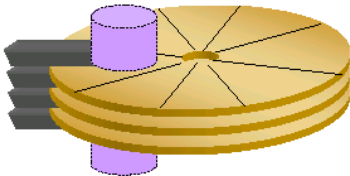
Primary Partitions (max 4) and Secondary (Logical) Partitions (3 primary + 1 extended)

Disk Naming Conventions

- Floppy /dev/fd0
- Hard Disk
 - IDE (hda1, hda2, ...)
 - SCSI (sda1, sda2, ...)



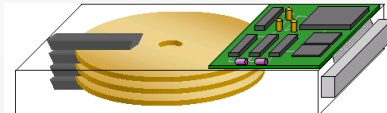
Physical Disks



- Head
- Cylinder
- Sector

Data Transfer speed depends on

- Speed of I/O technology (SCSI, Fibre Channel, ...)
- Seek time (time for heads to move to a new track)
- Latency (time for sector to rotate under heads)
- Speed and size of disk drive buffer memory



Typical Partitioning Schemes

Minimum Partition Structure

/ (filesystem root)

swap

Typical Multiple Partition Structure

/ (filesystem root)

/var

/usr

/tmp

/home

swap

What is *swap*? (Typically: 2 times RAM size)



Why Multiple Partitions

- Any damage is limited to a single partition, which can be reformatted without losing the data on other partitions.
- Files like mail and log files, which enlarge dynamically, can be prevented from using up too much disk space.
- It is easier to upgrade system files without overwriting other data.
- Backups are easier.
- The time required to check the filesystem at boot is reduced.
- Can have *multiple OS-es* on 1 disk.



fdisk is one of the popular tools.

```
root@slap:~ # fdisk /dev/hda
```

```
The number of cylinders for this disk is set to 5168.  
There is nothing wrong with that, but this is larger than 1024,  
and could in certain setups cause problems with:  
1) software that runs at boot time (e.g., old versions of LILO)  
2) booting and partitioning software from other OSs  
   (e.g., DOS FDISK, OS/2 FDISK)
```

```
Command (m for help): p
```

```
Disk /dev/hda: 40.0 GB, 40007761920 bytes  
240 heads, 63 sectors/track, 5168 cylinders  
Units = cylinders of 15120 * 512 = 7741440 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	813	6146248+	7	HPFS/NTFS
/dev/hda2		814	883	529200	82	Linux swap / Solaris
/dev/hda3	*	884	1716	6297480	83	Linux
/dev/hda4		1717	5168	26097120	83	Linux



Using fdisk

```
Command (m for help): m
Command action
  a   toggle a bootable flag
  b   edit bsd disklabel
  c   toggle the dos compatibility flag
  d   delete a partition
  l   list known partition types
  m   print this menu
  n   add a new partition
  o   create a new empty DOS partition table
  p   print the partition table
  q   quit without saving changes
  s   create a new empty Sun disklabel
  t   change a partition's system id
  u   change display/entry units
  v   verify the partition table
  w   write table to disk and exit
  x   extra functionality (experts only)
```



Using fdisk

Command (m for help): l

0	Empty	1e	Hidden W95 FAT1	80	Old Minix	be	Solaris boot
1	FAT12	24	NEC DOS	81	Minix / old Lin	bf	Solaris
2	XENIX root	39	Plan 9	82	Linux swap / So	c1	DRDOS/sec (FAT-
3	XENIX usr	3c	PartitionMagic	83	Linux	c4	DRDOS/sec (FAT-
4	FAT16 <32M	40	Venix 80286	84	OS/2 hidden C:	c6	DRDOS/sec (FAT-
5	Extended	41	PPC PReP Boot	85	Linux extended	c7	Syrinx
6	FAT16	42	SFS	86	NTFS volume set	da	Non-FS data
7	HPFS/NTFS	4d	QNX4.x	87	NTFS volume set	db	CP/M / CTOS / .
8	AIX	4e	QNX4.x 2nd part	88	Linux plaintext	de	Dell Utility
9	AIX bootable	4f	QNX4.x 3rd part	8e	Linux LVM	df	BootIt
a	OS/2 Boot Manag	50	OnTrack DM	93	Amoeba	e1	DOS access
b	W95 FAT32	51	OnTrack DM6 Aux	94	Amoeba BBT	e3	DOS R/O
c	W95 FAT32 (LBA)	52	CP/M	9f	BSD/OS	e4	SpeedStor
e	W95 FAT16 (LBA)	53	OnTrack DM6 Aux	a0	IBM Thinkpad hi	eb	BeOS fs
f	W95 Ext'd (LBA)	54	OnTrackDM6	a5	FreeBSD	ee	EFI gpt
10	OPUS	55	EZ-Drive	a6	OpenBSD	ef	EFI (FAT-12/16/
11	Hidden FAT12	56	Golden Bow	a7	NeXTSTEP	f0	Linux/PA-RISC b
12	Compaq diagnost	5c	Priam Edisk	a8	Darwin UFS	f1	SpeedStor
14	Hidden FAT16 <3	61	SpeedStor	a9	NetBSD	f4	SpeedStor
16	Hidden FAT16	63	GNU HURD or Sys	ab	Darwin boot	f2	DOS secondary
17	Hidden HPFS/NTF	64	Novell Netware	b7	BSDI fs	fd	Linux raid auto
18	AST SmartSleep	65	Novell Netware	b8	BSDI swap	fe	LANstep
1b	Hidden W95 FAT3	70	DiskSecure Mult	bb	Boot Wizard hid	ff	BBT
1c	Hidden W95 FAT3	75	PC/IX				



<i>fdisk interactive command</i>	<i>Explanation and Example</i>																																																	
n	<p>Create new partition</p> <p>Command (m for help): n Command action l logical (5 or over) p primary partition (1-4)</p> <p>1 First cylinder (952-1024, default 952): 952 Last cylinder or +size or +sizeM or +sizeK (952-1015, default 1015): 1015</p> <p>Command (m for help):</p> <p>Note: you will need to specify the starting and either the ending cylinder, or the partition size e.g. +80M</p>																																																	
p	<p>Display Partition Table</p> <p>Command (m for help): p</p> <p>Disk /dev/hda: 255 heads, 63 sectors, 1027 cylinders Units = cylinders of 16065 * 512 bytes</p> <table border="1"><thead><tr><th>Device</th><th>Boot</th><th>Start</th><th>End</th><th>Blocks</th><th>Id</th><th>System</th></tr></thead><tbody><tr><td>/dev/hda1</td><td>*</td><td>1</td><td>261</td><td>2096451</td><td>b</td><td>Win95 FAT32</td></tr><tr><td>/dev/hda2</td><td></td><td>262</td><td>1024</td><td>6128797+</td><td>5</td><td>Extended</td></tr><tr><td>/dev/hda5</td><td></td><td>952</td><td>1015</td><td>514048+</td><td>6</td><td>FAT16</td></tr><tr><td>/dev/hda6</td><td></td><td>1016</td><td>1024</td><td>72261</td><td>82</td><td>Linux swap</td></tr><tr><td>/dev/hda7</td><td></td><td>262</td><td>516</td><td>2048224+</td><td>83</td><td>Linux</td></tr><tr><td>/dev/hda8</td><td></td><td>517</td><td>951</td><td>3494106</td><td>83</td><td>Linux</td></tr></tbody></table>	Device	Boot	Start	End	Blocks	Id	System	/dev/hda1	*	1	261	2096451	b	Win95 FAT32	/dev/hda2		262	1024	6128797+	5	Extended	/dev/hda5		952	1015	514048+	6	FAT16	/dev/hda6		1016	1024	72261	82	Linux swap	/dev/hda7		262	516	2048224+	83	Linux	/dev/hda8		517	951	3494106	83	Linux
Device	Boot	Start	End	Blocks	Id	System																																												
/dev/hda1	*	1	261	2096451	b	Win95 FAT32																																												
/dev/hda2		262	1024	6128797+	5	Extended																																												
/dev/hda5		952	1015	514048+	6	FAT16																																												
/dev/hda6		1016	1024	72261	82	Linux swap																																												
/dev/hda7		262	516	2048224+	83	Linux																																												
/dev/hda8		517	951	3494106	83	Linux																																												



<i>fdisk interactive command</i>	<i>Explanation and Example</i>
t	<p>Change Partition Type</p> <p>Command (m for help): t</p> <p>Partition number (1-8): 6</p> <p>Hex code (type L to list codes): 82</p> <p>Changed system type of partition 6 to 82 (Linux swap)</p> <p>Note: By default fdisk creates Linux native partitions. You will need to use the t command to turn a native partition into a swap partition. A list of partition types can be obtained using the l interactive command</p>
w	<p>Write changes to Partition Table - EXTREME CARE!</p> <p>Command (m for help): w</p> <p>The partition table has been altered!</p> <p>Calling ioctl() to re-read partition table. Syncing disks.</p> <p>[root@ext7144 /root]#</p>



Completing the Installation

After *partitioning* the main steps are

- Choosing Packages to install
- Setting root (and some user) passwords
- Setting up the Network
- Installing LILO or GRUB (multiple OS-es)
- Configuring and customizing your system

This will complete the installation (your first lab)

