LINUX ADMINISTARTION

UNIT – 1 & 2

- **Installing Linux as a Server:** Linux distributions, open source software and GNU, difference between windows and linux, installing linux in a server configuration, GNOME and KDE- X windows system, managing software.
- **Linux Administration:** Managing users, user text files-user management tools, command line, boot loaders, file systems, compiling linux kernel, linux security.
- **Internet Services:** DNS, FTP-Mechanics-installing and customizing the server, setting up web server using apache, SMTP- install, configure and run postfix server, POP and IMAP, SSH- public key cryptography, creating a secure tunnel.

UNIT 3 & 4

- **Intranet Services:** NFS- enable and configure NFS server and client, NIS- configuring master and secondary NIS server and client, NIS tools, SAMBA-administration, printing-install cups-add and manage print jobs, DHCP, virtualization.
- **Linux Process Control:** Linux process environment, login processes, parent child relationship, process variable, process monitoring, invoking foreground and background processes, terminating process, daemons.
- **Shell Programming:** Introduction, shell scripts, executing shell scripts, creating scripts, simple examples.

List of Practicals

- 1. Installation Linux operating system.
- 2. To study basic Linux Commands.
- 3. To study and create various types of files in linux.
- 4. To study vi and vim editors
- 5. To study user, group, owner and access permissions of a file.
- 6. Study of Bash shell, Bourne shell and C shell in linux operating system
- 7. Study Shell scripting in Linux.
- 8. To study various filters in Linux.
- 9. Administration of LINUX Operating System.
- 10. Introduction to variables in shell scripting.
- 11. Introduction of various constructs in shell scripting.
- 12. Write the program to mount the various devices (i.e. floppy, CD-Rom etc).
- 13. To study Process synchronization.

Books and Web Links

- **Textbooks:**
- 1. Wale Soyinka, **—*Linux Administration A Beginners Guide*ll,** Tata McGraw Hill.
- 2. Mc Kinnon, **—*Installing and Administrating Linux*ll,** Wiley.
- **Reference Books:**
- 1. Richard Peterson, **—*Linux: The complete Reference*ll,** Tata McGraw Hill.
- 2. Mark G. Sobell, **—*Practical Guide to Fedora and Red Hat Enterprise Linux*ll,** Prentice Hall.
- 3. www.linuxhomenetworking.com
- www.linux.org www.linux.com
- https://www.youtube.com/watch?v=Wgi-OfbP2Gw

LINUX INSTALLATION

Download LINUX

- To install Red Hat, you will need to download the ISO images (CD Images) of the installation CD-ROMs from http://fedora.redhat.com
- Download the i386 images for 32 Intel Processors, PPC images for for Apple Macintosh and x86_64 for for 64 bit AMD Processors
- Burn the iso CD images on CDs and use these CDs as Installation CDs (typically 4)

redhat.

Red Hat Enterprise Linux

- To install or upgrade in graphical mode, press the <ENTER> key.

- To install or upgrade in text mode, type: linux text <ENTER>.

- Use the function keys listed below for more information.

[F1-Main] [F2-Options] [F3-General] [F4-Kernel] [F5-Rescue]
boot: _


Courtesy of OSDir.com
Running anaconda, the Red Hat Enterprise Linux system installer - please wait...

redhat.

Welcome to Red Hat Enterprise Linux ES

During this installation, you can use your mouse or keyboard to navigate through the various screens.

The **Tab** key allows you to move around the screen, the Up and Down arrow keys to scroll through lists, + and - keys expand and collapse lists, while **Space** and **Enter** selects or removes from selection a highlighted item. You can also use the **Alt-*X*** key command combination as a way of clicking on buttons or making other screen selections, where ***X*** is replaced with any

Hide Help | Release Notes | Back | Next

redhat.

Language Selection

Choose the language you would like to use during this installation.

What language would you like to use during the installation process?

- Bulgarian (Български)
- Catalan (Català)
- Chinese(Simplified) (简体中文)
- Chinese(Traditional) (繁體中文)
- Croatian (Hrvatski)
- Czech (Čeština)
- Danish (Dansk)
- Dutch (Nederlands)
- English (English)
- Estonian (eesti keel)
- Finnish (suomi)
- French (Français)
- German (Deutsch)
- Gujarati (ગુજરાતી)
- Hindi (हिन्दी)
- Hungarian (magyar)
- Icelandic (Íslenska)

Hide Help | Release Notes | Back | Next

redhat.

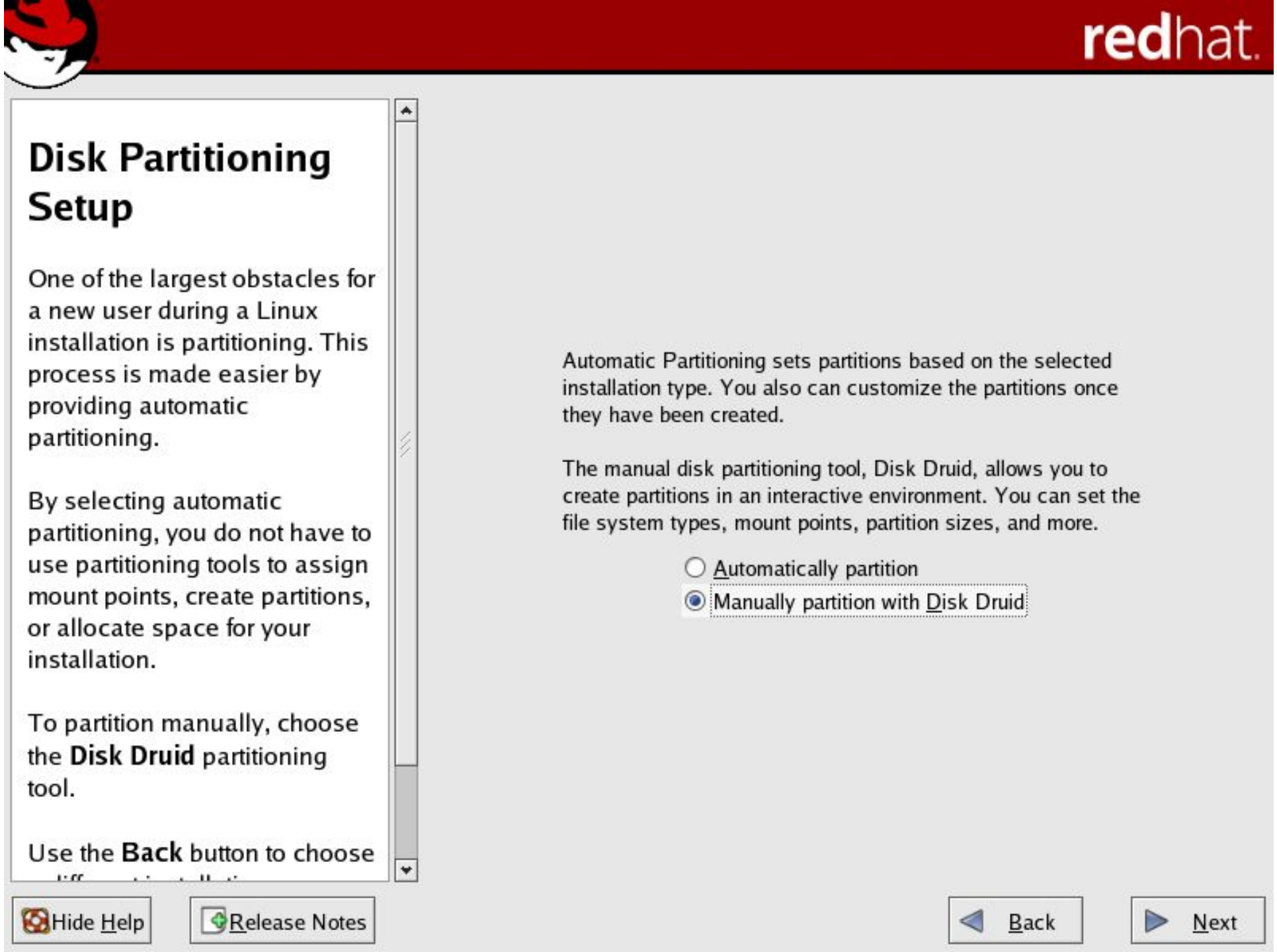
Keyboard Configuration

Choose the layout type for the keyboard (for example, U.S. English) that you would like to use for the system.

Select the appropriate keyboard for the system.

- Russian (Microsoft)
- Russian (ru1)
- Russian (ru2)
- Russian (win)
- Slovakian
- Slovenian
- Spanish
- Swedish
- Swiss French
- Swiss French (latin1)
- Swiss German
- Swiss German (latin1)
- Tamil (Inscript)
- Tamil (Typewriter)
- Turkish
- Ukrainian
- United Kingdom
- **U.S. English**
- U.S. International

Hide Help | Release Notes | Back | Next


redhat.
Disk Partitioning Setup
One of the largest obstacles for a new user during a Linux installation is partitioning. This process is made easier by providing automatic partitioning.
By selecting automatic partitioning, you do not have to use partitioning tools to assign mount points, create partitions, or allocate space for your installation.
To partition manually, choose the Disk Druid partitioning tool.
Use the Back button to choose
Automatic Partitioning sets partitions based on the selected installation type. You also can customize the partitions once they have been created.
The manual disk partitioning tool, Disk Druid, allows you to create partitions in an interactive environment. You can set the file system types, mount points, partition sizes, and more.
Automatically partition
Manually partition with Disk Druid
Hide Help
Release Notes
Back
Next

What is a Partition?

- Partitioning is a means to divide a single hard drive into many logical drives.
- A partition is a contiguous set of blocks on a drive that are treated as an independent disk.
- A partition table is an index that relates sections of the hard drive to partitions.

Why have multiple partitions?

- Reduce the risk of system failure in case a partition becomes full. Runaway processes or maniacal users can consume so much disk space that the operating system no longer has room on the hard drive for its bookkeeping operations. This will lead to disaster. By segregating space, you ensure that things other than the operating system die when allocated disk space is exhausted.

- Encapsulate your data. Since file system corruption is local to a partition, you stand to lose only some of your data if an accident occurs.

Partition Fields

- **Device:** This field displays the partition's device name.
- **Start:** This field shows the sector on your hard drive where the partition begins.
- **End:** This field shows the sector on your hard drive where the partition ends.
- **Size:** This field shows the partition's size (in MB).
- **Type:** This field shows the partition's type (for example, ext2, ext3, or vfat).
- **Mount Point:** A mount point is the location within the directory hierarchy at which a volume exists; the volume is "mounted" at this location. This field indicates where the partition will be mounted.

Filesystem Types

- **ext2 —** An ext2 filesystem supports standard Unix file types (regular files, directories, symbolic links, etc). It provides the ability to assign long file names, up to 255 characters. Versions prior to Red Hat Linux 7.2 used ext2 filesystems by default.
- **ext3 —** The ext3 filesystem is based on the ext2 filesystem and has one main advantage — journaling. Using a journaling filesystem reduces time spent recovering a filesystem after a crash as there is no need to fsck the filesystem.
- **swap —** Swap partitions are used to support virtual memory. In other words, data is written to a swap partition when there is not enough RAM to store the data your system is processing.
- **vfat —** The VFAT filesystem is a Linux filesystem that is compatible with Windows 95/NT long filenames on the FAT filesystem.

Recommended Partitioning Scheme

Unless you have a reason for doing otherwise, it is recommended that you create the following partitions:

- /boot partition – contains kernel images and grub configuration and commands
- / partition
- /var partition
- /home partition
- Any other partition based on application (e.g /usr/local for squid)
- swap partition — swap partitions are used to support virtual memory. In other words, data is written to a swap partition when there is not enough RAM to store the data your system is processing. The size of your swap partition should be equal to twice your computer's RAM.

Disk Partition

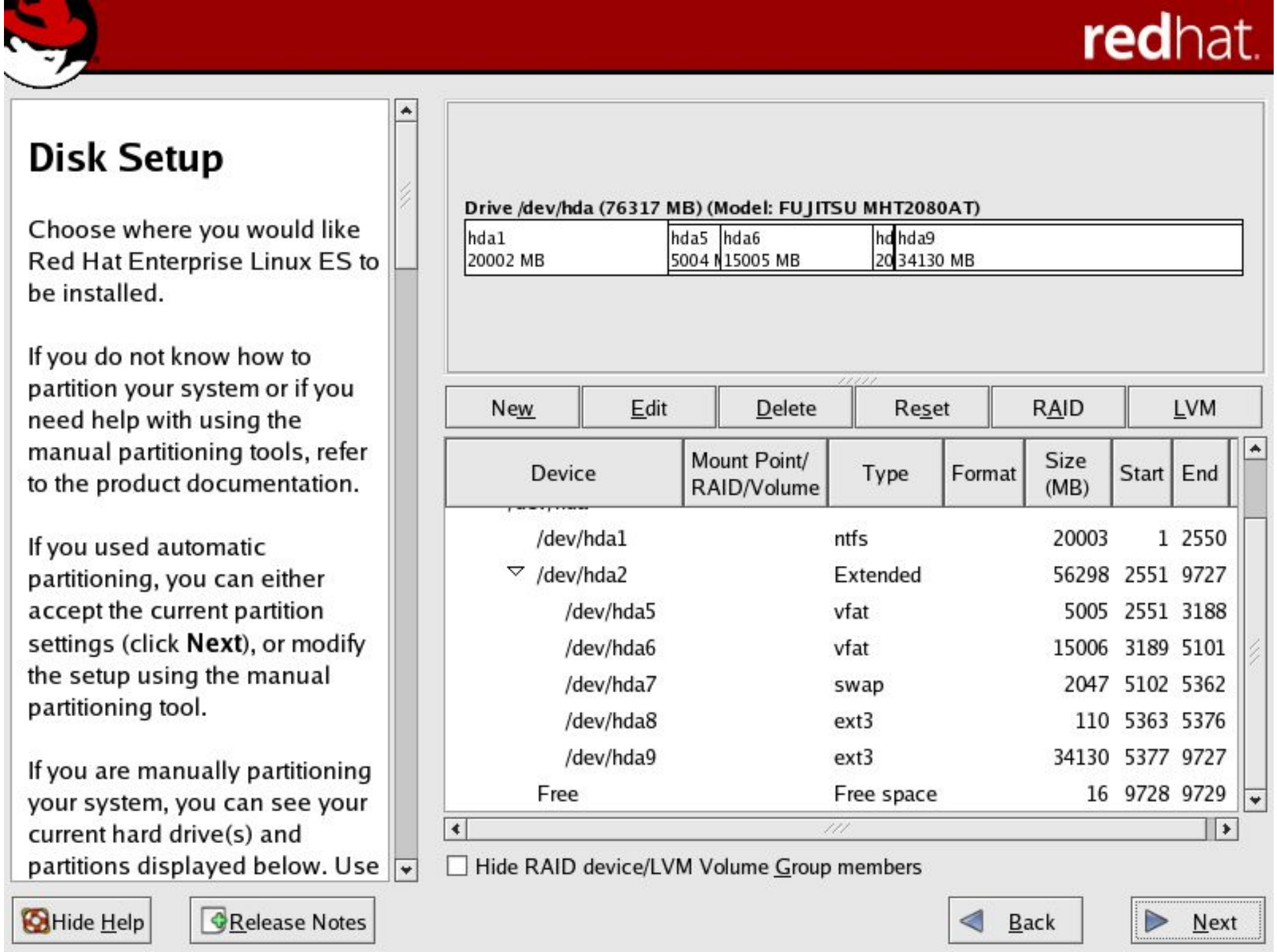
- **IDE Disk Partitions**
  - /dev/hda (Primary Master Disk)
    - /dev/hda1 (First Primary Partition)
    - /dev/hda2 (Second Primary Partition)
  - /dev/hdb (Primary Slave Partition)
    - /dev/hdb1
  - /dev/hdc (Secondary Master/Slave Partition)
    - /dev/hdc1
- **SCSI Disk Partitions**
  - /dev/sda1, /dev/sda2
  - /dev/sdb1, /dev/sdb2
  - /dev/sdc1, /dev/sdc2

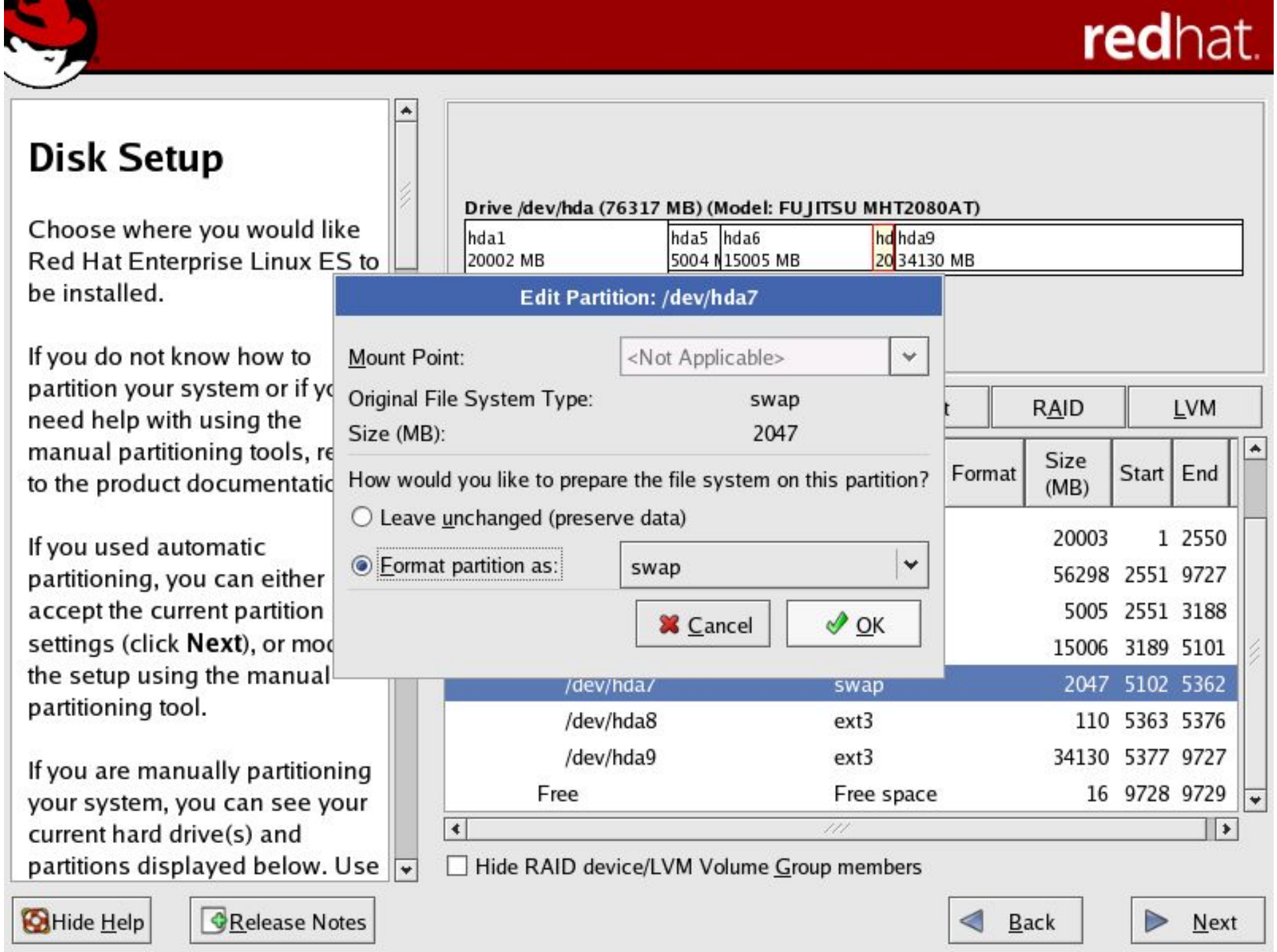
Software RAID and LVM

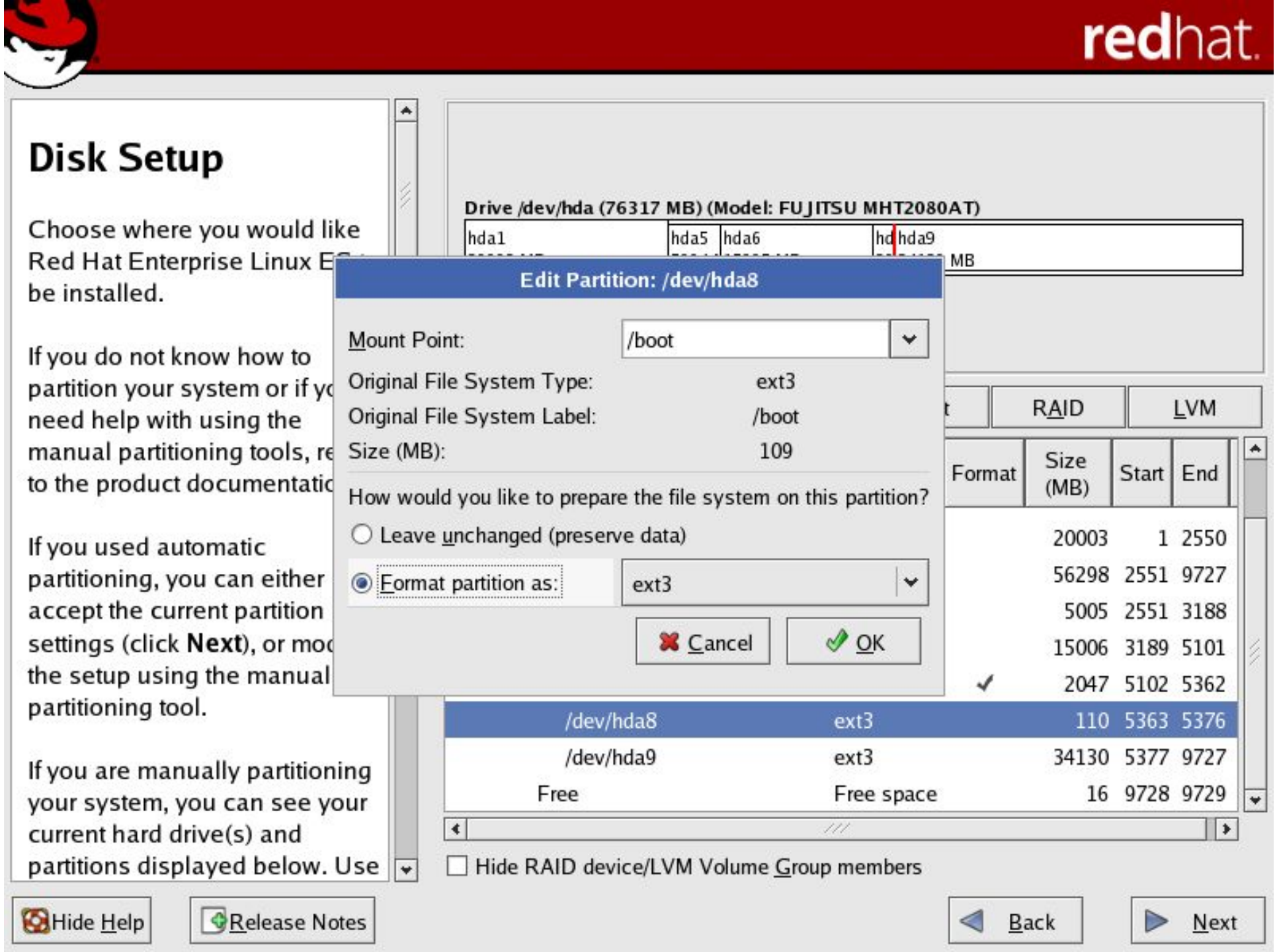
- **Software RAID (Redundant Array of Inexpensive Disk)**
  - RAID 0 (Striping)
  - RAID 1 (Mirroring)
  - RAID 5 (Striping with Parity)
- **LVM (Logical Volume Manager)**

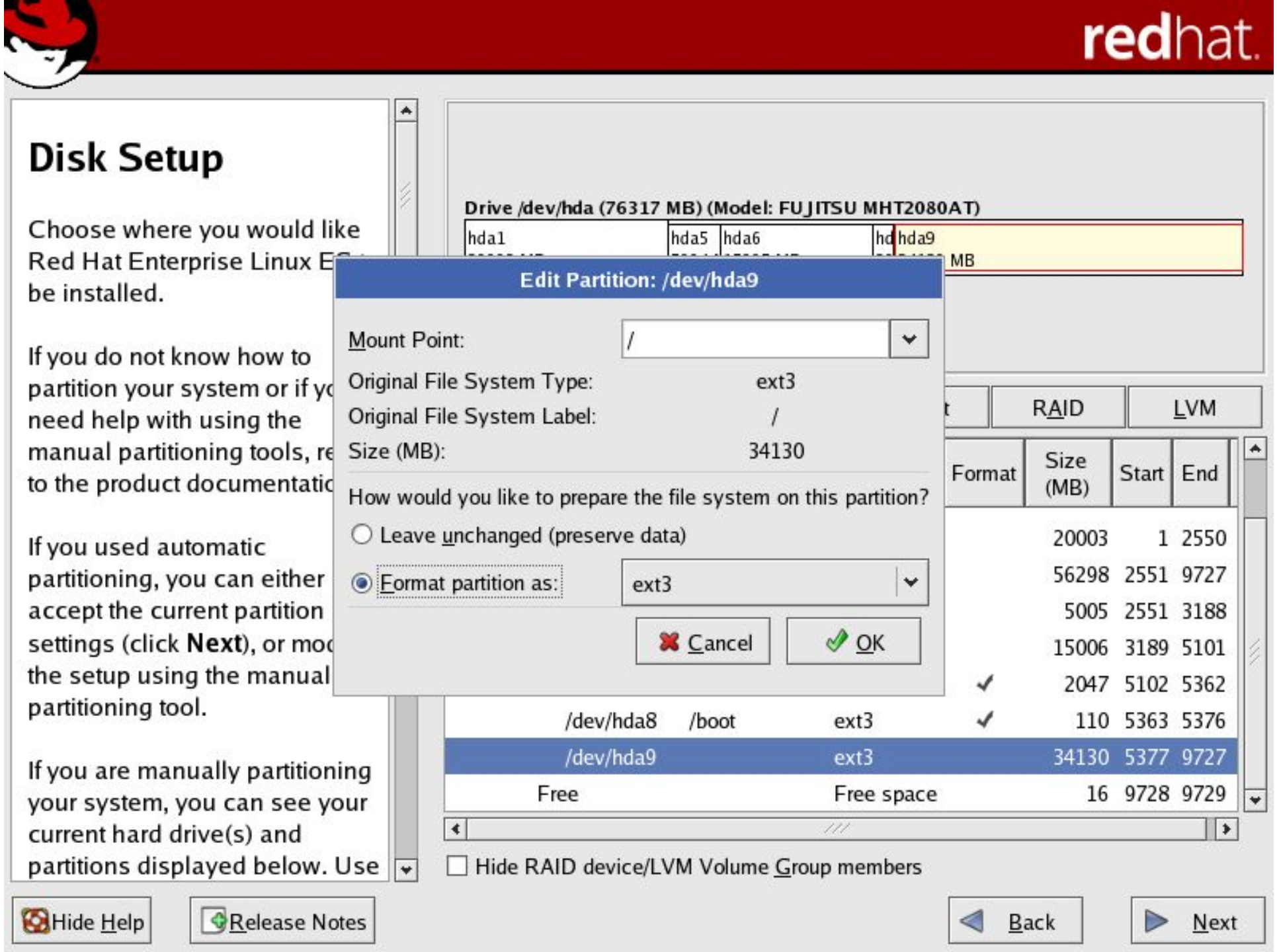
Boot Loader

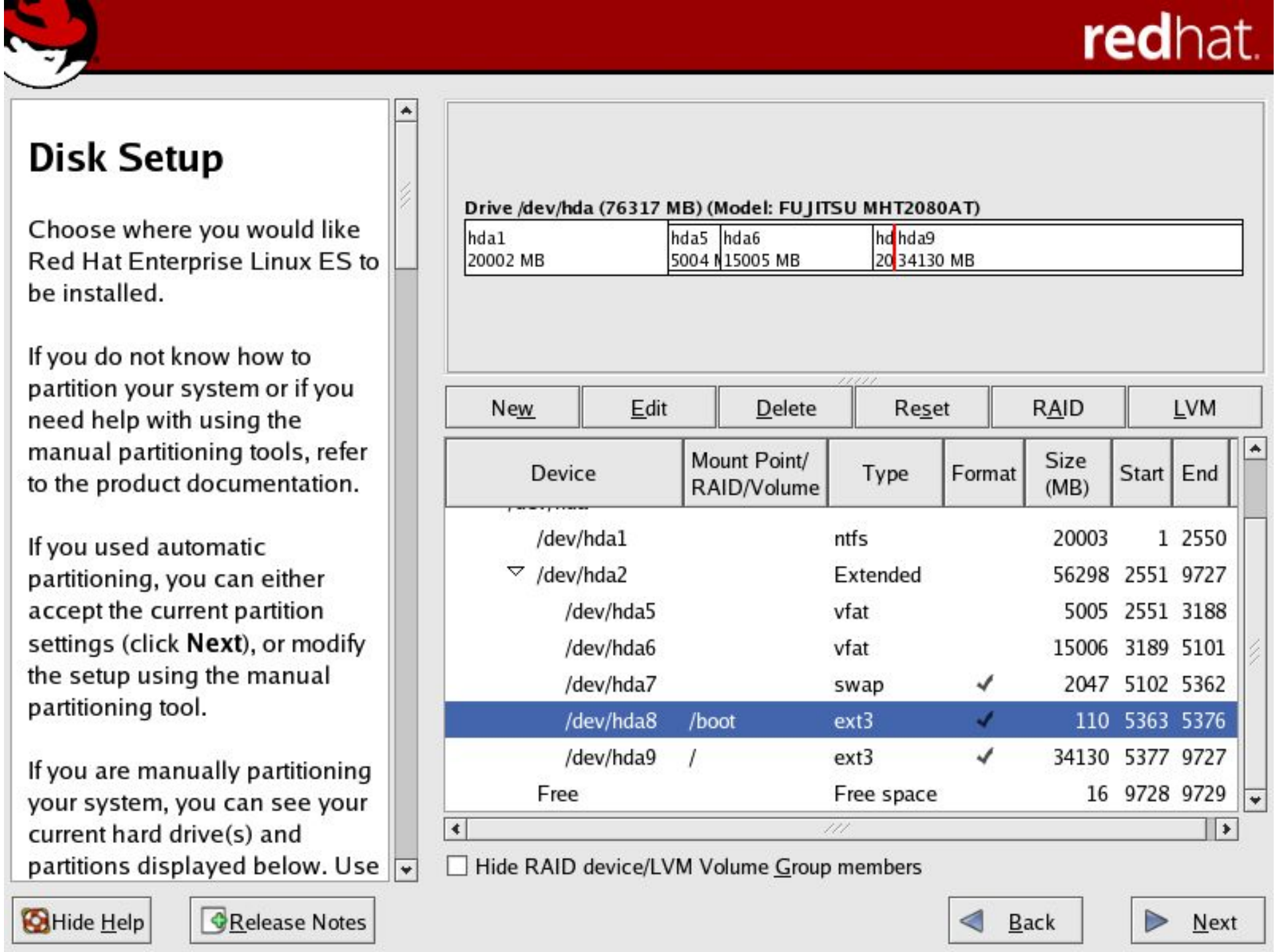
- In order for the BIOS to load an OS it looks for instructions on the first sector of a hard drive.
- On the first sector of the hard drive resides the master boot record (MBR), and is where a boot loader is initialized.
- Depending on the boot loader, additional files may be stored and read from a partition on the hard drive.
- After this step the boot loader begins to start the operating system, and is not used again until the next boot.

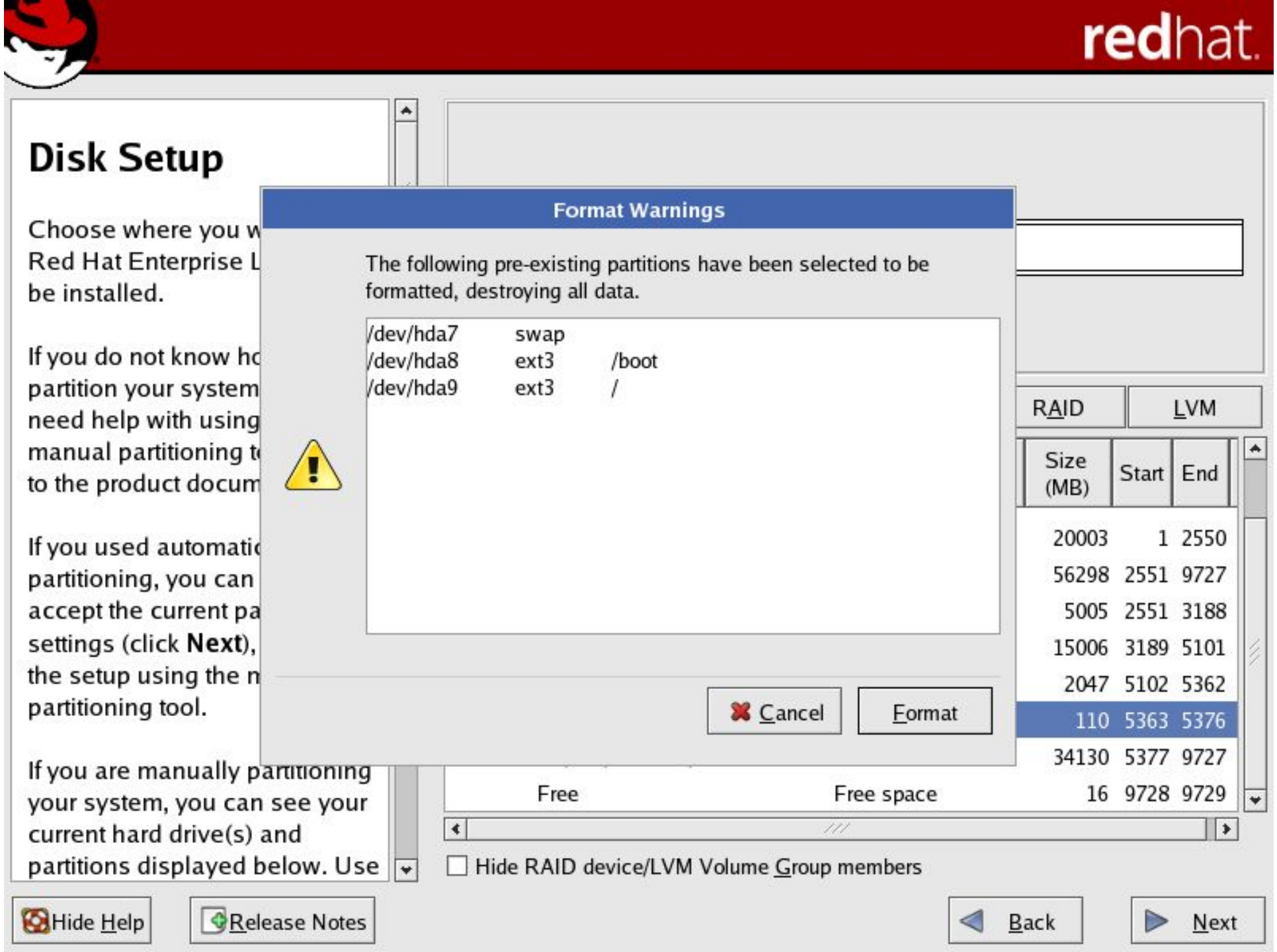

redhat.
Disk Setup
Choose where you would like Red Hat Enterprise Linux ES to be installed.
If you do not know how to partition your system or if you need help with using the manual partitioning tools, refer to the product documentation.
If you used automatic partitioning, you can either accept the current partition settings (click Next), or modify the setup using the manual partitioning tool.
If you are manually partitioning your system, you can see your current hard drive(s) and partitions displayed below. Use
Drive /dev/hda (76317 MB) (Model: FUJITSU MHT2080AT)
hda1 20002 MB
hda5 5004 M
hda6 15005 MB
hd hda9 20 34130 MB
New
Edit
Delete
Reset
RAID
LVM
Device
Mount Point/ RAID/Volume
Type
Format
Size (MB)
Start
End
/dev/hda1 ntfs 20003 1 2550
/dev/hda2 Extended 56298 2551 9727
/dev/hda5 vfat 5005 2551 3188
/dev/hda6 vfat 15006 3189 5101
/dev/hda7 swap 2047 5102 5362
/dev/hda8 ext3 110 5363 5376
/dev/hda9 ext3 34130 5377 9727
Free Free space 16 9728 9729
Hide RAID device/LVM Volume Group members
Hide Help
Release Notes
Back
Next

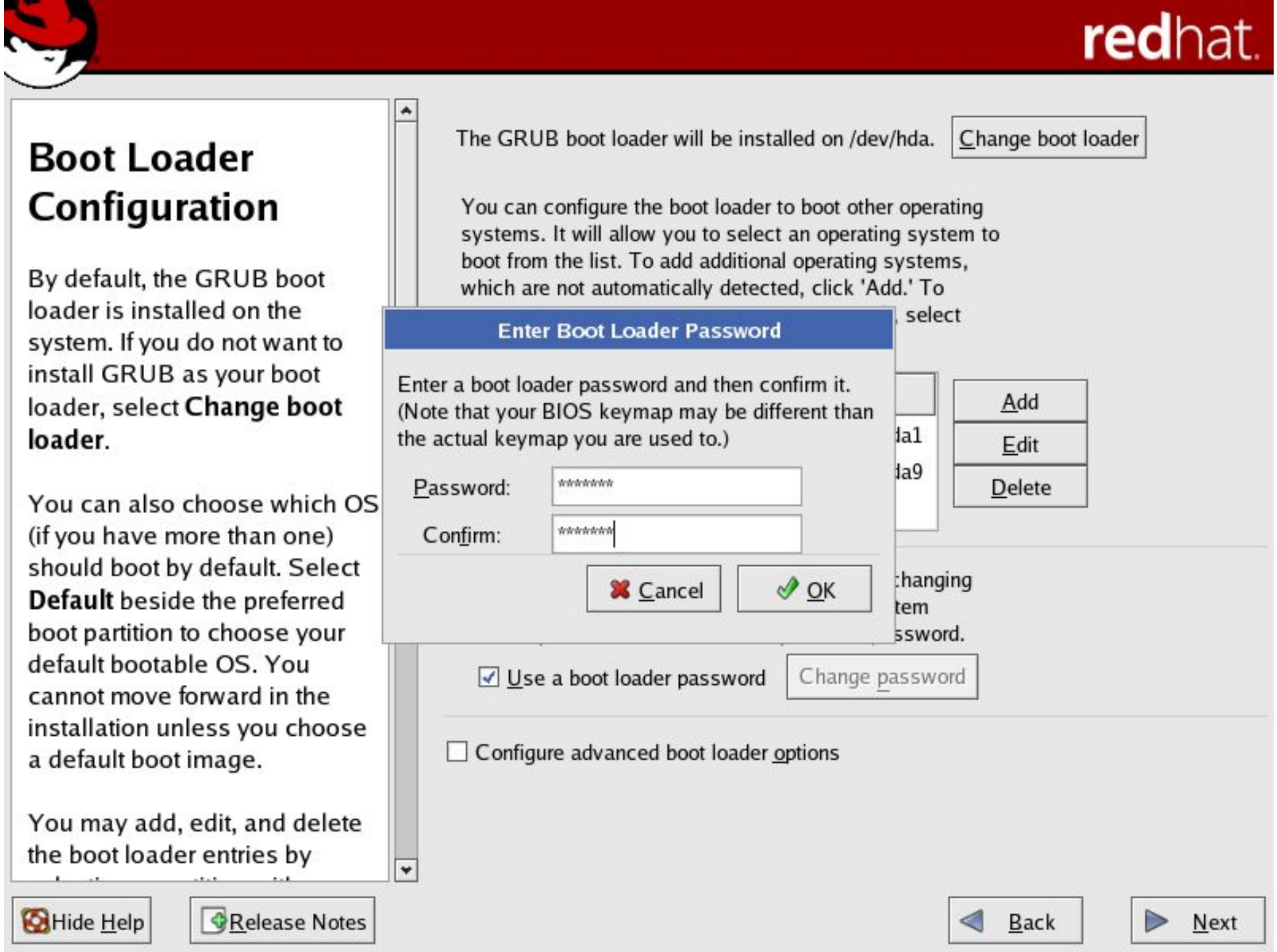

redhat.
Disk Setup
Choose where you would like Red Hat Enterprise Linux ES to be installed.
If you do not know how to partition your system or if yo need help with using the manual partitioning tools, re to the product documentatio
If you used automatic partitioning, you can either accept the current partition settings (click Next), or mod the setup using the manual partitioning tool.
If you are manually partitioning your system, you can see your current hard drive(s) and partitions displayed below. Use
Drive /dev/hda (76317 MB) (Model: FUJITSU MHT2080AT)
hda1 20002 MB
hda5 5004 M
hda6 15005 MB
hd 20
hda9 34130 MB
Edit Partition: /dev/hda7
Mount Point:
<Not Applicable>
Original File System Type:
swap
Size (MB):
2047
How would you like to prepare the file system on this partition?
Leave unchanged (preserve data)
Format partition as:
swap
Cancel
OK
RAID
LVM
Format
Size (MB)
Start
End
20003 1 2550
56298 2551 9727
5005 2551 3188
15006 3189 5101
/dev/hda7 swap 2047 5102 5362
/dev/hda8 ext3 110 5363 5376
/dev/hda9 ext3 34130 5377 9727
Free Free space 16 9728 9729
Hide RAID device/LVM Volume Group members
Hide Help
Release Notes
Back
Next

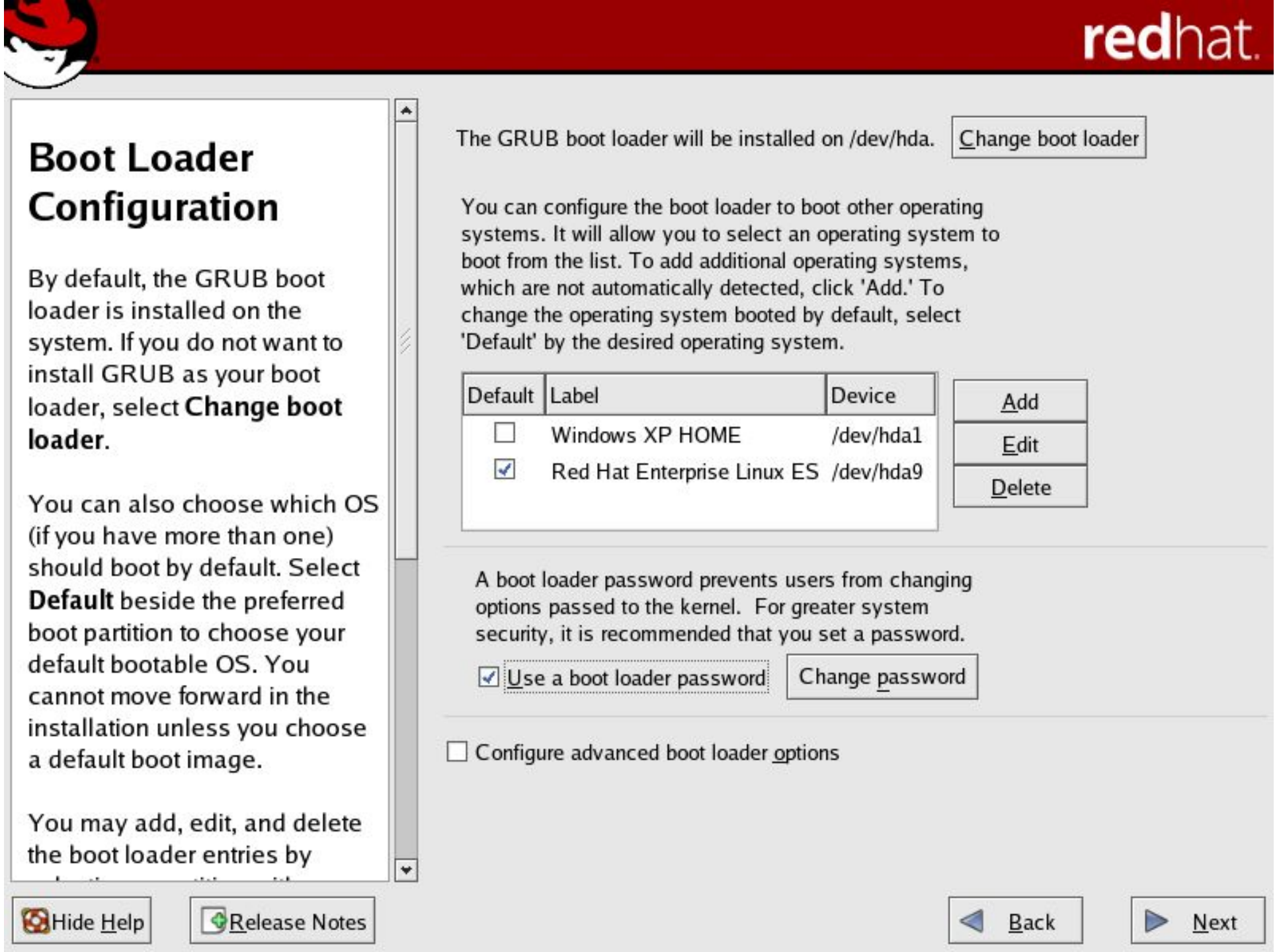

redhat.
Disk Setup
Choose where you would like Red Hat Enterprise Linux E be installed.
If you do not know how to partition your system or if yo need help with using the manual partitioning tools, re to the product documentatio
If you used automatic partitioning, you can either accept the current partition settings (click Next), or mod the setup using the manual partitioning tool.
If you are manually partitioning your system, you can see your current hard drive(s) and partitions displayed below. Use
Drive /dev/hda (76317 MB) (Model: FUJITSU MHT2080AT)
hda1
hda5
hda6
hda9
MB
Edit Partition: /dev/hda8
Mount Point:
/boot
Original File System Type:
ext3
Original File System Label:
/boot
Size (MB):
109
How would you like to prepare the file system on this partition?
Leave unchanged (preserve data)
Format partition as:
ext3
Cancel
OK
RAID
LVM
Format
Size (MB)
Start
End
20003 1 2550
56298 2551 9727
5005 2551 3188
15006 3189 5101
2047 5102 5362
/dev/hda8 ext3 110 5363 5376
/dev/hda9 ext3 34130 5377 9727
Free Free space 16 9728 9729
Hide RAID device/LVM Volume Group members
Hide Help
Release Notes
Back
Next

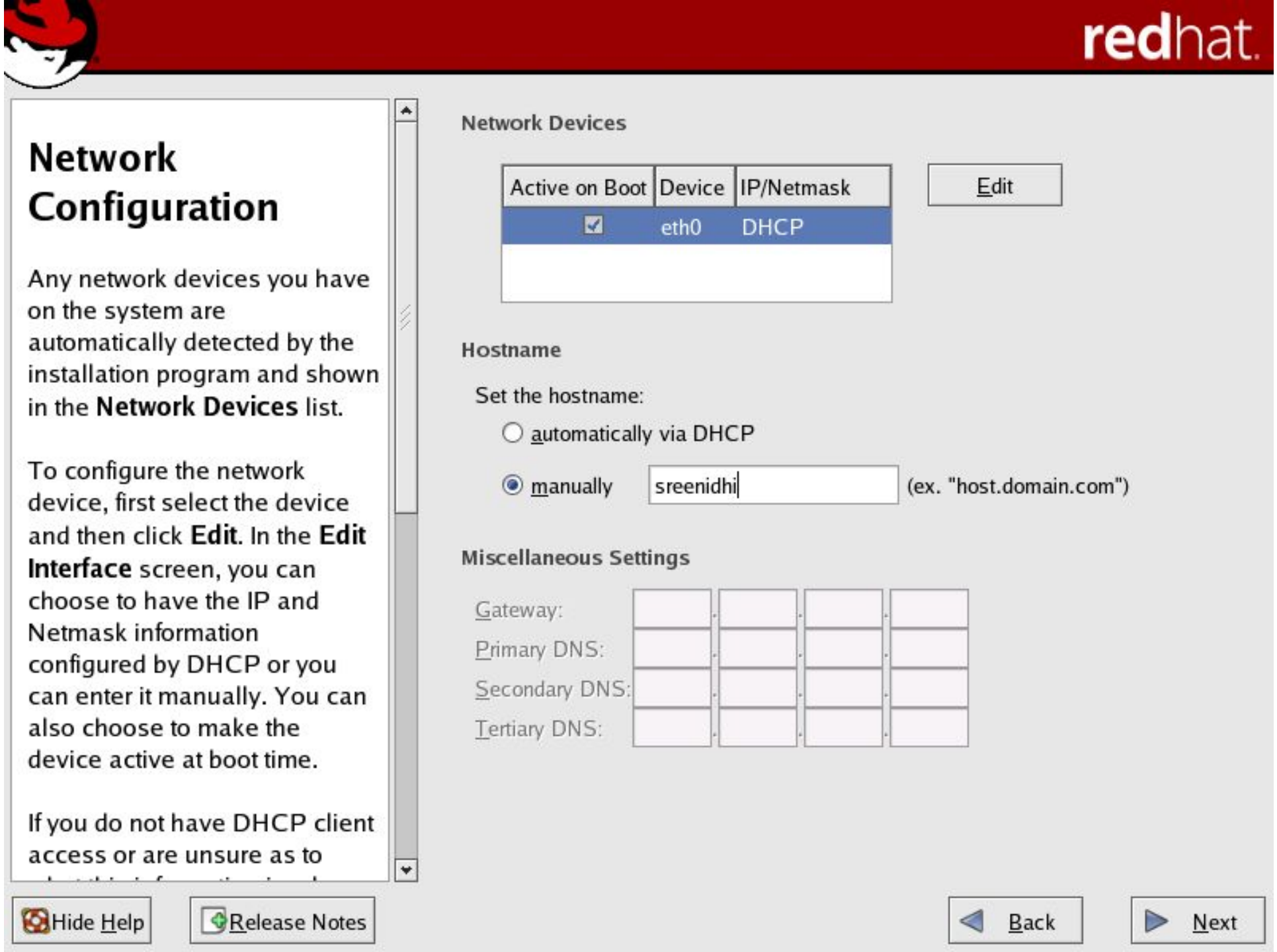

redhat.
Disk Setup
Choose where you would like Red Hat Enterprise Linux E be installed.
If you do not know how to partition your system or if yo need help with using the manual partitioning tools, re to the product documentatio
If you used automatic partitioning, you can either accept the current partition settings (click Next), or mod the setup using the manual partitioning tool.
If you are manually partitioning your system, you can see your current hard drive(s) and partitions displayed below. Use
Drive /dev/hda (76317 MB) (Model: FUJITSU MHT2080AT)
hda1
hda5
hda6
hda9
MB
Edit Partition: /dev/hda9
Mount Point:
/
Original File System Type:
ext3
Original File System Label:
/
Size (MB):
34130
How would you like to prepare the file system on this partition?
Leave unchanged (preserve data)
Format partition as:
ext3
Cancel
OK
RAID
LVM
Format
Size (MB)
Start
End
20003 1 2550
56298 2551 9727
5005 2551 3188
15006 3189 5101
2047 5102 5362
/dev/hda8 /boot ext3 110 5363 5376
/dev/hda9 ext3 34130 5377 9727
Free Free space 16 9728 9729
Hide RAID device/LVM Volume Group members
Hide Help
Release Notes
Back
Next

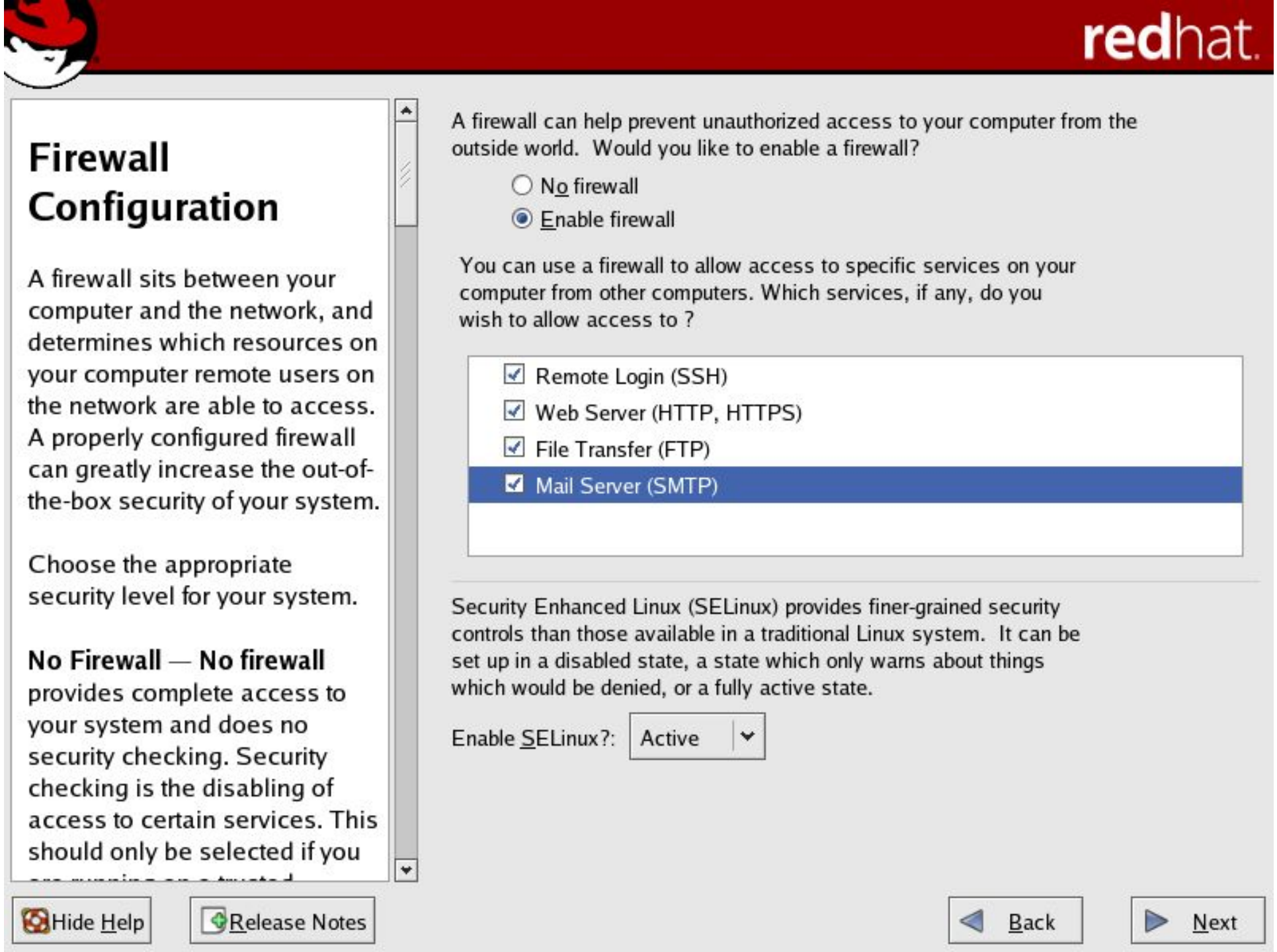

redhat.
Disk Setup
Choose where you would like Red Hat Enterprise Linux ES to be installed.
If you do not know how to partition your system or if you need help with using the manual partitioning tools, refer to the product documentation.
If you used automatic partitioning, you can either accept the current partition settings (click Next), or modify the setup using the manual partitioning tool.
If you are manually partitioning your system, you can see your current hard drive(s) and partitions displayed below. Use
Drive /dev/hda (76317 MB) (Model: FUJITSU MHT2080AT)
hda1 20002 MB
hda5 5004 M
hda6 15005 MB
hd hda9
20 34130 MB
New
Edit
Delete
Reset
RAID
LVM
Device
Mount Point/ RAID/Volume
Type
Format
Size (MB)
Start
End
/dev/hda1 ntfs 20003 1 2550
/dev/hda2 Extended 56298 2551 9727
/dev/hda5 vfat 5005 2551 3188
/dev/hda6 vfat 15006 3189 5101
/dev/hda7 swap 2047 5102 5362
/dev/hda8 /boot ext3 110 5363 5376
/dev/hda9 / ext3 34130 5377 9727
Free Free space 16 9728 9729
Hide RAID device/LVM Volume Group members
Hide Help
Release Notes
Back
Next

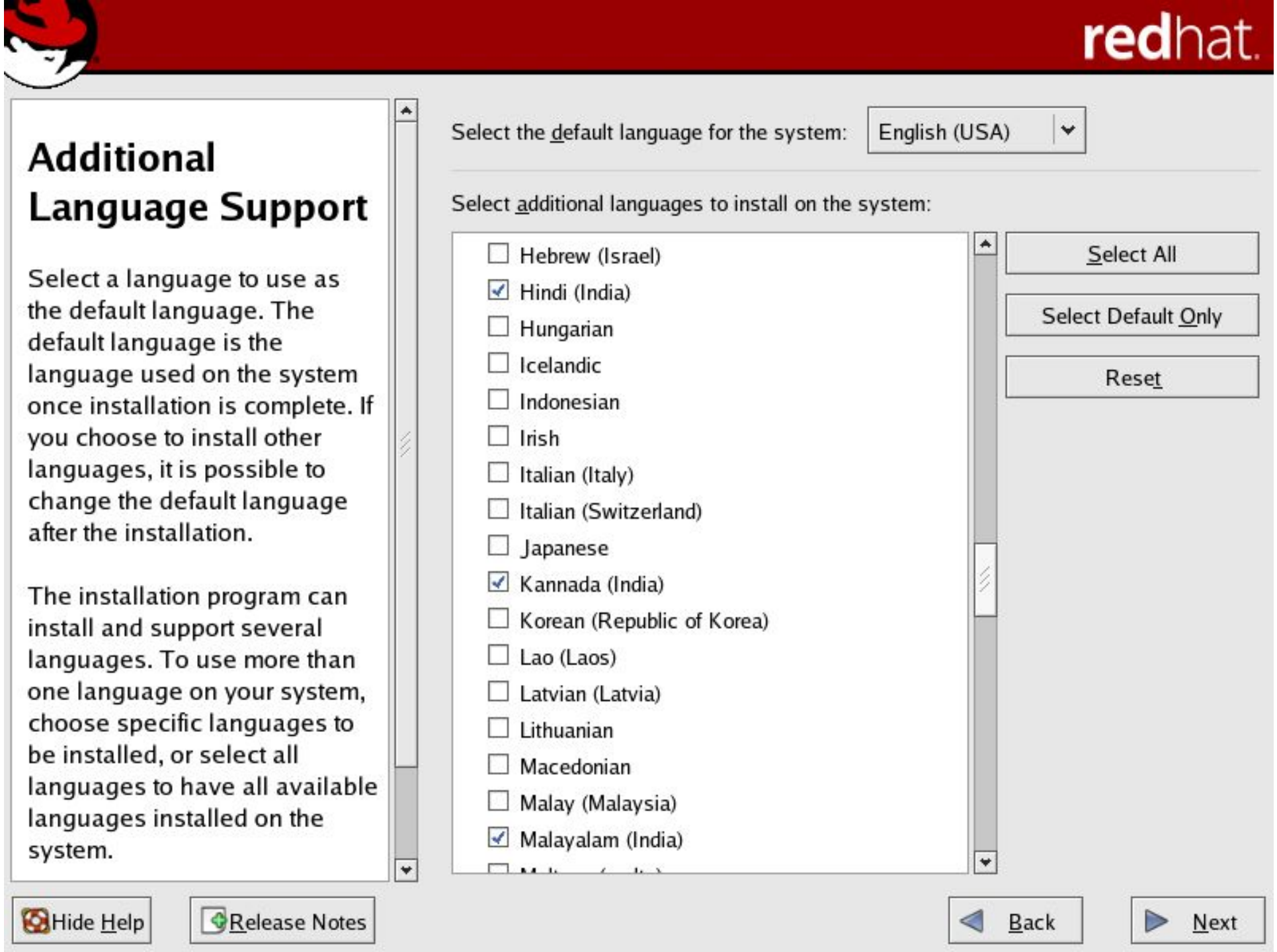

redhat.
Disk Setup
Choose where you w
Red Hat Enterprise L
be installed.
If you do not know ho
partition your system
need help with using
manual partitioning t
to the product docum
If you used automatic
partitioning, you can
accept the current pa
settings (click Next),
the setup using the n
partitioning tool.
If you are manually partitioning
your system, you can see your
current hard drive(s) and
partitions displayed below. Use
Format Warnings
The following pre-existing partitions have been selected to be formatted, destroying all data.
/dev/hda7 swap
/dev/hda8 ext3 /boot
/dev/hda9 ext3 /
Cancel
Format
RAID
LVM
Size (MB)
Start
End
20003 1 2550
56298 2551 9727
5005 2551 3188
15006 3189 5101
2047 5102 5362
110 5363 5376
34130 5377 9727
Free
Free space
16 9728 9729
Hide RAID device/LVM Volume Group members
Hide Help
Release Notes
Back
Next


redhat.
Boot Loader Configuration
By default, the GRUB boot loader is installed on the system. If you do not want to install GRUB as your boot loader, select Change boot loader.
You can also choose which OS (if you have more than one) should boot by default. Select Default beside the preferred boot partition to choose your default bootable OS. You cannot move forward in the installation unless you choose a default boot image.
You may add, edit, and delete the boot loader entries by
The GRUB boot loader will be installed on /dev/hda.
Change boot loader
You can configure the boot loader to boot other operating systems. It will allow you to select an operating system to boot from the list. To add additional operating systems, which are not automatically detected, click 'Add.' To
select
Add
Edit
Delete
Enter Boot Loader Password
Enter a boot loader password and then confirm it. (Note that your BIOS keymap may be different than the actual keymap you are used to.)
Password:
Confirm:
Cancel
OK
changing
tem
ssword.
Use a boot loader password
Change password
Configure advanced boot loader options
Hide Help
Release Notes
Back
Next


redhat.
Boot Loader Configuration
By default, the GRUB boot loader is installed on the system. If you do not want to install GRUB as your boot loader, select Change boot loader.
You can also choose which OS (if you have more than one) should boot by default. Select Default beside the preferred boot partition to choose your default bootable OS. You cannot move forward in the installation unless you choose a default boot image.
You may add, edit, and delete the boot loader entries by
The GRUB boot loader will be installed on /dev/hda.
Change boot loader
You can configure the boot loader to boot other operating systems. It will allow you to select an operating system to boot from the list. To add additional operating systems, which are not automatically detected, click 'Add.' To change the operating system booted by default, select 'Default' by the desired operating system.
Default
Label
Device
Windows XP HOME
/dev/hda1
Red Hat Enterprise Linux ES
/dev/hda9
Add
Edit
Delete
A boot loader password prevents users from changing options passed to the kernel. For greater system security, it is recommended that you set a password.
Use a boot loader password
Change password
Configure advanced boot loader options
Hide Help
Release Notes
Back
Next


redhat.
Network Configuration
Any network devices you have on the system are automatically detected by the installation program and shown in the Network Devices list.
To configure the network device, first select the device and then click Edit. In the Edit Interface screen, you can choose to have the IP and Netmask information configured by DHCP or you can enter it manually. You can also choose to make the device active at boot time.
If you do not have DHCP client access or are unsure as to
Network Devices
Active on Boot
Device
IP/Netmask
eth0
DHCP
Edit
Hostname
Set the hostname:
automatically via DHCP
manually
sreenidhi
(ex. "host.domain.com")
Miscellaneous Settings
Gateway:
Primary DNS:
Secondary DNS:
Tertiary DNS:
Hide Help
Release Notes
Back
Next


redhat.
Firewall Configuration
A firewall sits between your computer and the network, and determines which resources on your computer remote users on the network are able to access. A properly configured firewall can greatly increase the out-of-the-box security of your system.
Choose the appropriate security level for your system.
No Firewall — No firewall provides complete access to your system and does no security checking. Security checking is the disabling of access to certain services. This should only be selected if you
A firewall can help prevent unauthorized access to your computer from the outside world. Would you like to enable a firewall?
No firewall
Enable firewall
You can use a firewall to allow access to specific services on your computer from other computers. Which services, if any, do you wish to allow access to ?
Remote Login (SSH)
Web Server (HTTP, HTTPS)
File Transfer (FTP)
Mail Server (SMTP)
Security Enhanced Linux (SELinux) provides finer-grained security controls than those available in a traditional Linux system. It can be set up in a disabled state, a state which only warns about things which would be denied, or a fully active state.
Enable SELinux?:
Active
Hide Help
Release Notes
Back
Next


redhat.
Additional Language Support
Select a language to use as the default language. The default language is the language used on the system once installation is complete. If you choose to install other languages, it is possible to change the default language after the installation.
The installation program can install and support several languages. To use more than one language on your system, choose specific languages to be installed, or select all languages to have all available languages installed on the system.
Select the default language for the system:
English (USA)
Select additional languages to install on the system:
Hebrew (Israel)
Hindi (India)
Hungarian
Icelandic
Indonesian
Irish
Italian (Italy)
Italian (Switzerland)
Japanese
Kannada (India)
Korean (Republic of Korea)
Lao (Laos)
Latvian (Latvia)
Lithuanian
Macedonian
Malay (Malaysia)
Malayalam (India)
Select All
Select Default Only
Reset
Hide Help
Release Notes
Back
Next

redhat.

Time Zone Selection

Set your time zone by selecting your computer's physical location.

On the interactive map, click on a specific city (marked by a yellow dot) and a red **X** appears indicating your selection.

You can also scroll through the list of locations to select your desired time zone.

You can also select the **System Clock uses UTC** option. (UTC, or Coordinated Universal Time, allows your system to properly handle

Please select the nearest city in your timezone:

Location	Description
America/Nassau	
America/New_York	Eastern Time
America/Nipigon	Eastern Time - Ontario & Quebec - places th

☐ System clock uses UTC

Hide Help | Release Notes | Back | Next

redhat.

Set Root Password

Use the root account *only* for administration. Once the installation has been completed, create a non-root account for your general use and su – to gain root access when you need to fix something quickly. These basic rules minimize the chances of a typo or incorrect command doing damage to your system.

The root account is used for administering the system.
Enter a password for the root user.

Root Password: ********

Confirm: ********

Hide Help | Release Notes | Back | Next

redhat.

Package Installation Defaults

The installation program automatically chooses package groups to be installed on the system.

Select **Accept the current package list** to accept the default package groups and to continue with the installation process.

Select **Customize the set of packages to be installed** if you wish to select different or additional package groups.

The default installation environment includes our recommended package selection, including:

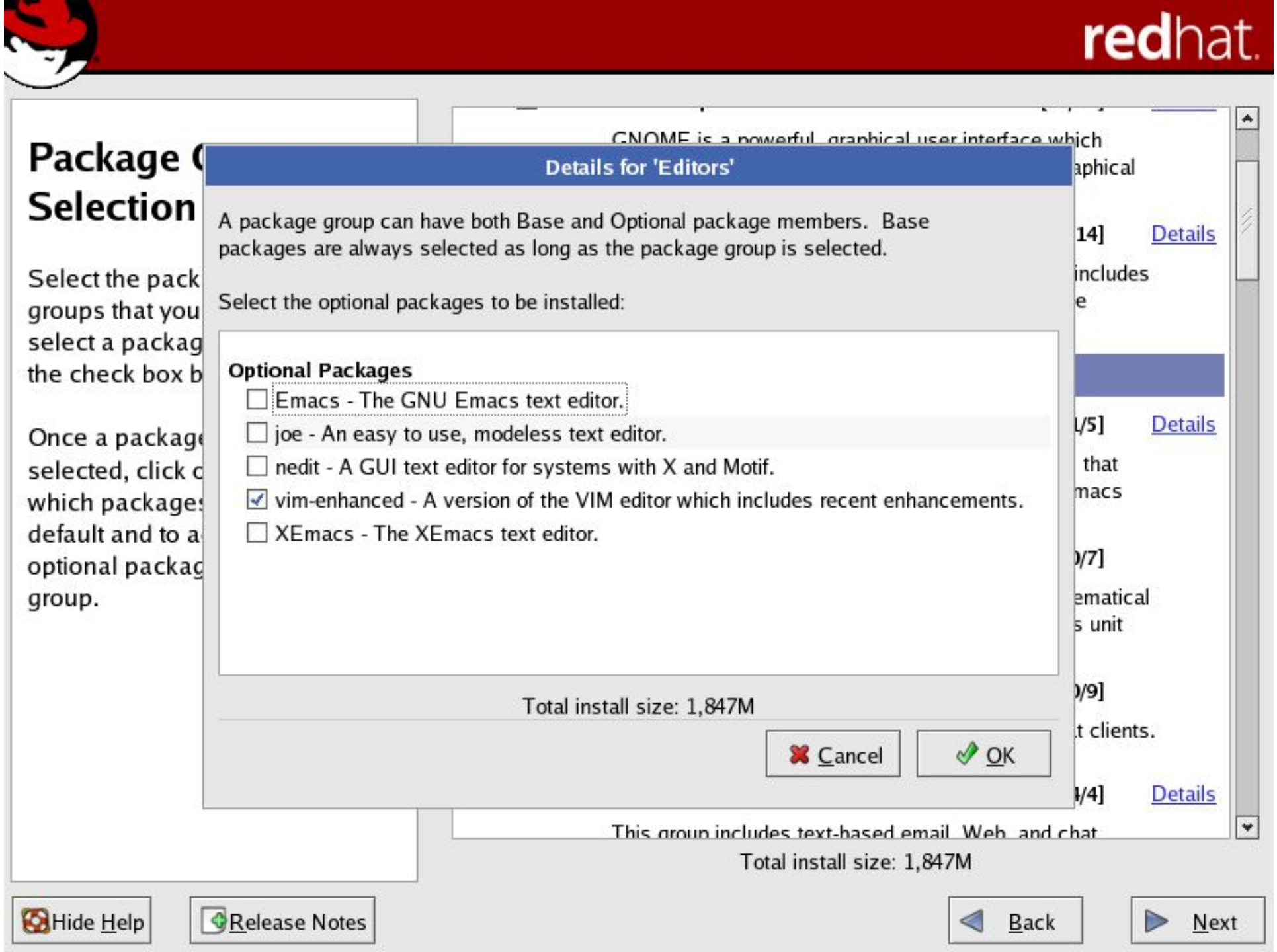
Desktop shell (GNOME)
Administration Tools
Server Configuration Tools
Web Server
Windows File Server (SMB)

After installation, additional software can be added or removed using the 'system-config-packages' tool.

If you are familiar with Red Hat Enterprise Linux ES, you may have specific packages you would like to install or avoid installing. Check the box below to customize your installation.

- ○ Install default software packages
- ◉ Customize software packages to be installed

Hide Help | Release Notes | Back | Next


redhat.
Package
Selection
Select the pack
groups that you
select a packag
the check box b
Once a package
selected, click c
which packages
default and to a
optional packag
group.
GNOME is a powerful, graphical user interface which
aphical
14]
Details
includes
e
Details
that
macs
0/7]
ematical
s unit
0/9]
t clients.
4/4]
Details
This group includes text-based email, Web, and chat
Total install size: 1,847M
Details for 'Editors'
A package group can have both Base and Optional package members. Base packages are always selected as long as the package group is selected.
Select the optional packages to be installed:
Optional Packages
Emacs - The GNU Emacs text editor.
joe - An easy to use, modeless text editor.
nedit - A GUI text editor for systems with X and Motif.
vim-enhanced - A version of the VIM editor which includes recent enhancements.
XEmacs - The XEmacs text editor.
Total install size: 1,847M
Cancel
OK
Hide Help
Release Notes
Back
Next

redhat.

Package Group Selection

Select the package (application) groups that you want to install. To select a package group, click on the check box beside it.

Once a package group has been selected, click on **Details** to view which packages are installed by default and to add or remove optional packages from that group.

System

- [x] **Administration Tools** [12/12] Details

This group is a collection of graphical administration tools for the system, such as for managing user accounts and configuring system hardware.

- [x] **System Tools** [24/34] Details

This group is a collection of various tools for the system, such as the client for connecting to SMB shares and tools to monitor network traffic.

- [x] **Printing Support** [12/12] Details

Install these tools to enable the system to print or act as a print server.

Miscellaneous

- [ ] **Everything**

This group includes all the packages available. Note that there are substantially more packages than just the ones in all the other package groups on this page.

- [ ] **Minimal**

Choose this group to get the minimal possible set of packages. Useful for creating small router/firewall boxes, for example.

Total install size: 3,500M

Hide Help | Release Notes | Back | Next

redhat.

About to Install

Caution: Once you click **Next**, the installation program begins writing the operating system to the hard drive(s). This process cannot be undone. If you have decided not to continue with this installation, this is the last point at which you can safely abort the installation process.

To abort this installation, press your computer's Reset button or reset using **Control-Alt-Delete**, and then remove the installation media between the unmounting and reboot screen messages.

Click next to begin installation of Red Hat Enterprise Linux ES.

A complete log of the installation can be found in the file '/root/install.log' after rebooting your system.

A kickstart file containing the installation options selected can be found in the file '/root/anaconda-ks.cfg' after rebooting the system.

Hide Help | Release Notes | Back | Next


redhat.
About to Install
Caution: Once you click Next,
the installation program
writing the operating sys
the hard drive(s). This p
cannot be undone. If you
decided not to continue
installation, this is the la
which you can safely ab
installation process.
To abort this installation
your computer's Reset
reset using Control-Alt-
and then remove the ins
media between the unm
and reboot screen messages.
Required Install Media
The software you have selected to install will require the following CDs:
Red Hat Enterprise Linux ES 4 CD #1
Red Hat Enterprise Linux ES 4 CD #2
Red Hat Enterprise Linux ES 4 CD #3
Red Hat Enterprise Linux ES 4 CD #4
Please have these ready before proceeding with the installation. If you need to abort the installation and reboot please select "Reboot".
Reboot
Continue
ext to begin installation
Hat Enterprise Linux ES.
lete log of the
ion can be found in the
t/install.log' after
g your system.
tart file containing the
ion options selected can
d in the file '/root/
da-ks.cfg' after rebooting
tem.
Hide Help
Release Notes
Back
Next

redhat.

Installing Packages

We have gathered all the information needed to install Red Hat Enterprise Linux ES on the system. It may take a while to install everything, depe
how many packages n
installed.

Change CDROM

Please insert Red Hat Enterprise Linux ES disc 2 to continue.

OK

Installing hwdata-0.146.10.EL-1.noarch (956 KB)
Hardware identification data.

Hide Help | Release Notes | Back | Next

redhat.

Installing Packages

We have gathered all the information needed to install Red Hat Enterprise Linux ES on the system. It may take a while to install everything, depe
how many packages n
installed.

Change CDROM

Please insert Red Hat Enterprise Linux ES disc 4 to continue.

OK

Remaining time: 4 minutes

Installing rusers-0.17-41.i386 (36 KB)
Displays the names of users logged into machines on the local network.

Hide Help | Release Notes | Back | Next

redhat.

Installing Packages

We have gathered all the information needed to install Red Hat Enterprise Linux ES on the system. It may take a while to install everything, depe
how many packages n
installed.

Remaining time: 2 minutes

Installing redhat-release-4ES-2.4.i386 (2 MB)
The Red Hat Linux release file.

Hide Help | Release Notes | Back | Next

redhat.

Congratulations, the installation is complete.

Remove any installation media (diskettes or CD-ROMs) used during the installation process and press the "Reboot" button to reboot your system.

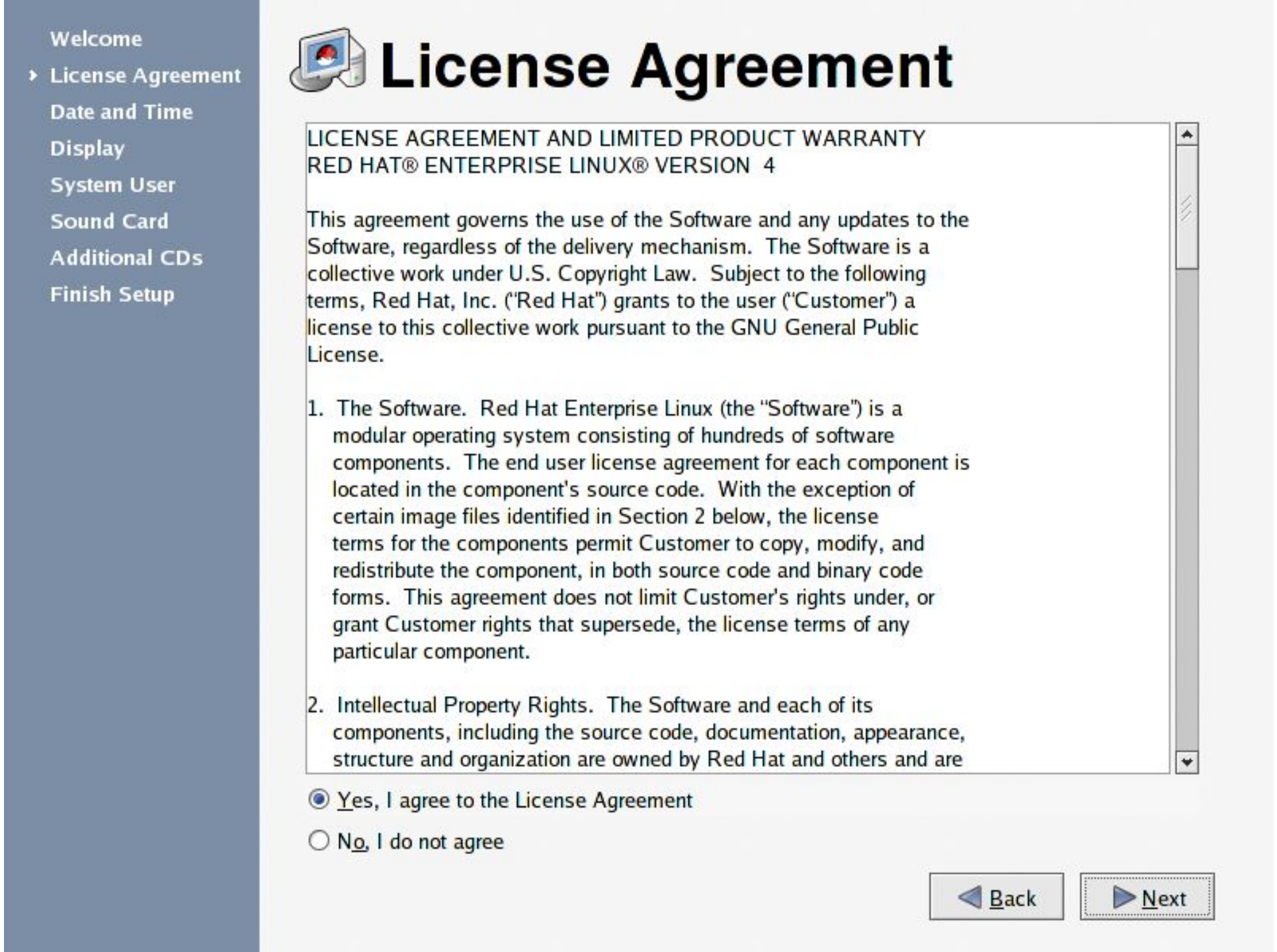
Show Help | Release Notes | Back | Reboot

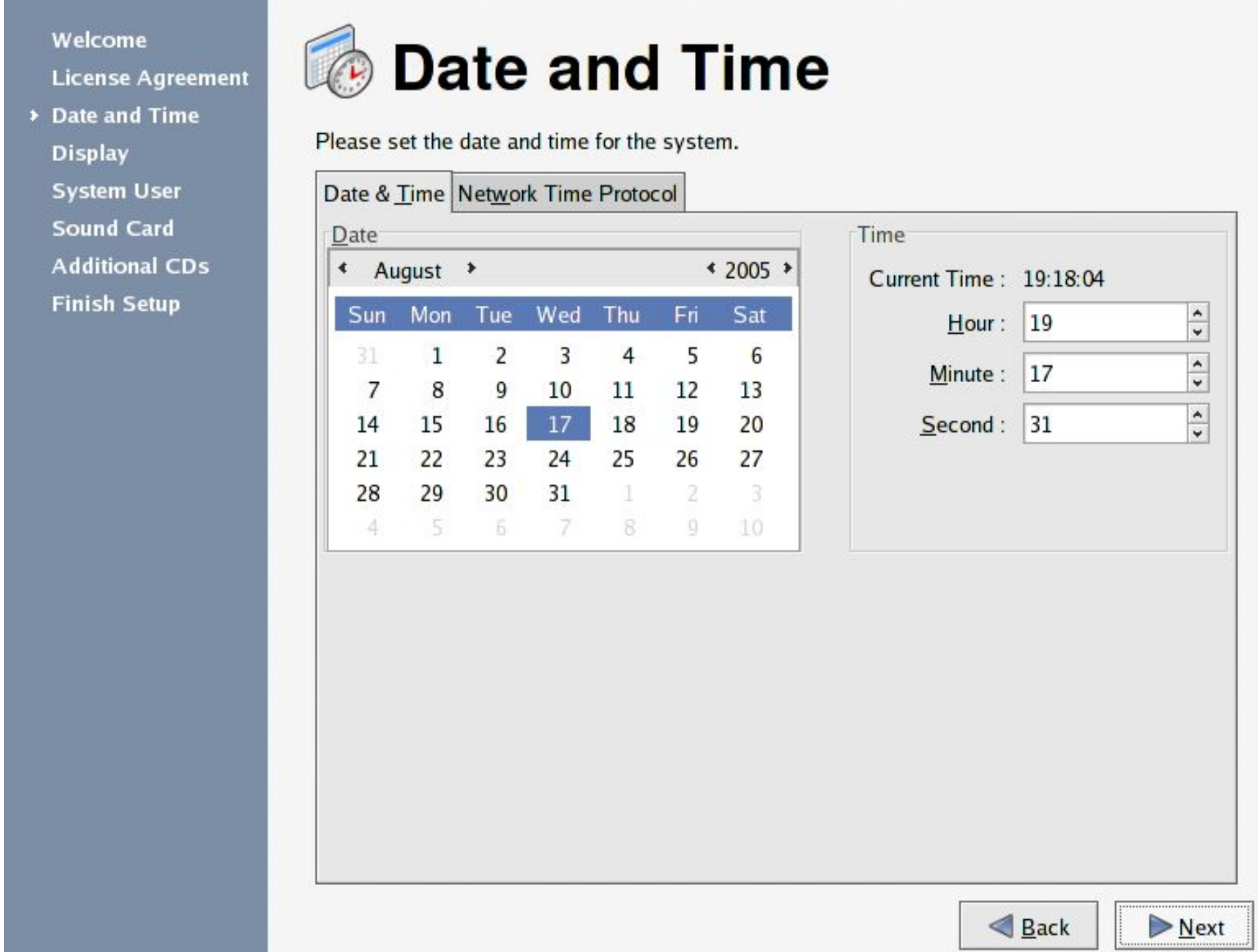
- Welcome
- License Agreement
- Date and Time
- Display
- System User
- Sound Card
- Additional CDs
- Finish Setup

Welcome

There are a few more steps to take before your system is ready to use. The Setup Agent will now guide you through some basic configuration. Please click the "Next" button in the lower right corner to continue.

Back Next

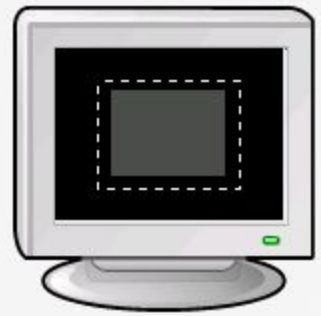

Welcome
License Agreement
Date and Time
Display
System User
Sound Card
Additional CDs
Finish Setup
License Agreement
LICENSE AGREEMENT AND LIMITED PRODUCT WARRANTY
RED HAT® ENTERPRISE LINUX® VERSION 4
This agreement governs the use of the Software and any updates to the Software, regardless of the delivery mechanism. The Software is a collective work under U.S. Copyright Law. Subject to the following terms, Red Hat, Inc. ("Red Hat") grants to the user ("Customer") a license to this collective work pursuant to the GNU General Public License.
1. The Software. Red Hat Enterprise Linux (the "Software") is a modular operating system consisting of hundreds of software components. The end user license agreement for each component is located in the component's source code. With the exception of certain image files identified in Section 2 below, the license terms for the components permit Customer to copy, modify, and redistribute the component, in both source code and binary code forms. This agreement does not limit Customer's rights under, or grant Customer rights that supersede, the license terms of any particular component.
2. Intellectual Property Rights. The Software and each of its components, including the source code, documentation, appearance, structure and organization are owned by Red Hat and others and are
Yes, I agree to the License Agreement
No, I do not agree
Back
Next


Welcome
License Agreement
Date and Time
Display
System User
Sound Card
Additional CDs
Finish Setup
Date and Time
Please set the date and time for the system.
Date & Time
Network Time Protocol
Date
August
2005
Sun Mon Tue Wed Thu Fri Sat
31 1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31 1 2 3
4 5 6 7 8 9 10
Time
Current Time : 19:18:04
Hour : 19
Minute : 17
Second : 31
Back
Next

- Welcome
- License Agreement
- Date and Time
- Display
- System User
- Sound Card
- Additional CDs
- Finish Setup

Display

Please select the resolution and color depth that you wish to use:



LCD Panel 1280x800 with ATI Radeon Mobility 9600 M10 [Configure...]

Resolution: [1024x768]

Color Depth: [Millions of Colors]

[Back] [Next]

Sound Card

A sound card has been detected on your computer.

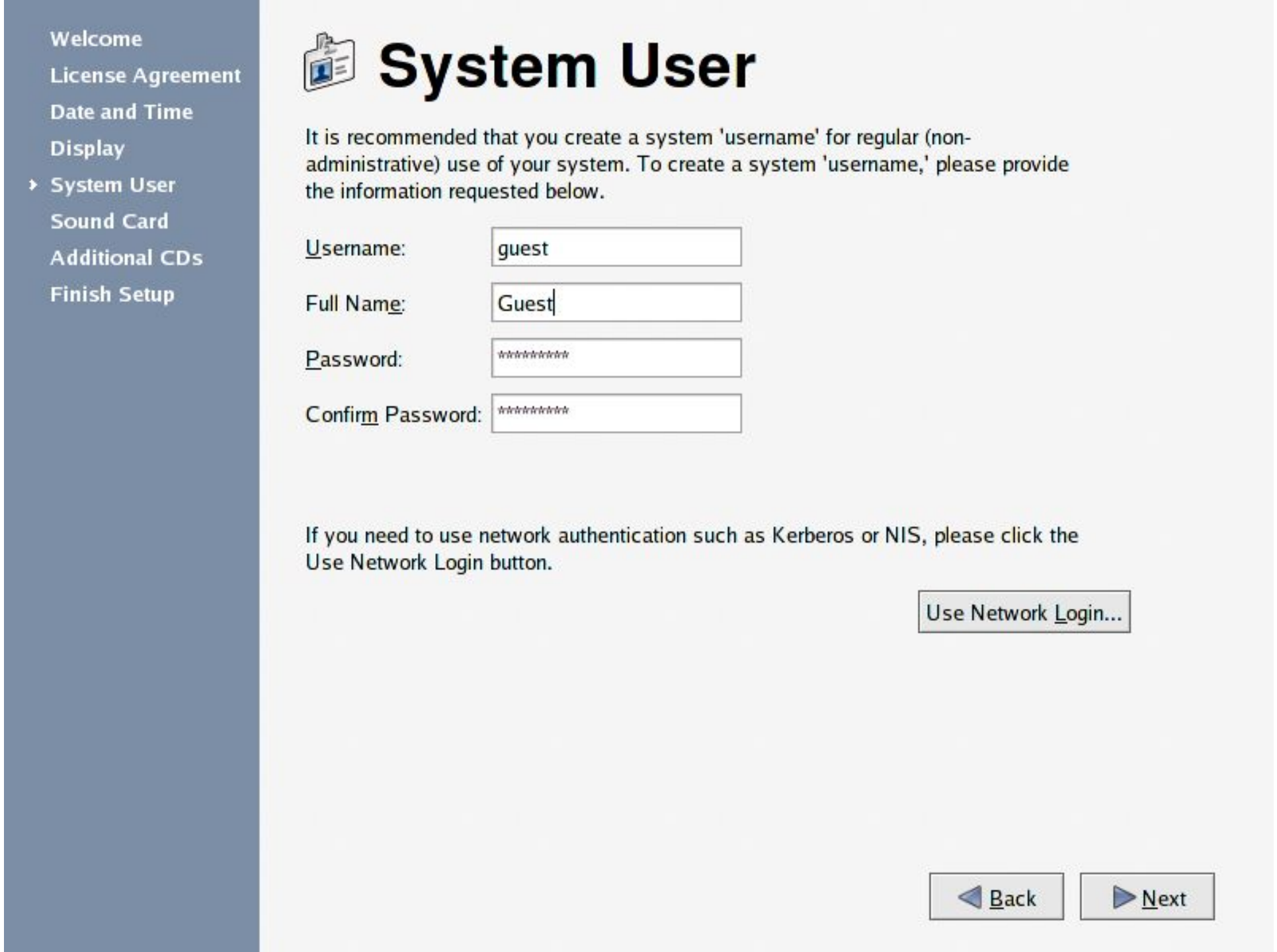
Click the "Play test sound" button to hear a sample sound. You should hear a series of three sounds. The first sound will be in the right channel, the second sound will be in the left channel, and the third sound will be in the center.

Vendor: Intel

Model: Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) High Definition Audio Controlle

Module: snd-azx

Play test sound


Welcome
License Agreement
Date and Time
Display
System User
Sound Card
Additional CDs
Finish Setup
System User
It is recommended that you create a system 'username' for regular (non-administrative) use of your system. To create a system 'username,' please provide the information requested below.
Username:
guest
Full Name:
Guest
Password:
Confirm Password:
If you need to use network authentication such as Kerberos or NIS, please click the Use Network Login button.
Use Network Login...
Back
Next

- Welcome
- License Agreement
- Date and Time
- Display
- System User
- Sound Card
- Additional CDs
- Finish Setup

Additional CDs

Please insert the disc labeled "Red Hat Enterprise Linux Extras" to allow for installation of third-party plug-ins and applications. You may also insert the Documentation disc, or other Red Hat-provided discs to install additional software at this time.

Additional CDs Install...

Back Next

History of Linux

- **History of Linux** began in 1991 with the commencement of a personal project by Finnish student Linus Torvalds to create a new free operating system kernel.
- Since then, the resulting Linux kernel has been marked by constant growth throughout its history.
- Initial release of its source code in 1991, it has grown from a small number of C files under a license prohibiting commercial distribution to the 4.2.3 version in 2015 with more than 18 million lines of source code under the GNU General Public License.
- After AT&T had dropped out of the Multics project, the Unix operating system was conceived and implemented by Ken Thompson and Dennis Ritchie (both of AT&T Bell Laboratories) in 1969 and first released in 1970. Later they rewrote it in a new programming language, C, to make it portable. The availability and portability of Unix caused it to be widely adopted, copied and modified by academic institutions and businesses.

What's Linux?

- Linux is an operating system (OS) that allows the user (you) greater flexibility and control. Linux is an alternative to your standard Windows or Mac OS X. It's a system that allows you to run applications and perform desired functions on your computer. If you didn't have Windows on your computer, how would you do anything? You can't. You need an operating system. It's also a heck of a lot safer, not to mention more fun if you're a programming geek.
- What makes Linux different than the aforementioned OS's is its collaborative development and incredible flexibility. One company does not own Linux, or update Linux or receive economic benefits from Linux. Instead, it's a community of developers. Because of this, the Linux OS is an incredibly efficient and unsung software innovation. As for flexibility, well, you can control virtually every aspect of your computer's operation. You'll be able to have a secure, totally-yours, speedy and current operating system. Can be installed in GUI or Command Mode.

Why Linux?

- There are a few primary reasons users switch to Linux. **First, security.** This is big. Security is like the number one reason I hear why people switch to Linux. Viruses are less of a threat. Not many of these buggers are designed to attack the Linux OS, which means less hassle and worry for you. And then there are network security testers, Linux is a great tool to find the loopholes that the maliciously intended could use to hack your system and steal information. Plus, you can do things like encrypt your entire hard drive or make yourself invisible on the net. It’s pretty cool stuff.
- Secondly, the cost. It’s free! Yes, you don’t have to pay anything to use this wonderfully customizable system. A lot of distributors will have a donation button on their page

- Third, geek-central. Ever wondered what would happen if you could communicate directly with your computer, no buffer? You're not alone. (Surprised?) A lot of Linux users got tired of telling Windows to run a program without being able to see or control the inner workings of this process. Linux will allow you to get your hands dirty in running commands and completely customizing your computer operating experience. The knowledge you'll gain is amazing and, you can impress you less geeky friends.
- And lastly, there are frequent updates. Since this is a community run operating system it's constantly being updated to be the best version of itself it can be. Developers listen to the feedback, make changes and release new versions of their systems. Again, at no cost to you. Thank these developers kindly.

Some Most Popular Linux Distributions

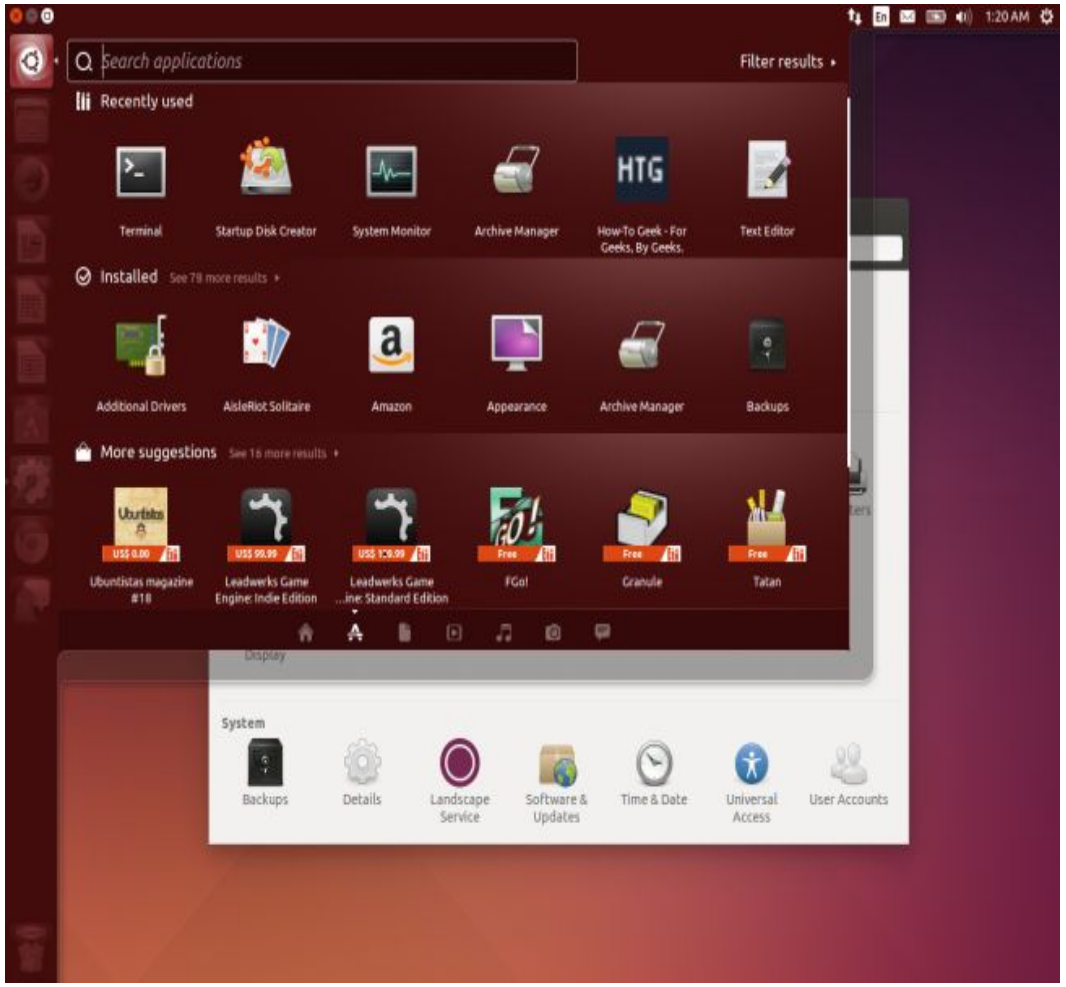
- Linux isn't a complete operating system — it's just a kernel. Linux distributions take the Linux kernel and combine it with other free software to create complete packages. There are many different Linux distributions out there.
- If you want to "install Linux," you'll need to choose a distribution. You could also use Linux From Scratch to compile and assemble your own Linux system from the ground up, but that's a huge amount of work.

Which Linux to Go For

Beginners	Ubuntu	Mint	Zorin	Peppermint	Pinguy
Gaming	Gentoo	Slackware	Ubuntu		
Network Security	Backtrack 5r3	Network Security Toolkit (NST)	nUBUNTU	Kali	Helix
Enterprise Server and Desktop	Redhat	Suse			
Multimedia	Arch	Zeven OS Neptune			
Overall	Debian	Fedora	Mandrivia4	Mageie	Ubuntu

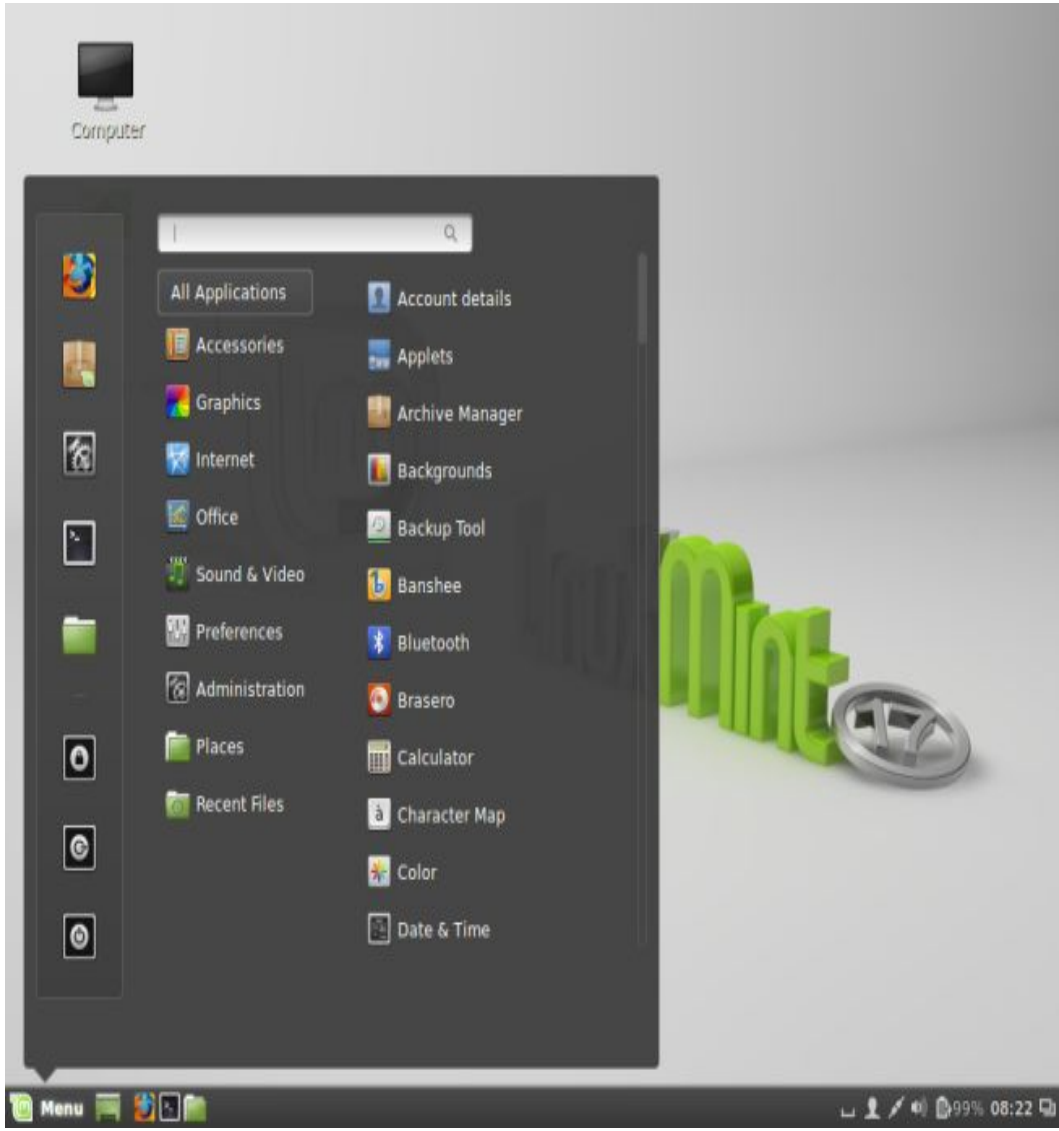
1 Ubuntu

- Probably the most well-known Linux distribution. Based on Debian, but it has its own software repositories. Much of the software in these repositories are synced from Debian's repositories.
- The Ubuntu project has a focus on providing a solid desktop (and server) experience, and it isn't afraid to build its own custom technology to do it. Ubuntu used to use the GNOME desktop environment, but it now uses its own Unity desktop environment.
- It offers releases every six months, with a more stable LTS (long term support) release every two years. Ubuntu is currently working on expanding the Ubuntu distribution to run on smartphones and tablets.



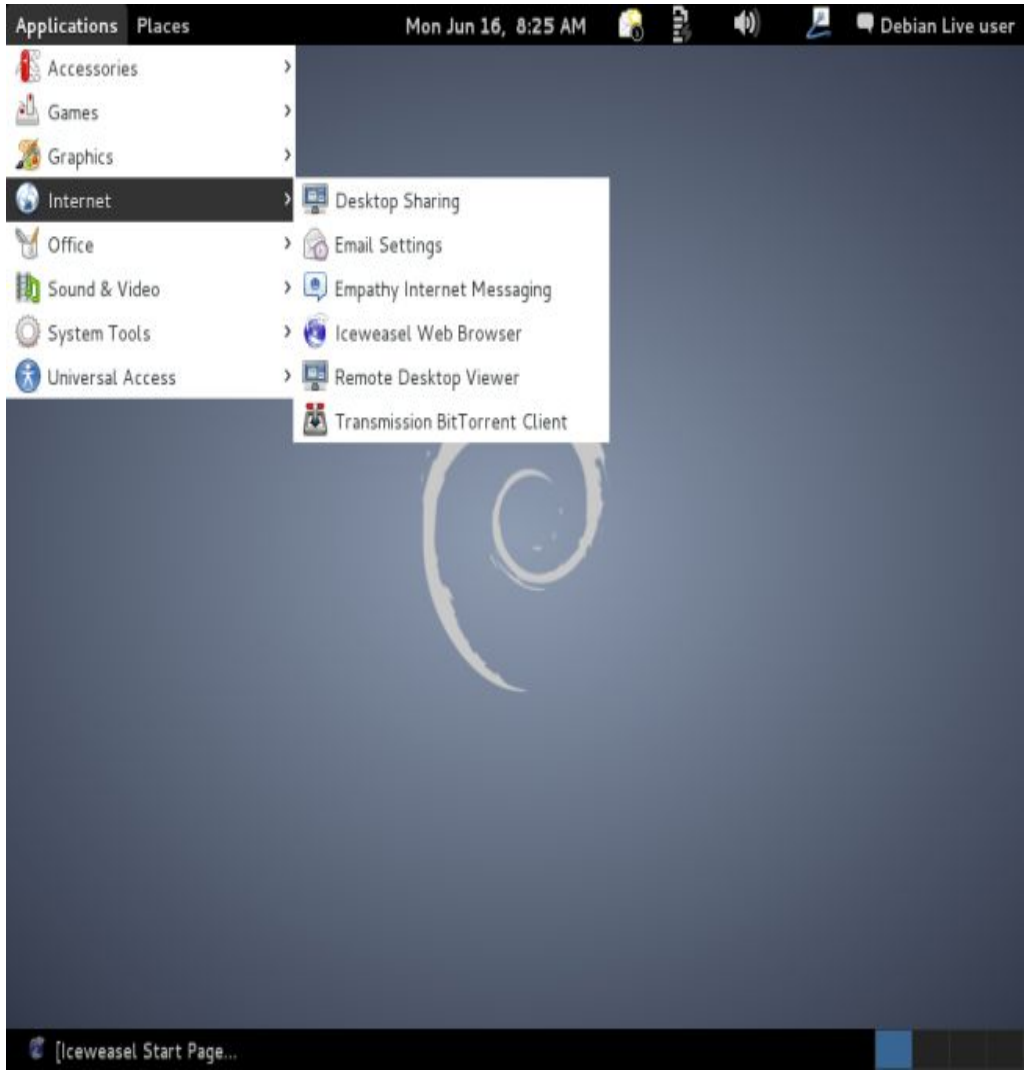
2 LINUX MINT

- Mint is a Linux distribution built on top of Ubuntu. It uses Ubuntu's software repositories, so the same packages are available on both. Originally, Mint was an alternative distribution loved mainly because it included media codecs and proprietary software that Ubuntu didn't include by default.
- This distribution now has its own identity. You won't find Ubuntu's own Unity desktop here — instead, you get a more traditional Cinnamon or MATE desktop. Mint takes a more relaxed approach to software updates and won't automatically install critical software updates. Controversially, this has led some Ubuntu developers to label it insecure.



3 Debian

- Debian is an operating system composed only of free, open-source software. The Debian project has been operating since 1993 — over 20 years ago! This widely respected project is still releasing new versions of Debian, but it's known for moving much more slowly than distributions like Ubuntu or Linux Mint. This can make it more stable and conservative, which is ideal for some systems.
- Ubuntu was originally founded to take the core bits of stable Debian and improve on them more quickly, packaging the software together into a user-friendly system that's more frequently updated.

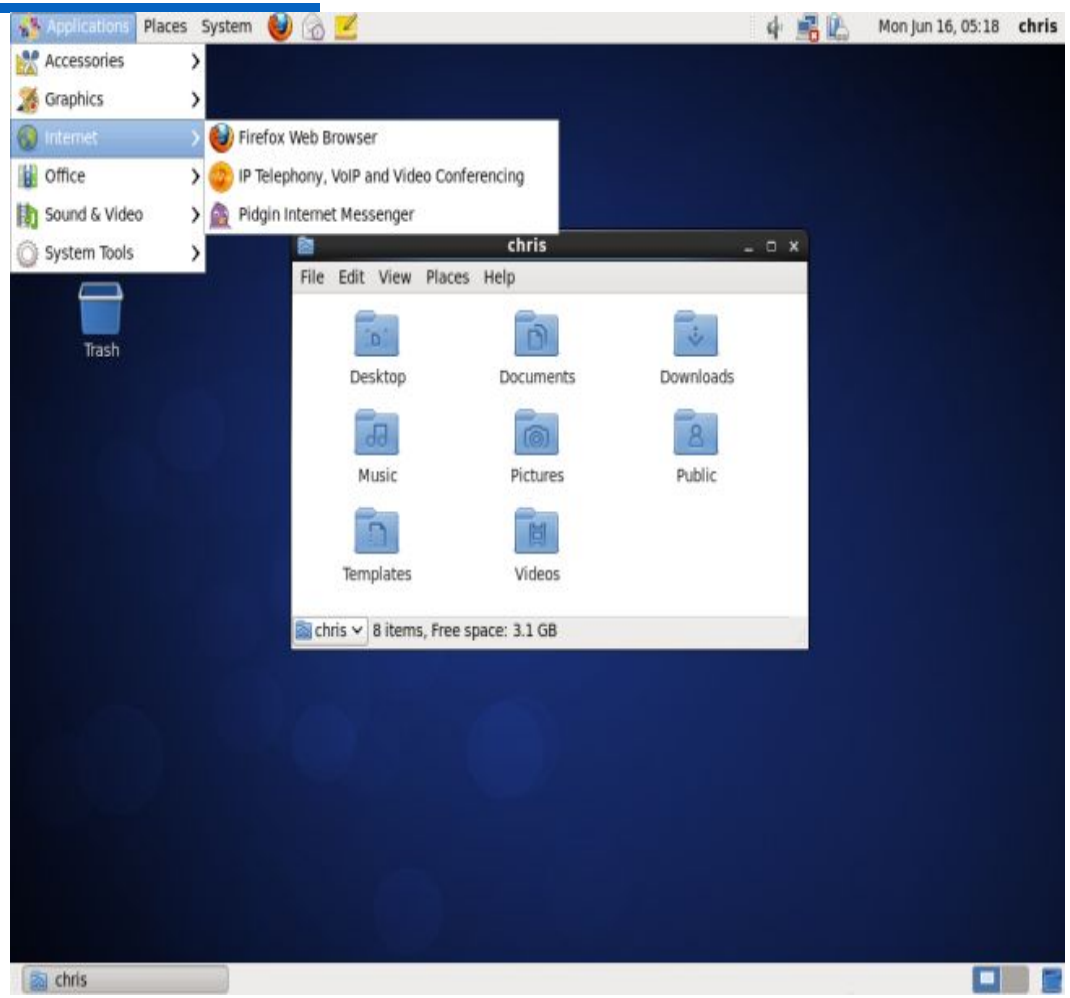


4 Fedora

- Fedora is a project with a strong focus on free software — you won't find an easy way to install proprietary graphics drivers here, although third-party repositories are available. Fedora is bleeding edge and contains the latest versions of software.
- Unlike Ubuntu, Fedora doesn't make its own desktop environment or other software Fedora comes with the GNOME 3 desktop environment by default.
- Fedora is sponsored by Red Hat, and is the foundation for the commercial Red Hat Enterprise Linux project. Unlike RHEL, Fedora is bleeding edge and not supported for long. If you want a more stable release that's supported for longer, Red Hat would prefer you use their Enterprise product.

5 CentOS / Red Hat Enterprise Linux

- Red Hat Enterprise Linux is a commercial Linux distribution intended for servers and workstations. It's based on the open-source Fedora project, but is designed to be a stable platform with long-term support.
- Red Hat uses trademark law to prevent their official Red Hat Enterprise Linux software from being redistributed. However, the core software is free and open-source. CentOS is a community project that takes the Red Hat Enterprise Linux code, removes all Red Hat's trademarks, and makes it available for free use and distribution. It's a free version of RHEL, so it's good if you want a stable platform that will be supported for a long time. CentOS and Red Hat recently announced they're collaborating, so CentOS is now part of Red Hat itself.

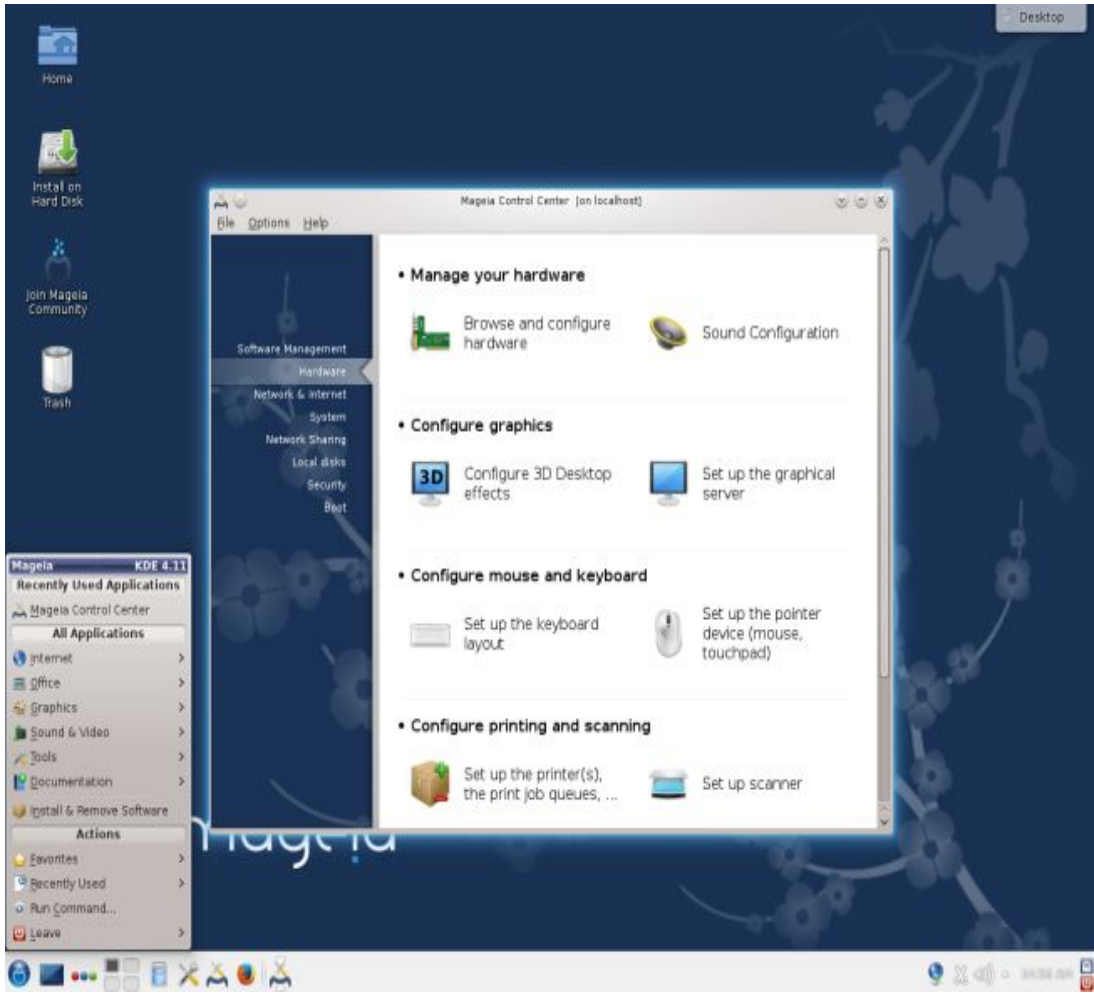


6 openSUSE / SUSE Linux Enterprise

- openSUSE is a community-created Linux distribution sponsored by Novell. Novell purchased SuSE Linux in 2003, and they still create an enterprise Linux project known as SUSE Linux Enterprise. Where Red Hat has the Fedora project that feeds into Red Hat Enterprise Linux, Novell has the openSUSE project that feeds into SUSE Linux Enterprise.
- Like Fedora, openSUSE is a more bleeding edge version of Linux. SUSE was once one of the great user-friendly desktop Linux distributions, but Ubuntu eventually took that crown.

7 Mageia / Mandriva

- Mageia is a fork of Mandriva Linux created in 2011. Mandriva — known as Mandrake before that — was once one of the great user-friendly Linux distributions.
- Like Fedora and openSUSE, this is a community-created project to create an open-source Linux distribution. Mandriva SA no longer creates a consumer Linux distribution for desktop PCs, but their business Linux server projects are based on Mageia code — just like how Fedora and openSUSE provide code to their enterprise equivalents.



Web Links for Linux Distributions

- https://en.wikipedia.org/wiki/Comparison_of_Linux_distributions
- https://en.wikipedia.org/wiki/List_of_Linux_distributions

What is open source and Open Source Software?

- The term "open source" refers to something people can modify and share because its design is publicly accessible. Open source software is software with **source code** that anyone can inspect, modify, and enhance
- **What's the difference between open source software and other types of software?**
- Some software has source code that only the person, team, or organization who created it—and maintains exclusive control over it—can modify. People call this kind of software "proprietary" or "**closed source**" software.
- Microsoft Office and Adobe Photoshop are examples of **proprietary** software.
- LibreOfficeand the GNU Image Manipulation Program are examples of open source software

Why do people prefer using open source software?

- People prefer open source software to proprietary software for a number of reasons, including:
- **Control.** Many people prefer open source software because they have more controlover that kind of software. They can examine the code to make sure it's not doing anything they don't want it to do, and they can change parts of it they don't like. Users who aren't programmers also benefit from open source software, because they can use this software for any purpose they wish—not merely the way someone else thinks they should.
- **Training.** Other people like open source software because it helps them become better programmers. Because open source code is publicly accessible, students can easily study it as they learn to make better software. Students can also share their work with others, inviting comment and critique, as they develop their skills. When people discover mistakes in programs' source code, they can share those mistakes with others to help them avoid making those same mistakes themselves.

- **Security.** Some people prefer open source software because they consider it more secure and stable than proprietary software. Because anyone can view and modify open source software, someone might spot and correct errors or omissions that a program's original authors might have missed. And because so many programmers can work on a piece of open source software without asking for permission from original authors, they can fix, update, and upgrade open source software more quicklythan they can proprietary software.
- **Stability.** Many users prefer open source software to proprietary software for important, long-term projects. Because programmers publicly distribute the source code for open source software, users relying on that software for critical tasks can be sure their tools won't disappear or fall into disrepair if their original creators stop working on them. Additionally, open source software tends to both incorporate and operate according to open standards.

Doesn't "open source" just mean something is free of charge?

- No. This is a common misconception about what "open source" implies, and the concept's implications are not only economic.
- Open source software programmers can charge money for the open source software they create or to which they contribute. But in some cases, because an open source license might require them to release their source code when they sell software to others, some programmers find that charging users money for *software services and support* (rather than for the software itself) is more lucrative. This way, their software remains free of charge, and they make money helping others install, use, and troubleshoot it.
- While some open source software may be free of charge, skill in programming and troubleshooting open source software can be quite valuable. Many employers specifically seek to hire programmers with experience working on open source software.

What is Free Software?

- **Free software means the users have the freedom to run, copy, distribute, study, change and improve the software.**
- Free software is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech”, not as in “free pizza”.
- More precisely, free software means users of a program have the four essential freedoms:
- The freedom to run the program as you wish, for any purpose.
- The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor.
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this.

Difference Between Windows and Linux

- The Windows is a commercial operating system, which means that the operating system and its corresponding programs must be purchased, in order to be used. The combined cost of the operating system and its corresponding programs can reach up to enormous highs while Linux is an open source operating system belongs to the GNU Public License. This ensures that users can get free access to the code to the very kernel that serves as the foundation of the Linux operating system.
- Not only that, users can even alter the kernel in order to make changes to the operating system itself. These changes can include preference changes for usability or the operating system can be changed so much that it counts as a completely new Linux distribution. this means that Windows costs a lot of money while, Linux is free.
- Windows is a graphical interface operating system developed, marketed, and sold by Microsoft.
- Linux, on the hand, technically is a kernel. A kernel is the central component of many operating systems.

- Linux is customizable whereas Windows is not.
- Linux has a very good security as you can log on to Linux only with a user id and password.
- Linux can boot from either a primary partition or a logical partition inside an extended partition, whereas Windows must boot from a primary partition.
- Linux can be booted from any hard disk in the computer, whereas Windows must be booted from the first hard disk.
- Linux uses a normal forward slash, whereas Windows separates directories with a back slash.
- Linux file names are case sensitive, whereas Windows file names are not case sensitive.
- Both, Linux and Windows support the concept of hidden files.

GNOME & KDE

- **GNOME**, **KDE** are the most popular desktop environments for Linux. Most people end up using the default desktop environment that ships with their preferred Linux distribution. But experienced users prefer one over the year for either usability, performance, design or customizability considerations.
- GNOME began as a project to develop a free and open-source desktop environment and corresponding applications in August 1997. Its design philosophy can be best described as streamlined and easy to use.
- KDE community started in October 1996. Its design philosophy is dedicated to functionality and expansion of its features.
- Most Linux distributions not only support both GNOME and KDE but also allow users to set either as their default.
- The second most popular Linux distribution — Linux Mint — offers different versions with different default desktop environments. While KDE is one of them; GNOME is not.

GNOME versus KDE comparison chart		
	GNOME	KDE
		
Introduction	GNOME (GNU Network Object Model Environment) is a desktop environment—a graphical user interface that runs on top of a computer operating system—composed entirely of free and open source software.	KDE is a desktop environment for an integrated set of cross-platform applications designed to run on Linux, FreeBSD, Microsoft Windows, Solaris and Mac OS, designed by the KDE Community.
Focus	Free software	Free software
Website	GNOME.org	kde.org
Developer	The GNOME Project	KDE
Founded	1999	1996

GNOME versus KDE comparison chart		
	GNOME	KDE
Minimum System Requirements	700 Mhz CPU, 768 MB RAM	1 Ghz CPU, 615 MB RAM
Design Goals	A top-to-bottom free desktop environment designed for simplicity, accessibility, and ease of internationalization and localization.	A one-stop, integrated desktop environment; carrying out day-to-day tasks without reliance on command-line interface.
Motto	An intuitive and attractive desktop for users	Experience Freedom!
Toolkit	GTK+	Qt
Founder(s)	Miguel de Icaza and Federico Mena	Matthias Ettrich
Products	GTK Apps	KDE Software Compilation, Calligra Suite, KDevelop, Amarok, etc

Features	GNOME	KDE
Development Priorities	free and open-source software, focused from its inception on freedom, accessibility, internationalization and localization, developer friendliness, organization, and support.	focuses on configurability and an attractive graphical user interface. These goals make it one of the most aesthetically pleasing, consistent, and integrated Linux environments with a high capacity for user customization.
User Experience	GNOME utilizes a top panel containing an activities button, clock, system status area, and user menu. Allows for quick access to and switching between open windows and applications. GNOME strives to utilize as few system resources as possible and offers a simple-to-use interface that may be more friendly to novice Linux users. While GNOME also offers advanced setting allowing for customization of its environment, experienced users may find its interface somewhat limiting.	KDE's focus on configurability and versatility comes with more complexity relative to GNOME. Its focus on presentation provides one of the most aesthetically pleasing desktop environments available, with menus set up very similar to Windows and a wide variety of configuration options built in. Navigating those options, however, can be challenging and may require a learning curve.

Features	GNOME	KDE
Appearance	On default settings, GNOME positions a toolbar at the top and a pop-out dock featuring large application icons on the left side of the screen that appears when the mouse hovers over its location. Its initial color scheme emphasizes dark grey, orange, and purple.	KDE's default settings keep things simple with one toolbar at the bottom of the screen and a single main menu. Its initial color scheme favors blue and grey.
	Both GNOME and KDE offer a wide range of user options for customizing the appearance and layout of their interfaces. The key difference lies in the tendency of GNOME to bury customization options under several layers of interface lists under the System Settings control panel, a design choice meant to streamline the basic user experience. On the other hand, KDE provides upfront shortcuts to many configuration options, and allows for fine tweaking the size of panels and removing or reorganizing preset applets.	

Features	GNOME	KDE
CPU/RAM Usage and User Experience	GNOME is generally regarded as streamlined and less resource-intensive than KDE. Interestingly, while GNOME's minimum system requirements are less demanding as far as CPU speed is concerned (700 Mhz, vs KDE's 1 Ghz requirement), KDE actually requires less minimum RAM (615 MB vs GNOME's 768 MB). On the other hand, on their default settings, KDE offers a more user-friendly environment, especially for those used to Windows as an operating system. For example, GNOME no longer offers users a true "minimize" option for open panels – something that Windows-acclimated users will likely miss – while KDE does.	
Development and Maintenance	GNOME operates on a 6-month release cycle that sees a new version released twice each year. KDE utilizes a dual release schedule that differentiates between major and minor releases; major releases generally take around 5 months from announcement to release, while minor ones take roughly 2 months. GNOME's development structure lends to better implementation of smaller, incremental changes. On the other hand, KDE's development cycle is suited for bigger, sweeping changes, such as the major design overhaul implemented between versions 3 and 4.	

What is GNU?

- GNU is an operating system that is free software—that is, it respects users' freedom. The development of GNU made it possible to use a computer without software that would trample your freedom.
- **What is the Free Software Movement?**
- The free software movement campaigns to win for the users of computing the freedom that comes from free software. Free software puts its users in control of their own computing. Non-free software puts its users under the power of the software's developer.
- GNU is a Unix-like operating system. That means it is a collection of many programs: applications, libraries, developer tools, even games. The development of GNU, started in January 1984, is known as the GNU Project. Many of the programs in GNU are released under the auspices of the GNU Project; those we call GNU packages.

- The name “GNU” is a recursive acronym for “GNU's Not Unix.” “GNU” is pronounced *g'noo*, as one syllable, like saying “grew” but replacing the *r* with *n*.
- The program in a Unix-like system that allocates machine resources and talks to the hardware is called the “kernel”. GNU is typically used with a kernel called Linux. This combination is the **GNU/Linux operating system**. GNU/Linux is used by millions, though many call it “Linux” by mistake.
- GNU's own kernel, The Hurd, was started in 1990 (before Linux was started). Volunteers continue developing the Hurd because it is an interesting technical project.

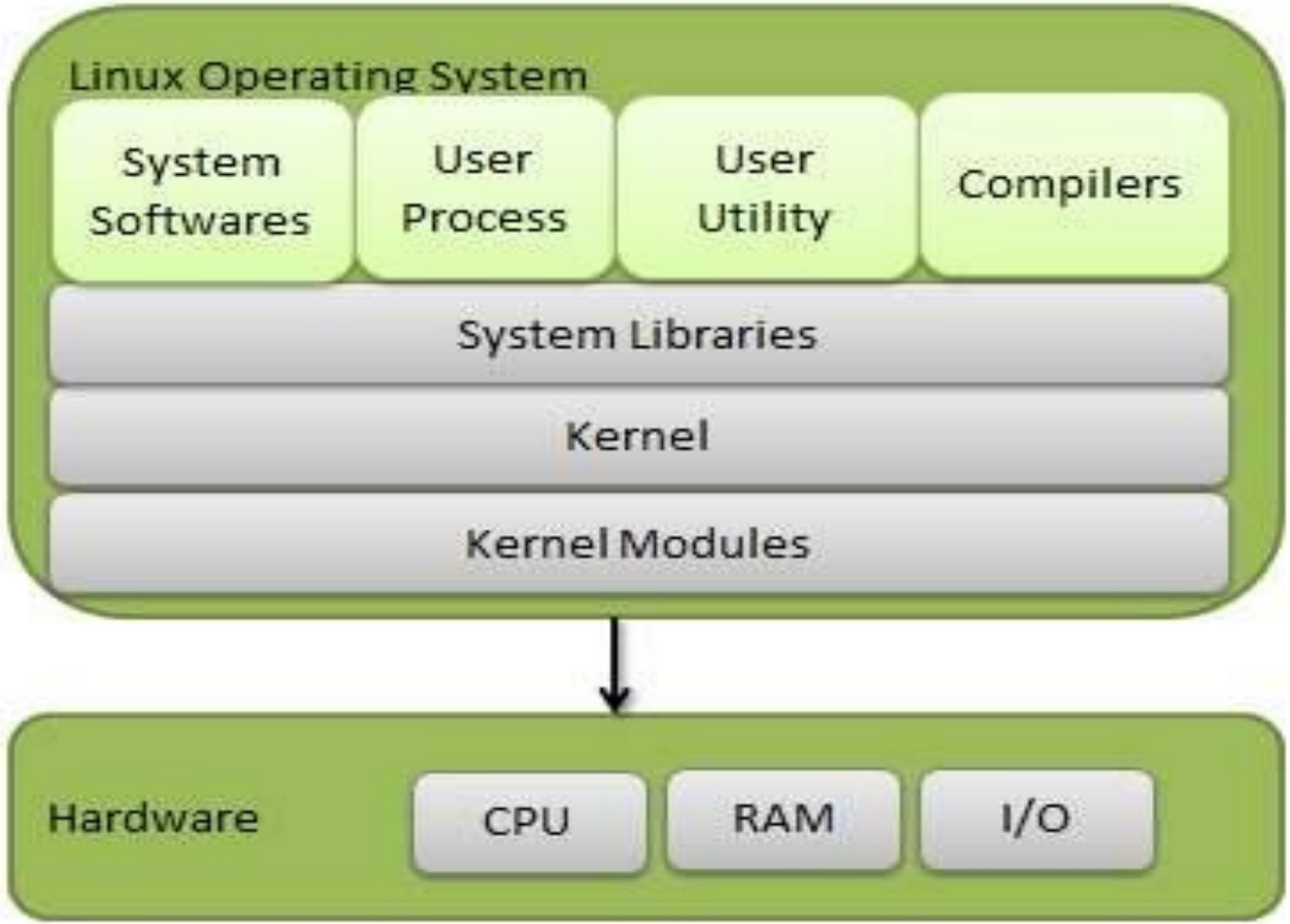
X Window System (commonly referred to as **X** or X11)

- The **X Window System** (commonly referred to as **X** or X11) is a network-transparent graphical windowing **system** based on a client/server model. Primarily used on Unix and Unix-like **systems** such as **Linux**, versions of **X** are also available for many other operating **systems.** X Window is primarily used in networks of interconnected mainframes, minicomputers, and workstations. It is also used on the X terminal, which is essentially a workstation with display management capabilities but without its own applications. (The X terminal can be seen as a predecessor of the network PC or thin client computer.)
- The X Window System is a graphical windowing system that was developed at MIT in 1984. X was developed as part of Project Athena, a cooperative effort between MIT, IBM, and Digital Equipment Corporation to develop a network of heterogeneous engineering terminals that could be used for teaching purposes. The current version, X11, was released in 1987 and is now up to X11 release 6, known as X11R6.

- One reason X has had such staying power is that from the beginning it incorporated many of the windowing capabilities that we now take for granted. These capabilities include network transparency, graphical capability, the use of a mouse, and the ability to link together a heterogeneous network of workstations from different vendors.
- In addition, X was intentionally designed to provide the low-level mechanism for managing the graphics display, but not to have any control over what is displayed. This means that X has never been locked into a single way of doing things; instead, it has the flexibility to be used in many different ways. Both the simplest window manager and the most complex desktop environment can, and do, use the X Window System to manage the display.
- The responsibility and stewardship of X is currently in the hands of the X.Org Foundation; the X.Org implementation of the X Window System is included with most of the major Linux and free Unix distributions. Until 2004, the standard version of X was XFree86, developed by the XFree86 Project. New licensing restrictions imposed by XFree86 led to the switch to X.org.

Components of Linux System

- Linux Operating System has primarily three components
- **Kernel** – Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
- **System Library** – System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries implement most of the functionalities of the operating system and do not requires kernel module's code access rights.
- **System Utility** – System Utility programs are responsible to do specialized, individual level tasks.


Linux Operating System
System Softwares
User Process
User Utility
Compilers
System Libraries
Kernel
Kernel Modules
Hardware
CPU
RAM
I/O

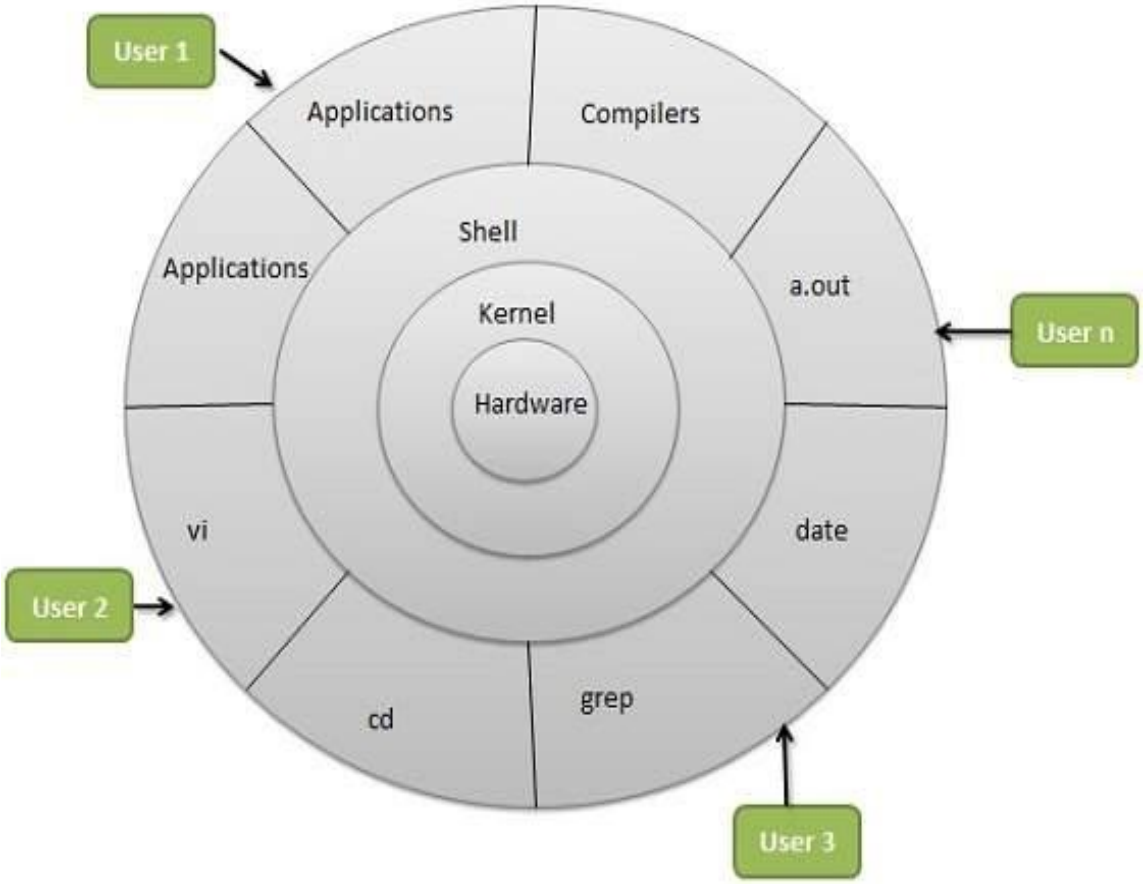
Kernel Mode vs User Mode

- Kernel component code executes in a special privileged mode called **kernel mode** with full access to all resources of the computer. This code represents a single process, executes in single address space and do not require any context switch and hence is very efficient and fast. Kernel runs each processes and provides system services to processes, provides protected access to hardware to processes.
- Support code which is not required to run in kernel mode is in System Library. User programs and other system programs works in **User Mode** which has no access to system hardware and kernel code. User programs/ utilities use System libraries to access Kernel functions to get system's low level tasks.

Basic Features of Linux

- **Portable** – Means software can works on different types of h/w in same way. Linux kernel & app programs supports their installation on any kind of h/w platform.
- **Open Source** – Source code is freely available & is community based development project. Multiple teams work in collaboration to enhance the capability of Linux operating system and it is continuously evolving.
- **Multi-User** – Multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.
- **Multiprogramming** – Multiple applications can run at same time.
- **Hierarchical File System** – It provides a standard file structure in which system files/ user files are arranged.
- **Shell** – Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs. etc.
- **Security** – Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

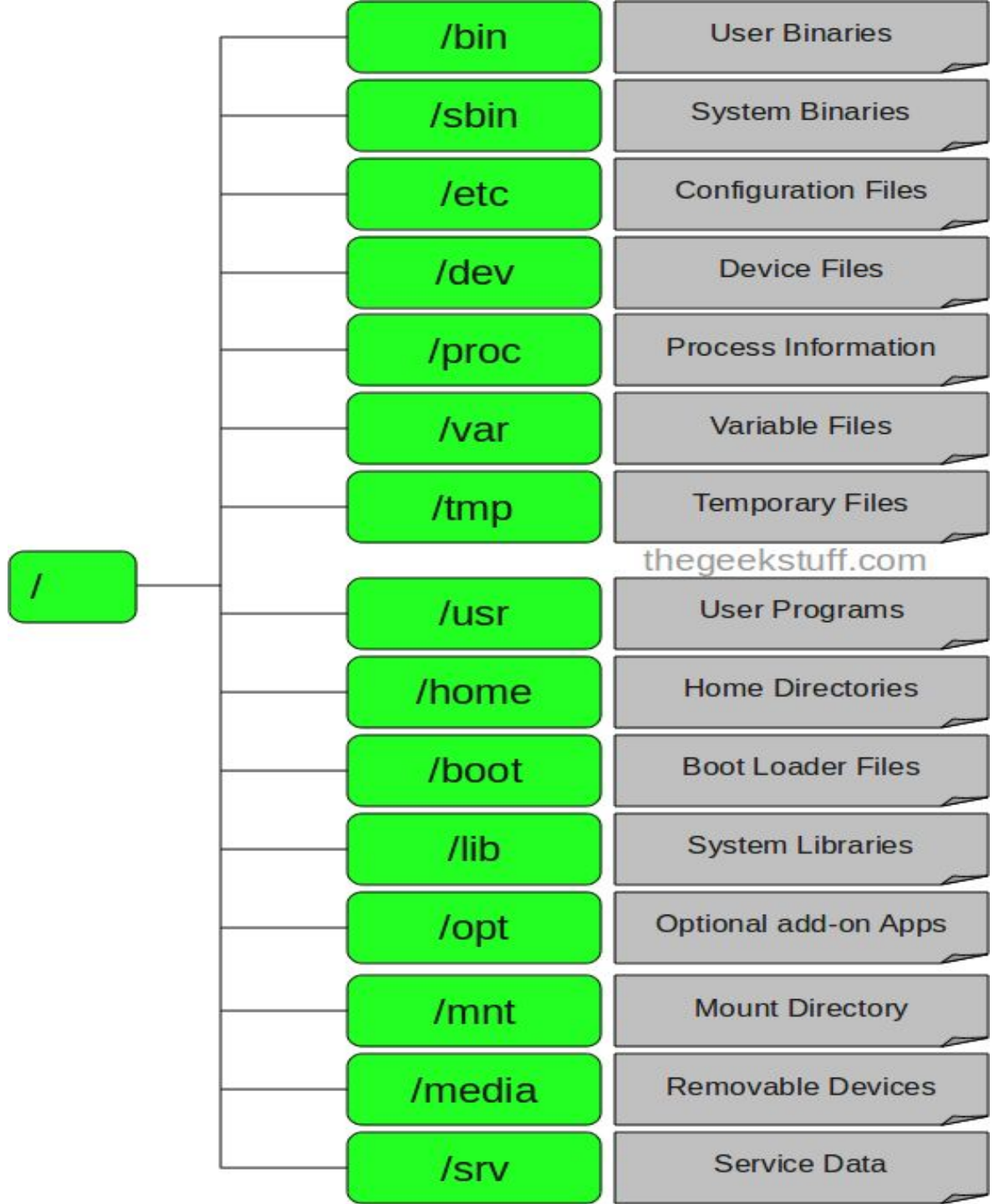
Architecture



- The architecture of a Linux System consists of the following layers −
- **Hardware layer** − Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
- **Kernel** − It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
- **Shell** − An interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.
- **Utilities** − Utility programs that provide the user most of the functionalities of an operating systems.

Linux Directory Structure (File System Structure) Explained with Examples

- Have you wondered why certain programs are located under /bin, or /sbin, or /usr/bin, or /usr/sbin?
- For example, less command is located under /usr/bin directory. Why not /bin, or /sbin, or /usr/sbin? What is the different between all these directories?
- In this article, let us review the Linux filesystem structures and understand the meaning of individual high-level directories.


/
/bin
User Binaries
/sbin
System Binaries
/etc
Configuration Files
/dev
Device Files
/proc
Process Information
/var
Variable Files
/tmp
Temporary Files
thegeekstuff.com
/usr
User Programs
/home
Home Directories
/boot
Boot Loader Files
/lib
System Libraries
/opt
Optional add-on Apps
/mnt
Mount Directory
/media
Removable Devices
/srv
Service Data

1. / – Root

- Every single file and directory starts from the root directory.
- Only root user has write privilege under this directory.
- Please note that /root is root user's home directory, which is not same as /.

2. /bin – User Binaries

- Contains binary executables.
- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here.
- For example: ps, ls, ping, grep, cp.

3. /sbin – System Binaries

- Just like /bin, /sbin also contains binary executables.
- But, the linux commands located under this directory are used typically by system aministrator, for system maintenance purpose.
- For example: iptables, reboot, fdisk, ifconfig, swapon

4. /etc – Configuration Files

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- For example: /etc/resolv.conf, /etc/logrotate.conf

5. /dev – Device Files

- Contains device files.
- These include terminal devices, usb, or any device attached to the system.
- For example: /dev/tty1, /dev/usbmon0

6. /proc – Process Information

- Contains information about system process.
- This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime

7. /var – Variable Files

- var stands for variable files.
- Content of the files that are expected to grow can be found under this directory.
- This includes — system log files (/var/log); packages and database files (/var/lib); emails (/var/mail); print queues (/var/spool); lock files (/var/lock); temp files needed across reboots (/var/tmp);

8. /tmp – Temporary Files

- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when system is rebooted.

9. /usr – User Programs

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can’t find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- /usr/sbin contains binary files for system administrators. If you can’t find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
- /usr/lib contains libraries for /usr/bin and /usr/sbin

- **10. /home – Home Directories**
  - Home directories for all users to store their personal files.
  - For example: /home/john, /home/nikita
- **11. /boot – Boot Loader Files**
  - Contains boot loader related files.
  - Kernel initrd, vmlinux, grub files are located under /boot
  - For example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic
- **12. /lib – System Libraries**
  - Contains library files that supports the binaries located under /bin and /sbin
  - Library filenames are either ld* or lib*.so.*
  - For example: ld-2.11.1.so, libncurses.so.5.7

13. /opt – Optional add-on Applications

- opt stands for optional.
- Contains add-on applications from individual vendors.
- add-on applications should be installed under either /opt/ or /opt/ sub-directory.

14. /mnt – Mount Directory

- Temporary mount directory where sysadmins can mount filesystems.

15. /media – Removable Media Devices

- Temporary mount directory for removable devices.
- For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer

16. /srv – Service Data

- srv stands for service.
- Contains server specific services related data.
- For example, /srv/cvs contains CVS related data.

- Example prompt when we login to a machine:
- **[surendra@linuxnix common]$**
- If you see above prompt it actually say that "surendra" is the user who logged in to a machine whose name is "linuxnix" to the present working directory "common" and he is a normal user($)
- How about superuser login prompt?
- **[root@linuxnix dev]#**

- If you see above prompt it actually say that "root" is the user who logged in to a machine whose name is "linuxnix" to the present working directory "dev" and he is a superuser user(#). So this prompt changes depending on the user name, his working directory and his privileged access.

LINUX COMMANDS

UNIX Commands

- A command is a program which interacts with the kernel to provide the environment and perform the functions called for by the user.
- A command can be: a built-in shell command; an executable shell file, known as a shell script; or a source compiled, object code file.
- The shell is a command line interpreter. The user interacts with the kernel through the shell. You can write ASCII (text) scripts to be acted upon by a shell.

UNIX Shell

- The shell sits between you and the operating system, acting as a command interpreter.
- It reads your terminal input and translates the commands into actions taken by the system. The shell is analogous to command.com in DOS.
- When you log into the system you are given a default shell.
- When the shell starts up it reads its startup files and may set environment variables, command search paths, and command aliases, and executes any commands specified in these files.

UNIX Shell

- The original shell was the Bourne shell, sh.
- Every Unix platform will either have the Bourne shell, or a Bourne compatible shell available.
- The default prompt for the Bourne shell is $ (or #, for the root user).
- Another popular shell is C Shell. The default prompt for the C shell is %.

UNIX Shell

- Numerous other shells are available from the network. Almost all of them are based on either sh or csh with extensions to provide job control to sh, allow in-line editing of commands, page through previously executed commands, provide command name completion and custom prompt, etc.
- Some of the more well known of these may be on your favorite Unix system: the Korn shell, ksh, by David Korn and the Bourne Again SHell, bash, from the Free Software Foundations GNU project, both based on sh, the T-C shell, tcsh, and the extended C shell, cshe, both based on csh.

Shell Programming

- You can write shell programs by creating scripts containing a series of shell commands.
- The first line of the script should start with #! which indicates to the kernel that the script is directly executable.
- You immediately follow this with the name of the shell, or program (spaces are allowed), to execute, using the full path name. So to set up a Bourne shell script the first line would be: #! /bin/sh

Shell Programming

- **The first line is followed by commands**
- **Within the scripts # indicates a comment from that point until the end of the line, with #! being a special case if found as the first characters of the file.**

```
#!/bin/bash
cd /tmp
mkdir t
```

- **You also need to specify that the script is executable by setting the proper bits on the file with chmod, e.g.:**

$ chmod +x shell_script

LINUX COMMANDS

- File Management and Viewing
- Filesystem Mangement
- Help, Job and Process Management
- Network Management
- System Management
- User Management
- Printing and Programming
- Document Preparation
- Miscellaneous

Outline

1. Overview of Unix System
2. Basic Commands
3. Relative & Absolute Path
4. Redirect, Append and Pipe
5. Permission
6. Process Management
7. Install Software
8. Text Editor
9. Foreground and Background Jobs

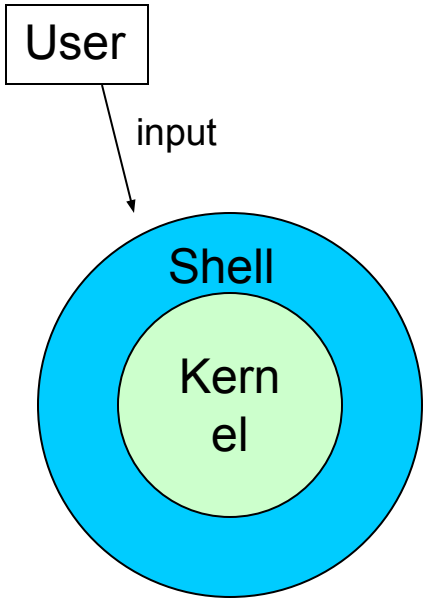
Overview of Unix System

Kernel & Shell

- Unix/Linux is operating system (OS).
- Unix system is described as kernel & shell.

- Kernel is a main program of Unix system. it controls hard wares, CPU, memory, hard disk, network card etc.

- Shell is an interface between user and kernel. Shell interprets your input as commands and pass them to kernel.



Unix Overview (cont.)

Multi-user & Multi-process

- Many people can use one machine at the same time.

File & Process

- Data, directory, process, hard disk etc (almost everything) are expressed as a file.
- Process is an running program identified by a unique id (PID).

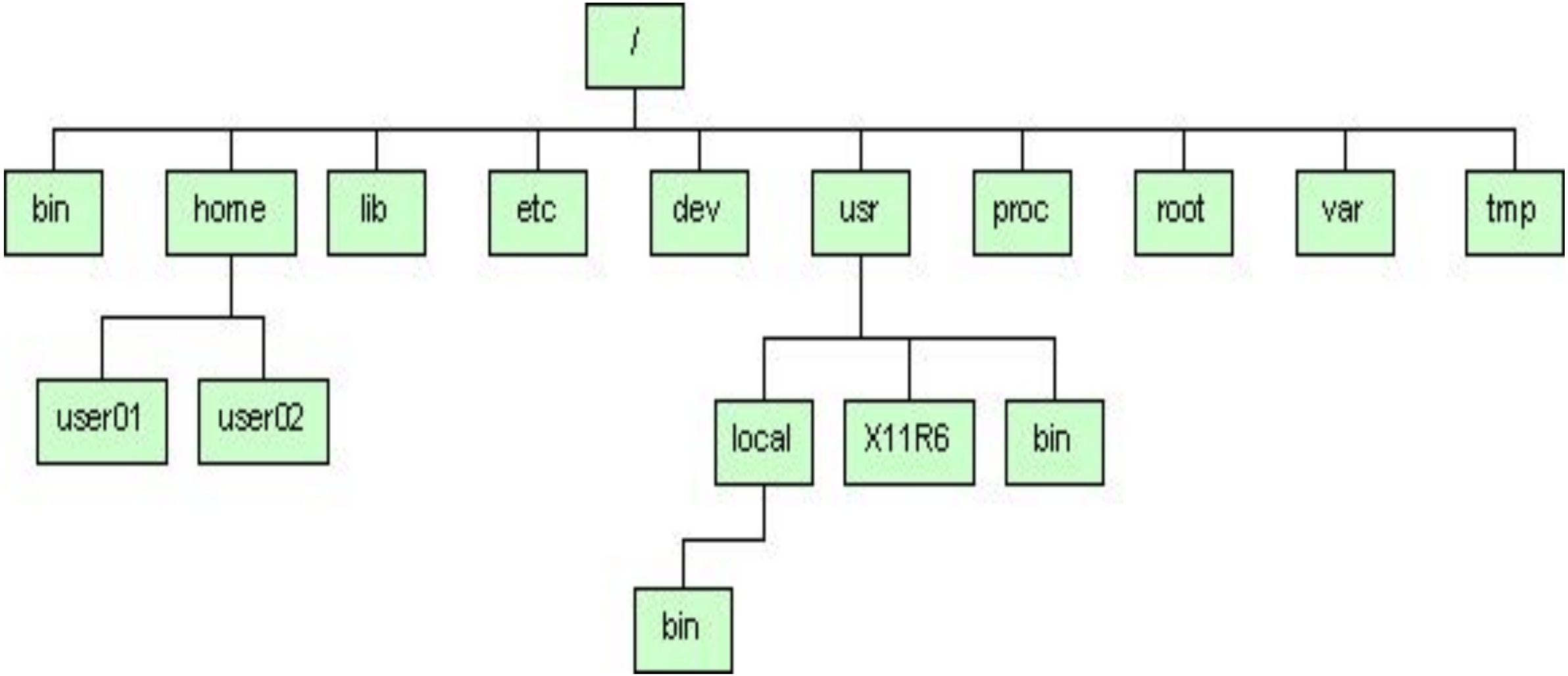
Unix Overview (cont.)

Directory Structure

- Files are put in a directory.
- All directories are in a hierarchical structure (tree structure).
- User can put and remove any directories on the tree.
- Top directory is “/”, which is called slash or root.
- Users have the own directory. (home directory)

Unix Overview (cont.)

Directory Structure



Command Structure

- Command <Options> <Arguments>
- Multiple commands separated by ; can be executed one after the other

Basic Commands

- **Command <Options> <Arguments>**
- **Multiple commands separated by ; can be executed one after the other**
- When you log on Linux machine, you will see,

[someone]$

- One command consists of three parts, i.e. command name, options, arguments.

Example)

[someone~]$ command-name optionA optionB argument1 argument2

Basic Commands

How to run commands

- Between command name, options and arguments, space is necessary.

- Opitions always start with "-"

- Example:

  cd ..

  ls –l .bashrc

  mv fileA fileB

Help Facilities for Commands

- To understand the working of the command and possible options use (man command)
- Using the GNU Info System (info, info command)
- Listing a Description of a Program (whatis command)
- Many tools have a long−style option, `−−help', that outputs usage information about the tool, including the options and arguments the tool takes. Ex: *whoami --help*

Pipes

- An important early development in Unix was the invention of "pipes," a way to pass the output of one tool to the input of another.

  eg. $ who | wc –l

  By combining these two tools, giving the wc command the output of who, you can build a new command to list the number of users currently on the system

Linux File Management and Viewing

File and Directory management

- **cd** Change the current directory. With no arguments "cd" changes to the users home directory. (cd <directory path>)
- **chmod** Change the file permissions.

  Ex: chmod 751 myfile : change the file permissions to rwx for owner, rx for group and x for others

  Ex: chmod go=+r myfile : Add read permission for the group and others (character meanings u-user, g-group, o-other, + add permission,-remove,r-read,w-write,x-exe)

  Ex: chmod +s myfile - Setuid bit on the file which allows the program to run with user or group privileges of the file.

Linux File Management and Viewing

There are three such special permissions within Linux. They are:

- ***setuid*** — used only for applications, this permission indicates that the application is to run as the owner of the file and not as the user executing the application. It is indicated by the character s in place of the x in the owner category. If the owner of the file does not have execute permissions, the S is capitalized to reflect this fact.
- ***setgid*** — used primarily for applications, this permission indicates that the application is to run as the group owning the file and not as the group of the user executing the application. The setgid permission is indicated by the character s in place of the x in the group category. If the group owner of the file or directory does not have execute permissions, the S is capitalized to reflect this fact.
- ***sticky bit*** — used primarily on directories, this bit dictates that a file created in the directory can be removed only by the user that created the file. It is indicated by the character t in place of the x in the everyone category. If the everyone category does not have execute permissions, the T is capitalized to reflect this fact.

Linux File Management and Viewing

- **chown** Change owner.

  Ex: chown <owner1> <filename> : Change ownership of a file to owner1.

- **chgrp** Change group.

  Ex: chgrp <group1> <filename> : Change group of a file to group1.

- **cp** Copy a file from one location to another.

  Ex: cp file1 file2 : Copy file1 to file2

  Ex: cp –R dir1 dir2 : Copy dir1 to dir2

- **md5sum** Prints the MD5 Checksum

Linux File Management and Viewing

- **ls** List contents of a directory.

  Ex: ls, ls –l , ls –al, ls –ld, ls –R

  (-rwxrwxr-x 1 juan juan 0 Sep 26 12:25 foo)

  |more will list page wise
- **mkdir** Make a directory.

  Ex: mkdir <directory name> : Makes a directory

  Ex *mkdir –p /www/chache/var/log* will create all the directories starting from www.
- **mv** Move or rename a file or directory.

  Ex: mv <source> <destination>

Linux File Management and Viewing

- **find** Find files (find <start directory> -name <file name> -print)

  Ex: *find /home –name readme -print*
  (Search for readme starting at home and output full path.)

  “/home" = Search starting at the home directory and proceed through all its subdirectories
  "-name readme" = Search for a file named readme
  "-print" = Output the full path to that file

- **locate** File locating program that uses the slocate database.
  Ex: locate –u to create the database,
  locate <file/directory> to find file/directory

Linux File Management and Viewing

- **pwd** Print or list the present working directory with full path.
- **rm** Delete files (Remove files). (rm –rf <directory/file>)
- **rmdir** Remove a directory. The directory must be empty. (rmdir <directory>)
- **touch** Change file timestamps to the current time. Make the file if it doesn't exist. (touch <filename>)
- **whereis** Locate the binary and man page files for a command. (whereis <program/command>)
- **which** Show full path of commands where given commands reside. (which <command>)

Linux File Management and Viewing

File viewing and editing

- **emacs** Full screen editor.
- **pico** Simple text editor.
- **vi** Editor with a command mode and text mode. Starts in command mode.
- **gedit** GUI Text Editor
- **tail** Look at the last 10 lines of a file.

  Ex: tail –f <filename> ,

  Ex: tail -100 <filename>
- **head** Look at the first 10 lines of a file. (head <filename>)

Linux File Management and Viewing

File compression, backing up and restoring

- **compress** Compress data.
- **uncompress** Expand data.
- **cpio** Can store files on tapes. to/from archives.
- **gzip** - zip a file to a gz file.
- **gunzip** - unzip a gz file.
- **tar** Archives files and directories. Can store files and directories on tapes.

  Ex: tar -zcvf <destination> <files/directories> - Archive copy groups of files. tar –zxvf <compressed file> to uncompress
- **zip** – Compresses a file to a .zip file.
- **unzip** – Uncompresses a file with .zip extension.

Linux File Management and Viewing

- **cat** View a file

  Ex: cat filename
- **cmp** Compare two files.
- **cut** Remove sections from each line of files.
- **diff** Show the differences between files.

  Ex: diff file1 file2 : Find differences between file1 & file2.
- **echo** Display a line of text.

Linux File Management and Viewing

- **grep** List all files with the specified expression. (*grep pattern <filename/directorypath>*)

  Ex: ls –l |grep sidbi : List all lines with a sidbi in them.

  Ex: grep " R " : Search for R with a space on each side
- **sleep** Delay for a specified amount of time.
- **sort** Sort a file alphabetically.
- **uniq** Remove duplicate lines from a sorted file.
- **wc** Count lines, words, characters in a file. (wc –c/w/l <filename>).

Unix Overview (cont.)

Important Directories

- /bin This contains files that are essential for correct operation of the system. These are available for use by all users.
- /home This is where user home directories are stored.
- /var This directory is used to store files which change frequently, and must be available to be written to.
- /etc Various system configuration files are stored here.

Unix Overview (cont.)

Important Directories

- /dev This contains various devices as files, e.g. hard disk, CD-ROM drive, etc.
- /sbin Binaries which are only expected to be used by the super user.
- /tmp Temporary files.

Unix Overview (cont.)

Normal user and Super user

- In Unix system, there is one special user for administrator, which can do anything.
- This special user is called root or superuser.

Case Sensitivity

- Unix is case-sensitive.
- MYFILE.doc, Myfile.doc, mYfiLe.Doc are different.

Online Manual

- Unix has well-written online manuals.

Basic Commands

Commands

- ls show files in current position
- cd change directory
- cp copy file or directory
- mv move file or directory
- rm remove file or directory
- pwd show current position
- mkdir create directory
- rmdir remove directory
- less, more, cat display file contents
- man display online manual

Basic Commands

Commands

- su switch user
- passwd change password
- useradd create new user account
- userdel delete user account
- mount mount file system
- umount unmount file system
- df show disk space usage
- shutdown reboot or turn off machine

Basic Commands

1. Type following command in your directory.

   ls

   ls –a

   ls –la

   ls -Fa

2. Make a directory

   mkdir linux

   pwd

   cd linux

   pwd

   cd

   pwd

   rmdir linux

3. In your home directory,

   ls .bash_profile

   cp .bash_profile sample.txt

   less sample.txt (note: to quit less, press "q")

   rm sample.txt

4. check disk space usage

   df

   df -h

Relative & Absolute Path

- Path means a position in the directory tree.
- To express a path, you can use relative path or absolute path.
- In relative path expression, the path is not defined uniquely, depends on your current path.
- In absolute path expression, the path is defined uniquely, does not depend on your current path.

Absolute Path

- Address from the root

  /home/linux/

  ~/linux

  ~: ~: Alt+N

- Similar to:

  Lausanne University/Lausanne/Canton de Vaud/ Switzerland/Europe/Earth/Solar System/

Relative Path

- Relative to your current location

  . : your current location

  .. : one directory above your current location

  pwd: gives you your current location

- Example

  ls ./linux : lists the content of the dir linux

  ls ../../ : lists everything that is two dir higer

- Similar to:

  Go Left/turn right/take the TSOL/go

Relative & Absolute Path

- Relative Path

```
pwd
cd .
pwd
cd ..
pwd
cd ..
pwd
cd
```

- Ablsoute Path

```
cd
mkdir mydir
pwd
cd /Users/invite
pwd
cd /Users
pwd
cd /
pwd
cd /Users/invite
cd ~/mydir
```

Redirect, Append and Pipe

Redirect and append

- Output of command is displayed on screen.
- Using “>”, you can redirect the output from screen to a file.
- Using “>>” you can append the output to the bottom of the file.

Pipe

- Some commands require input from a file or other commands.
- Using “|”, you can use output from other command as input to the command.
- On MacOSX, The Pipe sign: (**Shift+Alt+N: franc, Alt+7)**

Redirect, Append and Pipe

Commands

- head show first several lines and omit other lines.
- tail show last several lines and omit other lines.
- **grep XXX File** show lines matching pattern XXX in File

Redirect, Append and Pipe

- In home directory, type

  ls -1 > sample.txt

  less sample.txt

- Use redirect.

  head -3 sample.txt

  head -3 sample.txt > redirect.txt

- Use append.

  tail -3 sample.txt

  tail -3 sample.txt >> redirect.txt

  less redirect.txt

- Use pipe.

  less redirect.txt

  grep Desk redirect.txt

  grep –n Desk redirect.txt

  man grep

  tail redirect.txt | grep Desk

  rm sample.txt

  rm redirect.txt

Sorting

Commands

- sort Sorts using the first field of each line.
- -n Sorts considering the numeric value of the strings
- -k3 Sorts using the third field of each line
- -rnk3 Sorts in reverse order, using the numeric value of the third field

Redirect, Append and Pipe

- Identify the largest file in a directory:

```
ls –la /bin/ | sort –nk5 | tail -1
```

Permission

- All of files and directories have owner and permission.
- There are three types of permission, readable, writeable and executable.
- Permissions are given to three kinds of group. owner, group member and others.

Example:

```
ls -l .bash_profile
-rw-r--r--    1 cnotred    cnotred        191 Jan  4 13:11 .bash_profile
```

- r:readable, w:writable, x: executable

Chapter 16

- Administration Tasks

Figure 16-1 The User Manager window, Users tab

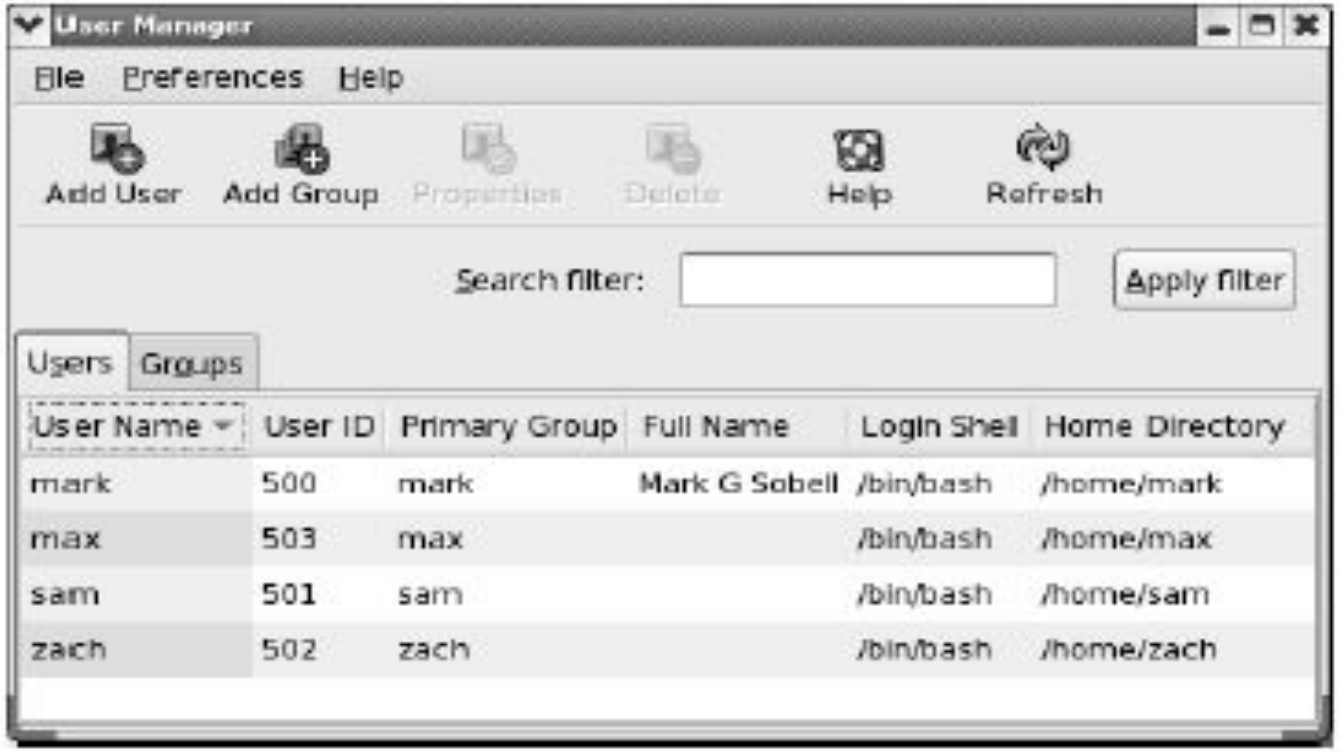
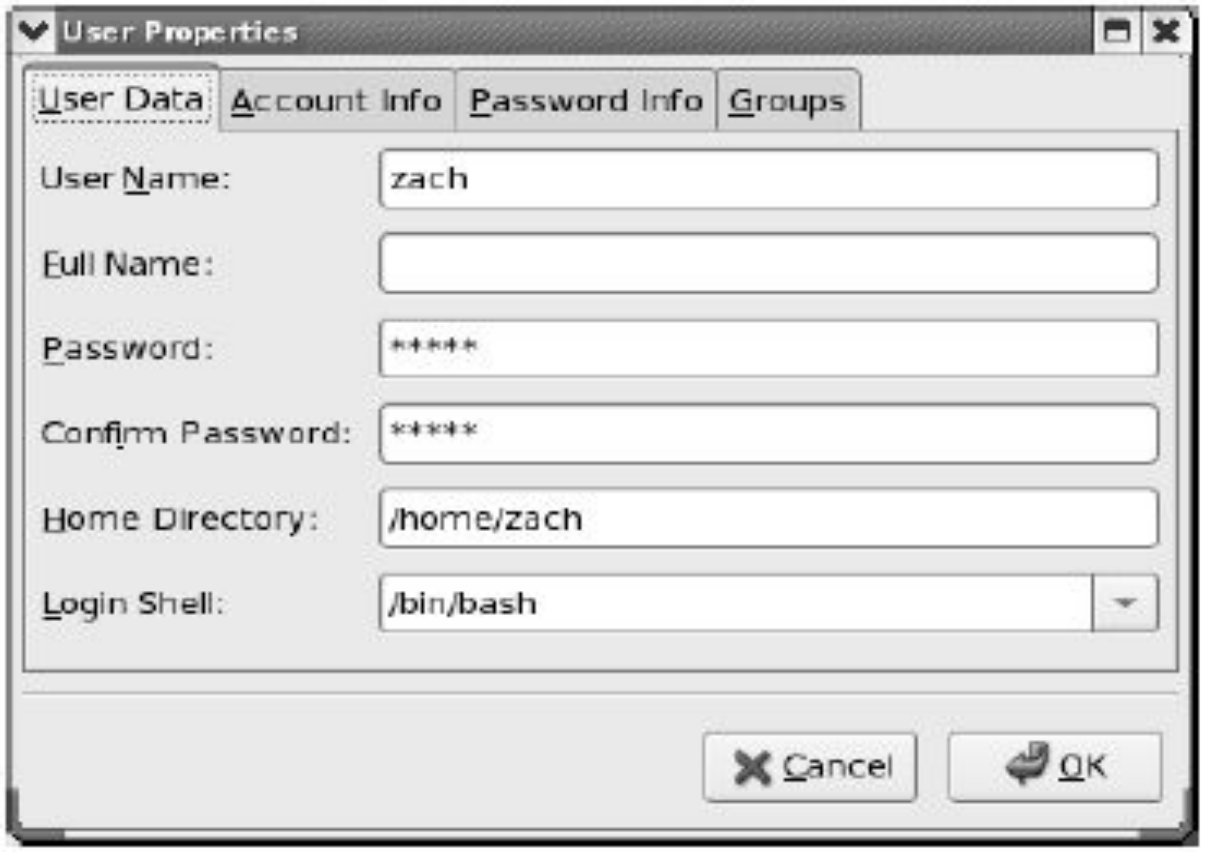


Figure 16-2 The User Properties window, User Data tab



Managing Users and Groups

- One of the key administrative tasks with Linux is managing users and groups.
- The primary reason for user accounts is to verify the identity of each individual using a computer system.
  - A secondary reason for user accounts is to permit the per-individual tailoring of resources and access privileges.
    - Resources can include files, directories, and devices. Controlling access to these resources is a primary task of an administrator

Groups tie together users that have a common purpose.

- an organization may have persons responsible for accounts payable and others responsible for payroll.
  - By placing the user accounts in an accounts payable group then common permissions can be given to all the members of that group.
  - Members of the accounts payable group would not have access to the information and resources of the payroll group. Users within the same group have the same read, write. or execute privileges of group resources.

etc

- Several files are used when creating users in Linux. The following are a few most commonly used.

  /etc/passwd

  /etc/shadow

  /etc/group

/etc/passwd

- The /etc/passwd file contains the user ID, and default home directory. Because this file is used by many tools it needs to be readable by any user.
- To view the /etc/passwd file use the less command.

  less /etc/passwd

  - The /etc/passwd file is a group of fields separated with a colon (**:**). They are username, password (shown as an x), numeric user ID, numeric group ID, full name, user's home directory, and user's shell account.

/etc/shadow

- The /etc/shadow file contains the encrypted passwords and other password information.
  - This file is viewable by the root user only.
- To view the /etc/shadow file use the following commands:

  su – root

  tail /etc/shadow
- The /etc/shadow file is a group of fields separated with a colon (**:**). They are:
  - Username
  - password (13 characters encrypted)
  - the number of days since the password was last changed
  - the number of days before the password may be changed
  - the number of days to warn a user of an expiring password
  - the number of days after a password expires that account is disabled
  - the number of days since an account has bee disabled
  - a reserved field for possible future use.

Adding a User 1

- Use the useradd command to add a user. The syntax of the command is displayed by typing the command alone.

  useradd

- The most basic command to add a user named John Smith with and user ID of jsmith is the following:

  useradd jsmith

- If you view /etc/passwd after the add you will see that jsmith has been added.

  jsmith:x:501:501::/home/jsmith:/bin/bash

  id**:**password(shadowed)**:**Full Name**:**homeDir**:**shell

  - The full name can be added with the –c option. **useradd –c "John Smith" jsmith**

Adding a User 2

- When a user is added the home directory is not created automatically. It is assigned but not created until the user logs in the first time. To force the home directory to be created use the –m option. The following command will create the jsmith user and create the associated home directory:

  useradd –c "John Smith" –m jsmith

- View the jsmith user home directory with the command:

  ls /home/directory

- Any files and directories that are in the /etc/skel directory are automatically copied into the newly created home directory. If /etc/skel has no files or directories (the default) then nothing is created.

- The /etc/skel directory on this Linux system has a Documents directory and a Welcome file.

Adding a User 3

- A look at the /etc/shadow file shows that jsmith was added but a password has not been assigned. The command to add a password is passwd. As root you can change the password for jsmith with the following command:

  passwd jsmith

- You will be asked to type the password and then repeat it to ensure it was typed correctly. The root user can change any user password. A user can change their own password by typing passwd without a user ID.

- passwd has options to configure the minimum password lifetime, maximum password lifetime, and other options.

Deleting a User

- Deleting a user is done as the root user with the userdel command. To delete the user jsmith use the command:

  userdel jsmith

- Userdel has one option, -r, which removes the user's home directory as well as the account. To remove jsmith and the associated home directory, /home/jsmith, use the following command:

  userdel –r jsmith

- Once a user's home directory is removed it would have to be restored from backup to recover it.
- There may be orphaned files – files that are not associated by a valid user – when you delete a user
  - Example: jsmith was storing some files in /tmp. Those files are not deleted when the user jsmith is removed. Those files in /tmp are now orphaned.

Adding a Group

- The mail file for groups is the /etc/group file. The file is a group of fileds that describe the group and who is a member of it. The fields in order are the group name, password (not used), numeric ID, and then a list of members separated by a comma.

- A group is used to assign rights and permissions to users. For example, if you have several files that should be made available to users in the Accounts Payable department you could create an Accounts Payable group then add users to the group. Once the group is populated then the permissions of the file or directory can be changed to allow access by the group.

- To add a group use the groupadd command.

  groupadd AccountsPayable

Assigning Users to a Group

- The usermod command is used to add users to a group.
  - It also has options to change the home directory, change the shell, and other values.
- To use usermod to add users jsmith and jdoe to the AccountsPayable group type the following:

  usermod –G AccountsPayable jsmith

  usermod –G AccountsPayable jdoe

Changing Group Permissions

- Groups can be used to assign privileges to resources, such as a directory.
  - to change the group to AccountsPayable. This is accomplished with the following command:

    chgrp –R AccountsPayable AP
  - Where –R causes it to recursively affect files and directories within the AP directory, AccountsPayable is the new group, and AP is the name of the directory.
  - Permissions for group are then changed to give full read/write/execute access with the chmod command.

    chmod 775 AP

Permission

Command

- chmod change file mode, add or remove permission
- chown change owner of the file

Example)

chmod a+w filename add writable permission to all users

chmod o-x filename remove executable permission from others

chmod a+x Gives permission to the usser to execute a file

- u: user (owner), g: group, o: others a: all

Permission

- Check permission

  ls –l .bash_profile

  cp .bash_profile sample.txt

  ls –l sample.txt

- Remove readable permission from all.

  chmod a-r sample.txt

  ls –l sample.txt

  less sample.txt

- Add readable & writable premissions to file owner.

  chmod u+rw sample.txt

  ls –l sample.txt

  less sample.txt

  rm sample.txt

Process Management

- Process is a unit of running program.

- Each process has some information, like process ID, owner, priority, etc.

Example) Output of “top” command

```
  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
12035 nomura    15   0  1080 1080   840 R     0.3  0.2   0:00 top
    1 root      15   0   472  436   420 S     0.0  0.0   0:04 init
    2 root      15   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root      15   0     0    0     0 SW    0.0  0.0   0:00 kapmd
    4 root      34  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    5 root      15   0     0    0     0 SW    0.0  0.0   0:59 kswapd
    6 root      15   0     0    0     0 SW    0.0  0.0   0:00 bdflush
```

Process Management

Commands

- kill Stop a program. The program is specified by process ID.
- killall Stop a program. The program is specified by command name.
- ps Show process status
- top Show system usage statistics

Process Management

- Check your process.

```
ps
ps –u
```

- Check process of all users.

```
top (To quit top, press “q”)
ps –e
ps –ef
```

- Find your process.

```
ps –ef | grep cnotred
```

Install Software

- Unix system has a "*de facto* standard" way to install a software.

  configure, make & make install

- Typical software installation procedure as following.

1. Download source code. Usually, it's archived with tar command and compressed with gzip command.
2. configure command creates Makefile automatically which is used to compile the source.
3. Program compilation is written in Makefile.

Install Software

Commands

- **gzip compress a file**
- **gunzip uncompress a file**
- **tar archive or expand files**
- **configure create Makefile**
- **make compile & install software**

Install Software

Example: parallel programming library installation

gunzip software.tar.gz

tar –xvf software.tar

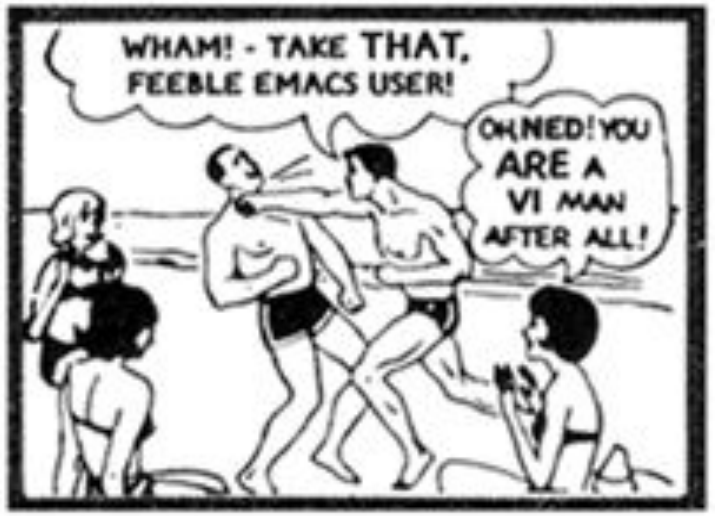
cd software

./install OR make all OR …

Program Development Tools

Programming Editors

Many good programming editors exist, but...



We will focus on *vi*...why?...it's the one I know!

vi (pronounced "*vee eye*", not "*vye*")

- -*vi* is available on all *nix systems.
- -*vi* is fast and stays out of your way
  - -no menus, short commands, no"mouseing"
- -high beginning threshold (~12 commands)
- -master the first dozen, then add to your repertoir
- -*vi* help sites
  - -http://staff.washington.edu/rells/R110/help_vi.html
  - -http://www.vmunix.com/~gabor/vi.html
  - -http://docs.freebsd.org/44doc/usd/12.vi/paper.html (Joy and Horton)
  - -http://thomer.com/vi/vi.html (fun vi site)

When I say `vi`, I really mean `vim`. *vi* must be aliased to `vim` on Ubuntu.

```
which vi <enter>
vi:  aliased to vim
```

Program Development Tools

Vi commands

To begin editing a file:

`vi`	opens vi
`vi file_name`	opens vi on `file_name`
`view filename`	opens vi in read-only mode on `file_name`

To close vi:

`:q`	quit (it will query you if changes were made)
`:q!`	quit without saving
`:x`	save and exit
`<shift>zz`	save and exit

Writing a file:

`:w`	write out current file
`:w new_name`	write out current file as `new_name`
`:wq`	write file and quit (like “x” or “<shift>zz)

Read in a file:

`:r file_name`	Read in `file_name` into current one after the cursor

Edit a new file:

`:e file_name`	Begin editing new file (need to save old one)

Program Development Tools

Vi commands – Modes

Input Mode:

Insert text:

`i`	insert text just prior to cursor

Append text:

`a`	append text just after cursor

Open line:

`o`	open a new line below the current one
`O`	open a new line above the current one

Replace:

`r`	replace the character under cursor
`R`	replace continuously

Exit input mode:

`<esc>`

Program Development Tools

Vi commands – Modes

Command mode:

In command mode we do everything else.....

-moving around in the file
-search and replace
-yank (copy) and put (paste)
-delete
-split screens

Command mode entered from input mode by <esc>
You enter vi in the command mode.

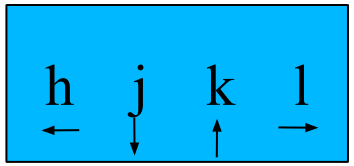
If you get totally confused and are afraid you have messed up the file, hit:
`<esc>`
then
`:q!`
You will exit insert mode and close the file leaving it unchanged.

Program Development Tools

Vi commands

Moving around in the file:

`h`	move one character left
`l`	move one character right
`k`	move one line up
`j`	move one line down
`<ctrl>u`	move up one page
`<ctrl>d`	move down one page
`w`	move forward 1 word (cursor at first character)
`b`	move back 1 word (cursor at first character)
`e`	move forwards by last letter in word
`$`	move to end of line
`0`	move to beginning of line
`gg`	move to top of page, column 1 (*home*)
`G`	move to last line, column 1
`10G`	move to line 10
`:n`	go to line n
`<ctrl>G`	tell me where I am (file, line, column)



Program Development Tools

Vi commands

Yank/paste, change, delete, repeat:

`yy`	yank current line
`p`	paste a line
`8yy`	yank next 8 lines (a subsequent paste will paste all 8 lines)
`x`	delete character (`4x` = delete next 4 characters)
`cw`	change word
`dw`	delete word (`8dw` = delete next 8 words)
`dd`	delete current line (line goes in paste buffer)
`10dd`	delete next 10 lines
`J`	join next to line to current
`u`	undo (multi-level repeat) (`4u` = undo last 10 commands)
`<ctrl>r`	redo last command
`.`	repeat last command (a *killer* command!)

Program Development Tools

Vi commands

Search and Replace:

`/pattern`	search for `pattern` (forwards)
`?pattern`	search for `pattern` (backwards)
`n`	repeat last search
`:%s/old/new/g`	replace every occurrence of `old` with `new` in entire file
`%`	show the matching “()”, “[]” or “{}”
*	find next occurance of word under cursor
#	find previous occurance of word under cursor

Program Development Tools

Vi commands

Multiple screens:

`:sp new_file`	open split screen with `new_file` displayed
`:vsp new_file`	open vertically split screen with `new_file` displayed
`<ctrl>ww`	move between screens
`<ctrl>wn`	split existing window
`<ctrl>wv`	split existing window vertically

Program Development Tools

Vi commands

Shell and make commands

`:make`	compile file specified in Makefile
`:make program`	compile, link, and program target
`:make clean`	clean up compiler generated files
`:! <cmd>`	execute shell command without leaving vi, e.g.
`:! ls`	list contents of current directory
`:! date`	what is date and time

Text Editor

pico

- **Programs & configuration files are text file.**
- **There are two popular text editors, vi and Emacs.**
- **Although they are very powerful and useful, it is also true that they are complicated for beginners and difficult to learn.**
- **pico is an easy and simple alternative.**

Text Editor

Commands

- Arrow-keys Move cursor
- CTRL+a Move to the beginning of the current line.
- CTRL+e Move to the end of the current line.
- CTRL+v Move forward one page.
- CTRL+y Move backward one page.
- CTRL+w Search for text.
- CTRL+d Delete the current character.
- CTRL+k Remove (cut) current line or selected text.
- CTRL+u Paste (uncut) last cut text at the cursor position.
- CTRL+o Save (output) the file.
- CTRL+x Exit Pico, saving the file.

- Autre: xemacs, emacs

Text Editor

- Create the file Hello
  pico hello.pl

- Write hello.pl as follows.

```
#!/usr/bin/perl
print “Hello World\n”;
```

- Make il executable
  chmod u+x hello.pl

- Run it!
  ./hello.pl

Foreground and Background

- Running job has two modes, “foreground” and “background”

- If program is running as “background”,
  the program keeps running even after your session was closed

- If program is running as “foreground”,
  Ctrl-C stop program
  Ctrl-Z let program background

Foreground and Background

- To run programs in background mode, use “&”

  [nomura@ssc-1]$ *command* &

- To get background job back into foreground mode, use “fg” command.

  [nomura@ssc-1]$ fg

Remote Login & File Transfer

- rshd, telnetd, ftpd, sshd are server program and provide similar services, remote login & file transfer.
- The major difference is security level.
  rshd < telnetd + ftpd < sshd

Commands

Client

- rsh & rcp
- telnet & ftp
- ssh & scp

Server

- rshd
- telnetd & ftpd
- sshd

Remote Login & File Transfer

Remote login & file transfer system are based on server and client model. client program on your machine ask sever program certain service remote machine.

For example, telnet server provides remote login service. ftp server provides file transfer service.

Sample client programs;

WS FTP	FTP client
Internet Exploror	HTTP client
Eudora	POP, SMTP client

LINUX PROCESS CONTROL

What is a Process?

- An instance of a program is called a Process. In simple terms, any command that you give to your Linux machine starts a new process.
- Having multiple processes for the same program is possible.
- Types of Processes:
- Foreground Processes: They run on the screen and need input from the user. For example Office Programs
- Background Processes: They run in the background and usually do not need user input. For example Antivirus.

Running a Foreground Process

- To start a foreground process, you can either run it from the dashboard, or you can run it from the terminal.
- When using the Terminal, you will have to wait, until the foreground process runs.

Running a Background process

- If you start a foreground program/process from the terminal, then you cannot work on the terminal, till the program is up and running.
- Particular, data-intensive tasks take lots of processing power and may even take hours to complete. You do not want your terminal to be held up for such a long time.
- To avoid such a situation, you can run the program and send it to the background so that terminal remains available to you. Let's learn how to do this -

Start the program and press ctrl+z

```
guru99@VirtualBox:~$ banshee
[Info  16:08:36.688] Running Banshee 2.2.1: [Ubuntu 11.
11-12-19 14:51:26 UTC]
^Z
[1]+  Stopped                 banshee
```

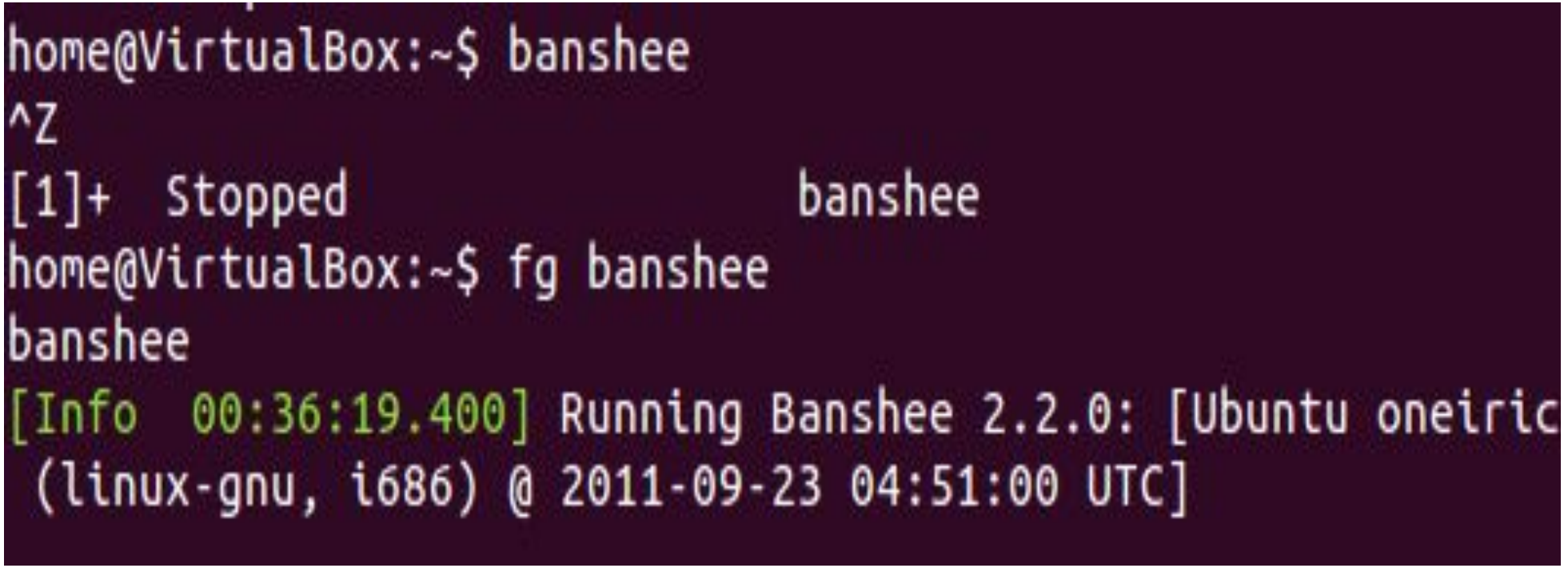
Type 'bg' to send the process to the background

```
guru99@VirtualBox:~$ bg
```

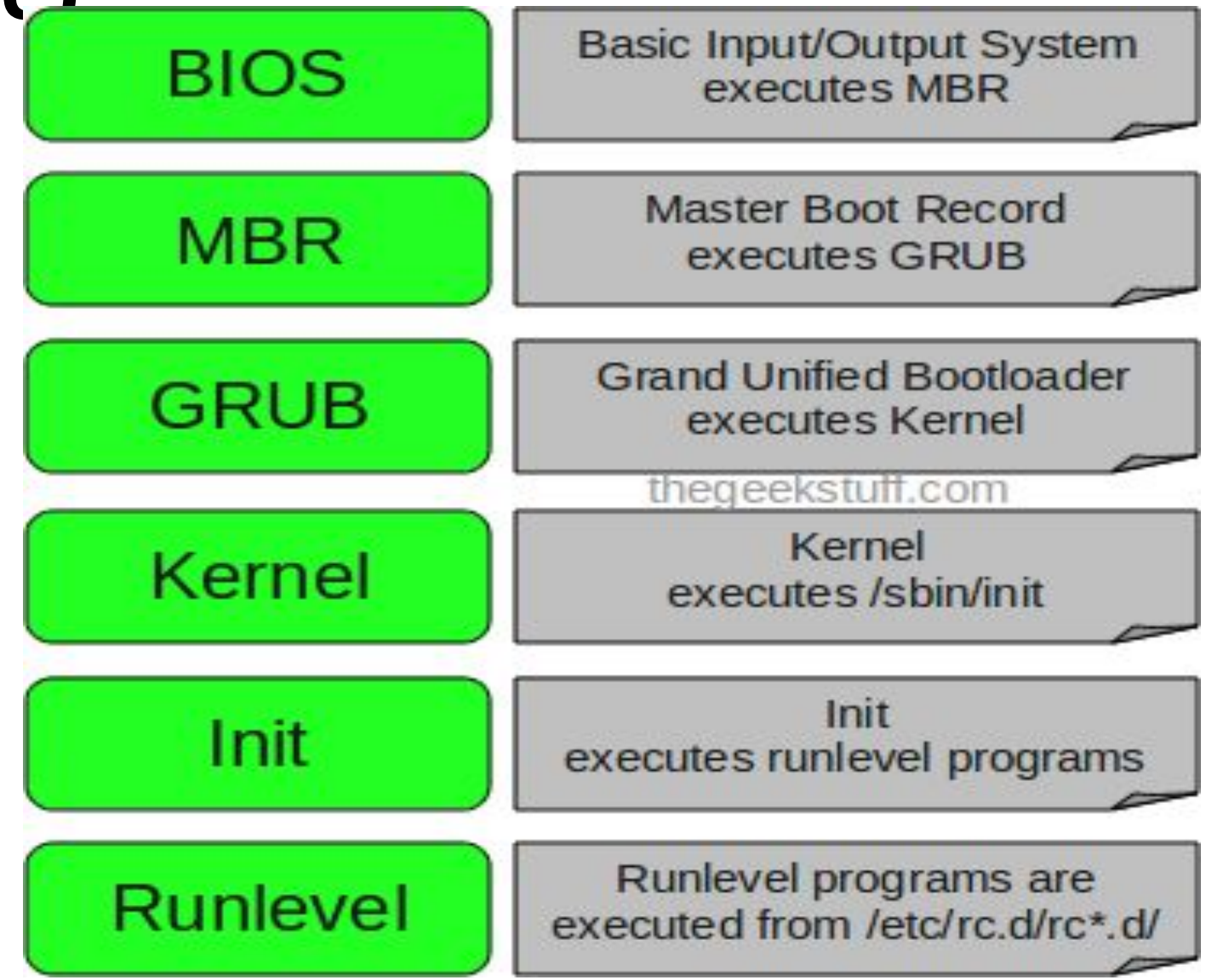
fg- You can use the command "fg" to continue a program which was stopped and bring it to the foreground.

- The simple syntax for this utility is: **fg jobnsme e.g.**

1. Launch 'banshee' music player
2. Stop it with the 'ctrl +z' command
3. Continue it with the 'fg' utility.



6 Stages of Linux Boot Process (Startup Sequence)



- **1. BIOS**
  - BIOS stands for Basic Input/Output System
  - Performs some system integrity checks
  - Searches, loads, and executes the boot loader program.
  - It looks for boot loader in floppy, cd-rom, or hard drive. You can press a key (typically F12 of F2, but it depends on your system) during the BIOS startup to change the boot sequence.
  - Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
  - So, in simple terms BIOS loads and executes the MBR boot loader.

- **2. MBR**
  - MBR stands for Master Boot Record.
  - It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
  - MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes.
  - It contains information about GRUB (or LILO in old systems). LILO (**Li**nux **Lo**ader)
  - So, in simple terms MBR loads and executes the GRUB boot loader.

- **3. GRUB**
  - GRUB stands for Grand Unified Bootloader.
  - If you have multiple kernel images installed on your system, you can choose which one to be executed.
  - GRUB displays a splash screen, waits for few seconds, if you don’t enter anything, it loads the default kernel image as specified in the grub configuration file.
  - GRUB has the knowledge of the filesystem (the older Linux loader LILO didn’t understand filesystem).
  - Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this).

The following is sample grub.conf

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-194.el5PAE)
          root (hd0,0)
          kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/
          initrd /boot/initrd-2.6.18-194.el5PAE.img
```

- As you notice from the above info, it contains kernel and initrd image.
- So, in simple terms GRUB just loads and executes Kernel and initrd images.

- **4. Kernel**
  - Mounts the root file system as specified in the “root=” in grub.conf
  - Kernel executes the /sbin/init program
  - Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a ‘ps -ef | grep init’ and check the pid.
  - initrd stands for Initial RAM Disk.
  - initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

•5. Init

- Looks at the /etc/inittab file to decide the Linux run level.
- Following are the available run levels
  - 0 – halt 1 – Single user mode
  - 2 – Multiuser, without NFS 3 – Full multiuser mode
  - 4 – unused 5 – X11
  - 6 – reboot
- Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.
- Execute ‘grep initdefault /etc/inittab’ on your system to identify the default run level
- If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
- Typically you would set the default run level to either 3 or 5.

- **6. Runlevel programs**
- When the Linux system is booting up, you might see various services getting started. For example, it might say “starting sendmail …. OK”. Those are the runlevel programs, executed from the run level directory as defined by your run level.
- Depending on your default init level setting, the system will execute the programs from one of the following directories.

  Run level 0 – /etc/rc.d/rc0.d/ Run level 1 – /etc/rc.d/rc1.d/

  Run level 2 – /etc/rc.d/rc2.d/ Run level 3 – /etc/rc.d/rc3.d/

  Run level 4 – /etc/rc.d/rc4.d/ Run level 5 – /etc/rc.d/rc5.d/

  Run level 6 – /etc/rc.d/rc6.d/
- Please note that there are also symbolic links available for these directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.

- Under the /etc/rc.d/rc*.d/ directories, you would see programs that start with S and K.
- Programs starts with S are used during startup. S for startup.
- Programs starts with K are used during shutdown. K for kill.
- There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.
- For example, S12syslog is to start the syslog deamon, which has the sequence number of 12. S80sendmail is to start the sendmail daemon, which has the sequence number of 80. So, syslog program will be started before sendmail.

Login Process in Linux

- When Linux booting process is completed it display the login prompt. There are different process running in Linux when any user login to their shell. these processes are as follows

Login process steps

1. Init starts getty process
2. getty process initiates login prompt on terminal
3. login command check user credentials from /etc/passwd
4. getty starts user shell process
5. shell reads the system wide files /etc/profile, /etc/bashrc
6. Shell reads user specific files .profile, .login
7. Now it reads shell specific configuration file .bashrc
8. Shell displays the default prompt

All these steps are explained in next slides in more detail.

1. **Init starts getty:** When Linux system boots it goes through various booting stages. In last stage it starts **init**, which reads a file called **inittab** which is located in **/etc**, where it find out in which run level it has to execute. Once init process completes run-level execution and executing commands in **/etc/rc.local**, it will start a process called **getty**. Getty is the process which will take care of complete login process.

2. **Getty shows login prompt: getty** is short for "**get terminal**". A getty program initiates login command, it opens the terminal device, initializes it, prints login: and waits for a user name to be entered.

3. **getty starts /etc/login:** Once user enters his login name getty starts **/ets/login**, this in-turn will prompt for user password. The password what user typed will be hidden and will not be shown on screen.

4. getty verifies the credential and starts users shell

In next stage **getty** checks the user credentials by verifying it with **/etc/passwd** and **/etc/shadow** file, if password matches then it initiates user properties gathering and starts users shell. If password doesn't match then getty terminates login process and re-initiates again with new login: prompt.

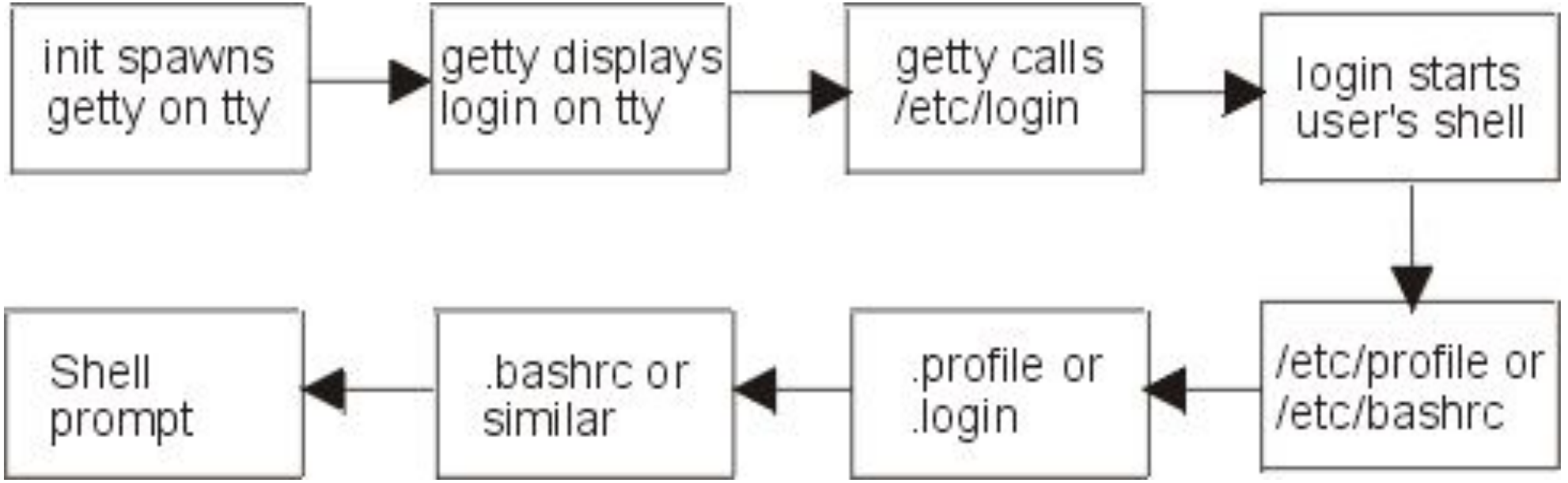
In next stage the getty process reads the user properties (username, UID, GID, home directory, user shell etc.) from **/etc/passwd** file. After that it reads **/etc/motd** file for displaying content banner message.

5. Shell reads system wide default files and specific default files

For getting shell related properties, user aliases and variables getty process reads appropriate system wide default files **/etc/profile** or **/etc/bashrc** . After the system wide default files are read the shell reads **user specific login files .profile** or **.login .**

6. **Shell specific file read:** At last stage it reads shell specific configuration files **(.bashrc, .bash_profile** etc. for **bash shell**) of that user which it gets on the users home directory.

7. **Shell Prompt:** When all startup files are read the shell displays the default prompt, normally PS1 prompt for user to execute their commands, here user can type any command to execute.



PROCESS MONITORING

- The Linux terminal has a number of useful commands that can display running processes, kill them, and change their priority level. This post lists the classic, traditional commands, as well as some more useful, modern ones.
- Many of the commands here perform a single function and can be combined — that’s the Unix philosophy of designing programs. Other programs, like htop, provide a friendly interface on top of the commands.

top: The **top** command is the traditional way to view your system’s resource usage and see the processes that are taking up the most system resources. Top displays a list of processes, with the ones using the most CPU at the top. This utility tells the user about all the running processes on the Linux machine. Press 'q' on the keyboard to move out of the process display. To exit top or htop, use the **Ctrl-C** keyboard shortcut. This keyboard shortcut usually kills the currently running process in the terminal.

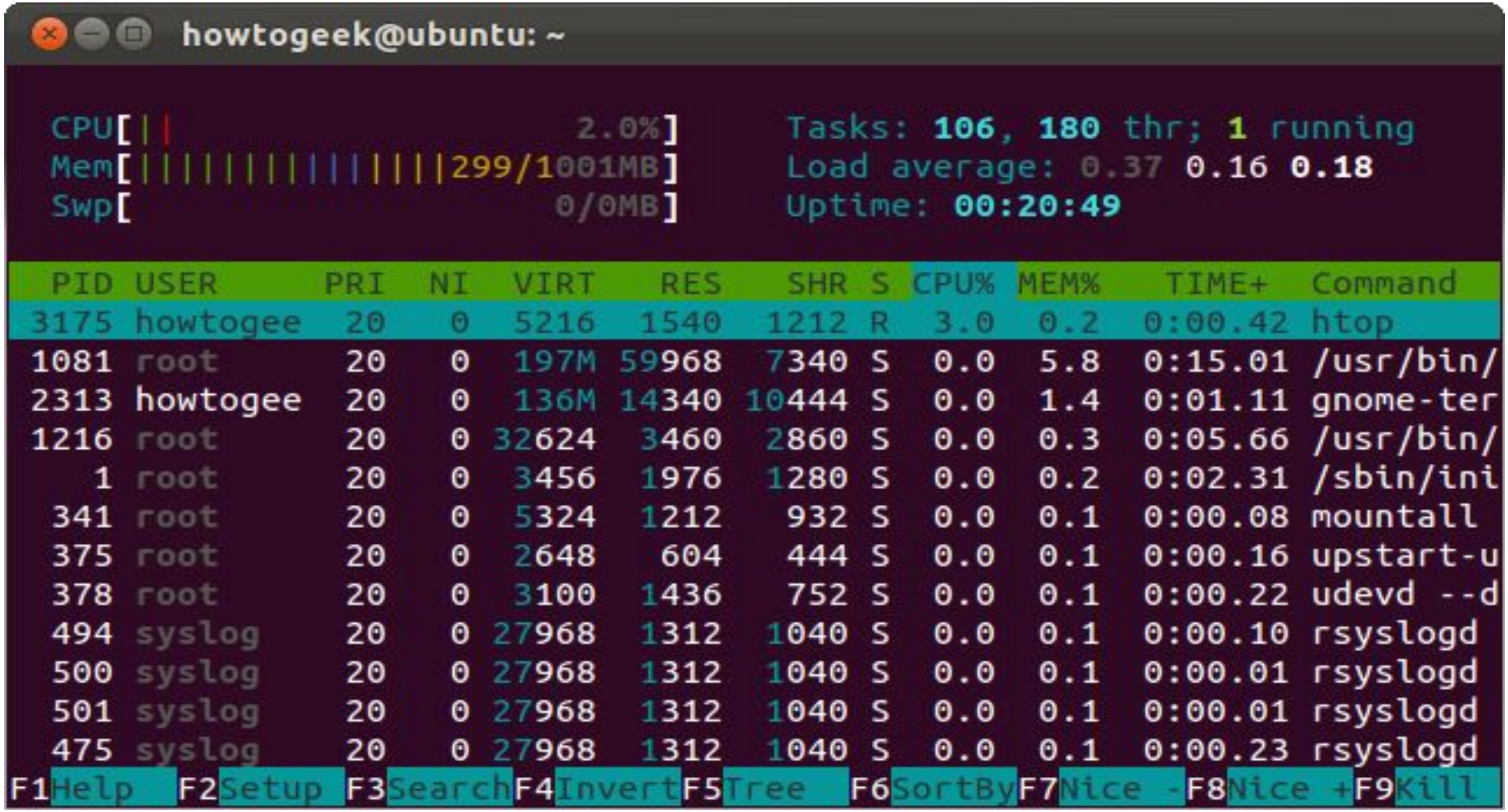
```
home@VirtualBox:~$ top

top - 23:57:43 up  2:54,  1 user,  load average: 0.00, 0.01, 0.05
Tasks: 189 total,   2 running, 187 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.7%us,  3.0%sy,  0.0%ni, 96.3%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1026080k total,   924508k used,   101572k free,    37000k buffers
Swap:  1046524k total,    21472k used,  1025052k free,   367996k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1525 home      20   0 1775m 100m  28m S  1.7 10.0   5:05.34 Photoshop.exe
  961 root      20   0 75972  51m 7952 R  1.0  5.1   2:23.42 Xorg
 1507 home      20   0  7644 4652  696 S  1.0  0.5   2:42.66 wineserver
 1564 home      20   0 75144  29m 9840 S  0.3  3.0   0:25.96 ubuntuone-syncd
 2999 home      20   0  127m  13m  10m S  0.3  1.4   0:01.36 gnome-terminal
 3077 home      20   0  2820 1188  864 R  0.3  0.1   0:00.76 top
    1 root      20   0  3200 1704 1260 S  0.0  0.2   0:00.98 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.95 ksoftirqd/0
```

Field	Description	Example 1	Example 2
PID	The process ID of each task	1525	961
User	The username of task owner	Home	Root
PR	Priority Can be 20(highest) or -20(lowest)	20	20
NI	The nice value of a task	0	0
VIRT	Virtual memory used (kb)	1775	75972
RES	Physical memory used (kb)	100	51
SHR	Shared memory used (kb)	28	7952

Field	Description	Example 1	Example 2
S	Status There are five types: 'D' = uninterruptible sleep 'R' = running 'S' = sleeping 'T' = traced or stopped 'Z' = zombie	S	R
%CPU	% of CPU time	1.7	1.0
%MEM	Physical memory used	10	5.1
TIME+	Total CPU time	5:05.34	2:23.42
		Photoshop.e	

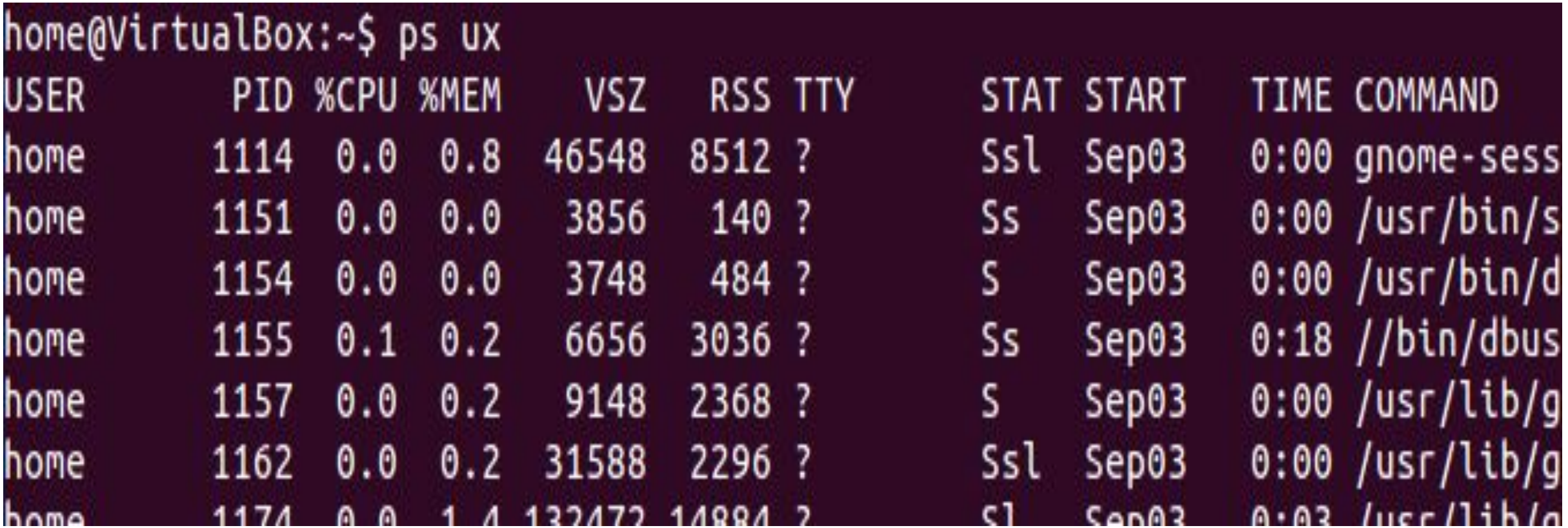
- **htop**
- The **htop** command is an improved top. It’s not installed by default on most Linux distributions — here’s the command you’ll need to install it on Ubuntu:
- sudo apt-get install htop

- htop displays the same information with an easier-to-understand layout. It also lets you select processes with the arrow keys and perform actions, such as killing them or changing their priority, with the F keys.


howtogeek@ubuntu: ~
CPU[|| 2.0%]
Mem[|||||||||||||||299/1001MB]
Swp[0/0MB]
Tasks: 106, 180 thr; 1 running
Load average: 0.37 0.16 0.18
Uptime: 00:20:49
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
3175 howtogee 20 0 5216 1540 1212 R 3.0 0.2 0:00.42 htop
1081 root 20 0 197M 59968 7340 S 0.0 5.8 0:15.01 /usr/bin/
2313 howtogee 20 0 136M 14340 10444 S 0.0 1.4 0:01.11 gnome-ter
1216 root 20 0 32624 3460 2860 S 0.0 0.3 0:05.66 /usr/bin/
1 root 20 0 3456 1976 1280 S 0.0 0.2 0:02.31 /sbin/ini
341 root 20 0 5324 1212 932 S 0.0 0.1 0:00.08 mountall
375 root 20 0 2648 604 444 S 0.0 0.1 0:00.16 upstart-u
378 root 20 0 3100 1436 752 S 0.0 0.1 0:00.22 udevd --d
494 syslog 20 0 27968 1312 1040 S 0.0 0.1 0:00.10 rsyslogd
500 syslog 20 0 27968 1312 1040 S 0.0 0.1 0:00.01 rsyslogd
501 syslog 20 0 27968 1312 1040 S 0.0 0.1 0:00.01 rsyslogd
475 syslog 20 0 27968 1312 1040 S 0.0 0.1 0:00.23 rsyslogd
F1Help F2Setup F3Search F4Invert F5Tree F6SortBy F7Nice - F8Nice + F9Kill

- **PS -** This command stands for 'Process Status'. It is similar to the "Task Manager" that pop-ups in a Windows Machine when we use Cntrl+Alt+Del. This command is similar to 'top' command but the information displayed is different. To check all the processes running under a user, use the command -

ps ux



- You can also check the process status of a single process, use the syntax –

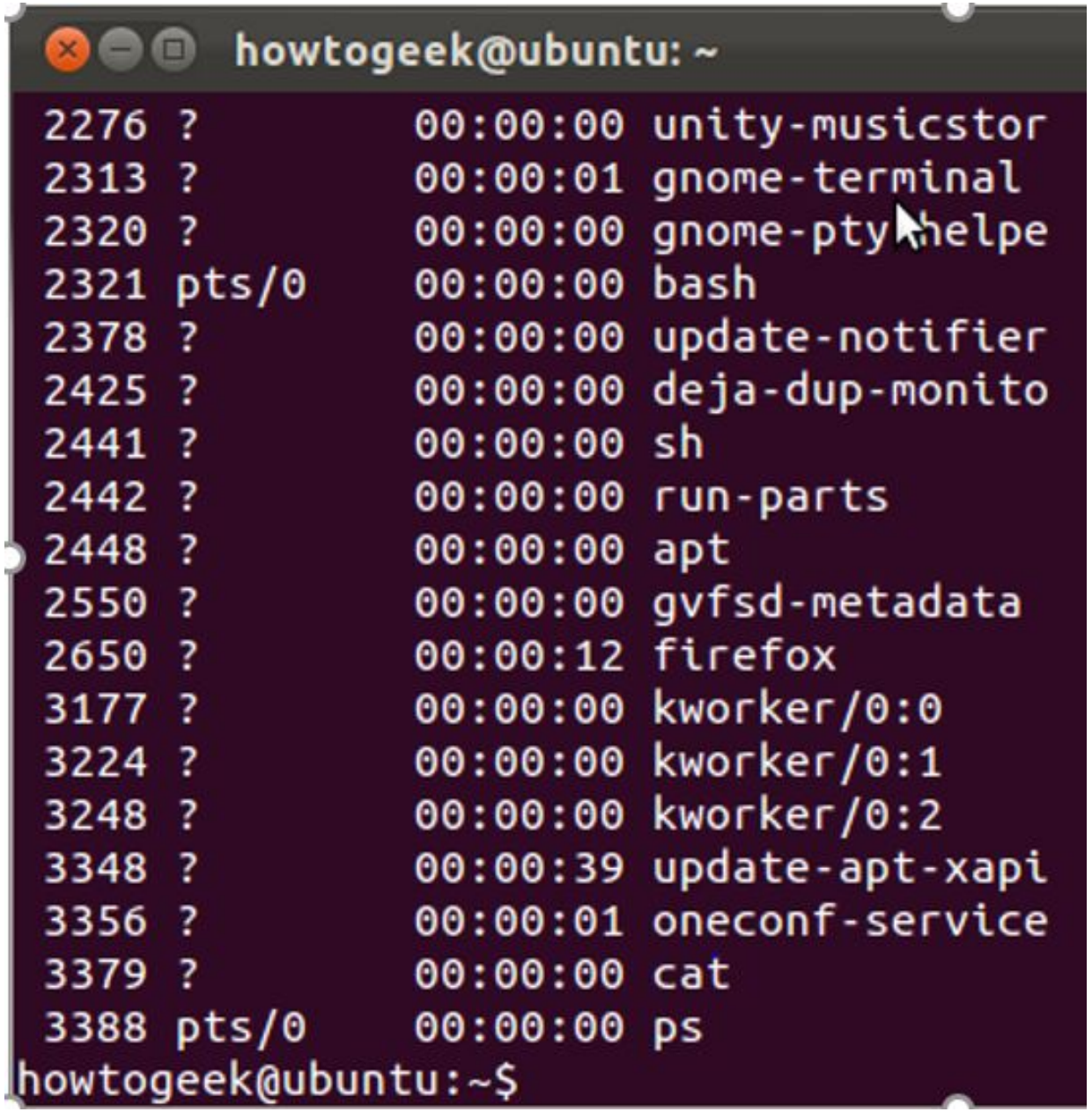
  ps PID

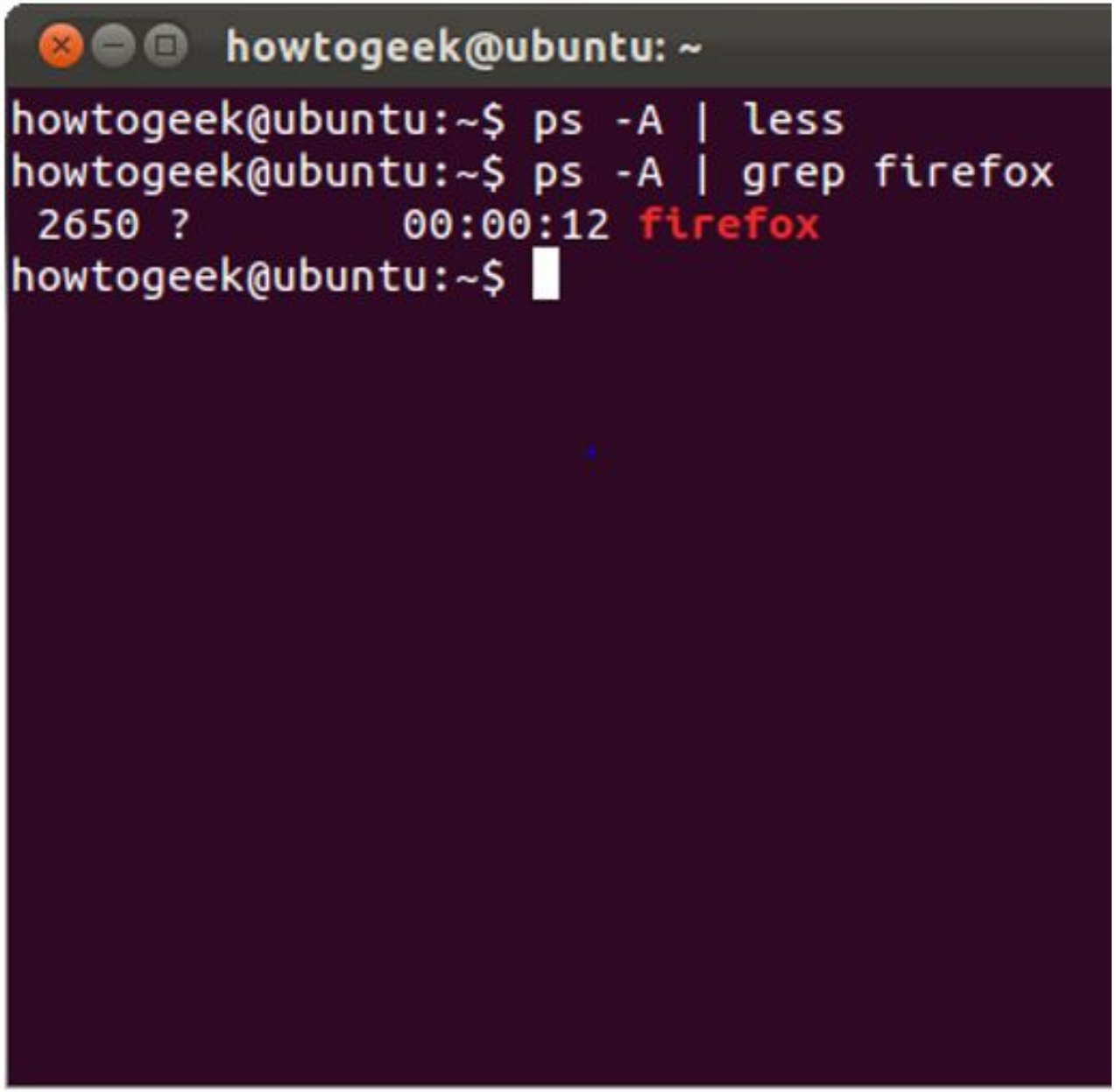
```
guru99@VirtualBox:~$ ps 1268
  PID TTY      STAT   TIME COMMAND
 1268 ?        S<l    0:02 /usr/bin/pulseaudio --start --log-target=syslog
```

- **ps**
- The **ps** command lists running processes. The following command lists all processes running on your system:
- ps –A

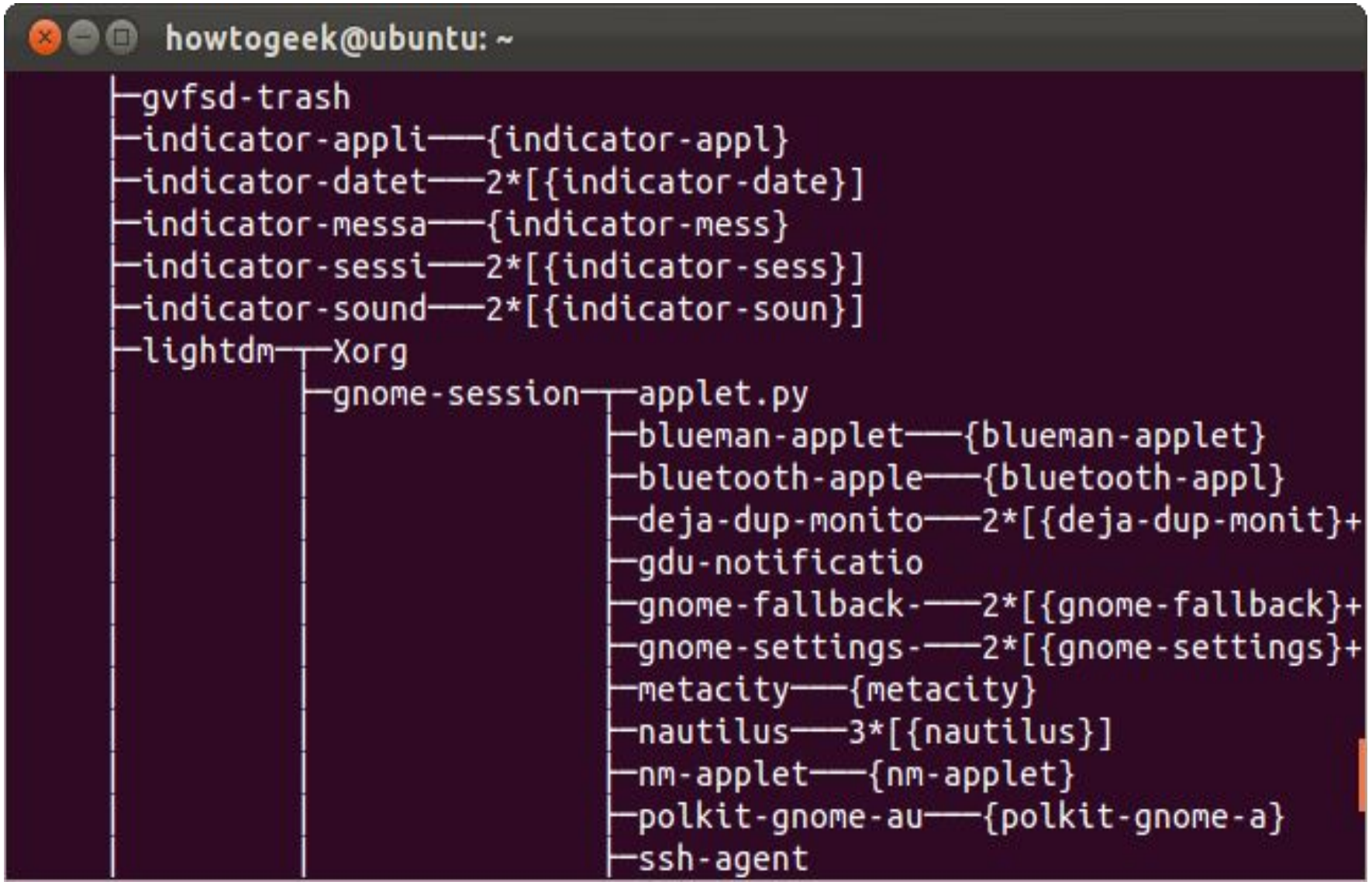
- This may be too many processes to read at one time, so you can pipe the output through the **less** command to scroll through them at your own pace:
- ps -A | less
- Press **q** to exit when you're done.
- You could also pipe the output through **grep** to search for a specific process without using any other commands. The following command would search for the Firefox process:
- ps -A | grep firefox

Results of ps





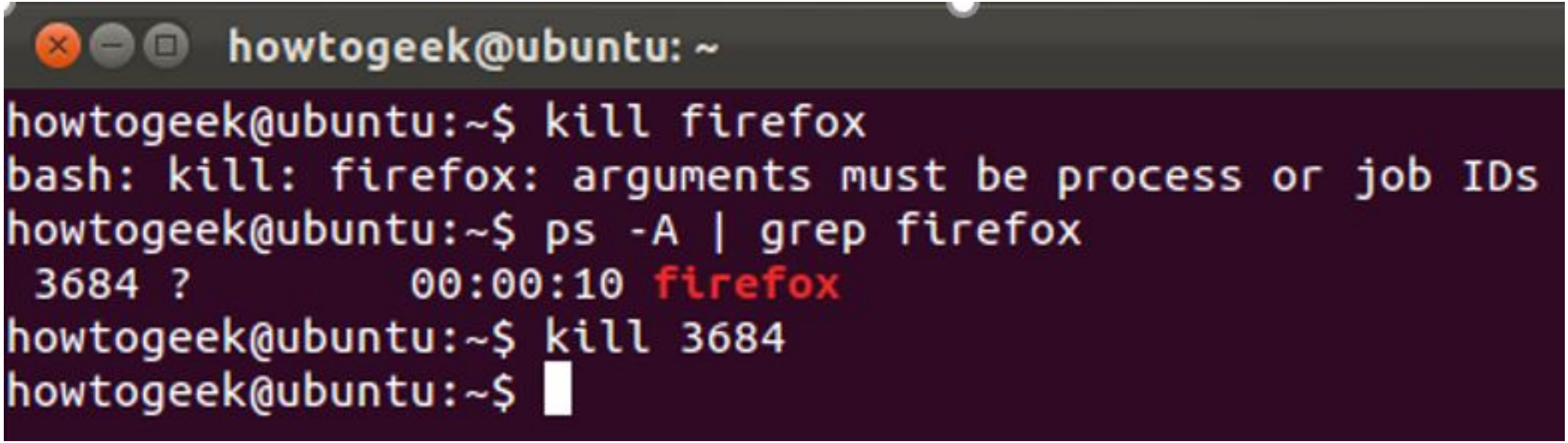
pstree: The **pstree** command is another way of visualizing processes. It displays them in tree format. So, for example, your X server and graphical environment would appear under the display manager that spawned them.



Kill: The **kill** command can kill a process, given its process ID. This command **terminates running processes** on a Linux machine. To use these utilities you need to know the PID (process id) of the process you want to kill. You can get this information from the **ps -A**, **top** or **pgrep** commands.

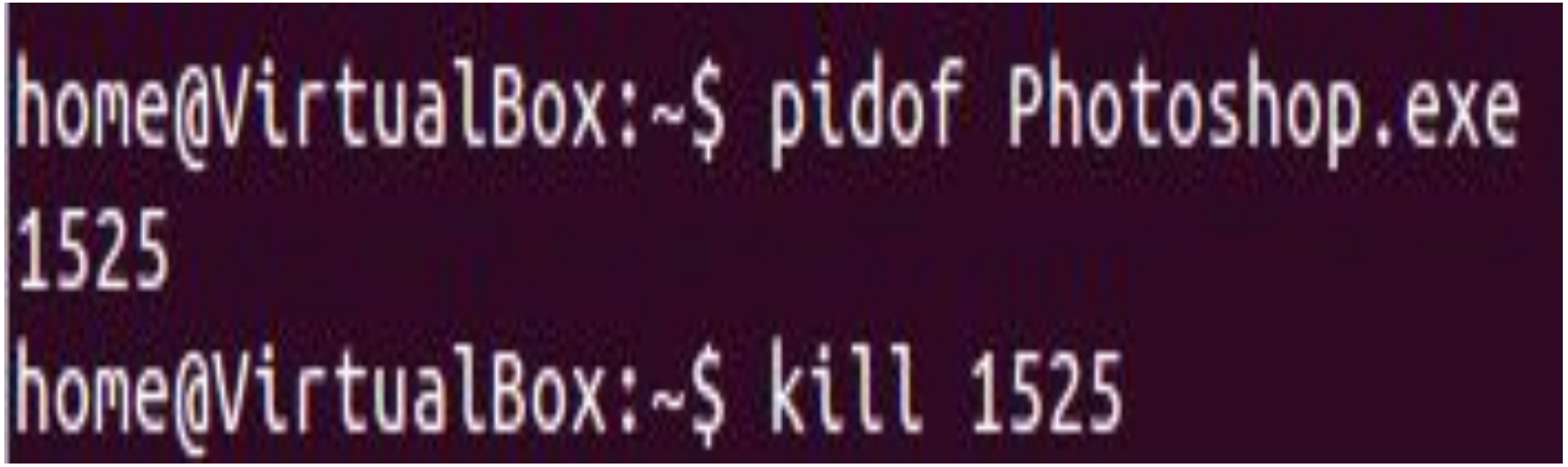
kill PID

Technically speaking, the kill command can send any signal to a process. You can use **kill -KILL** or **kill -9** instead to kill a stubborn process.

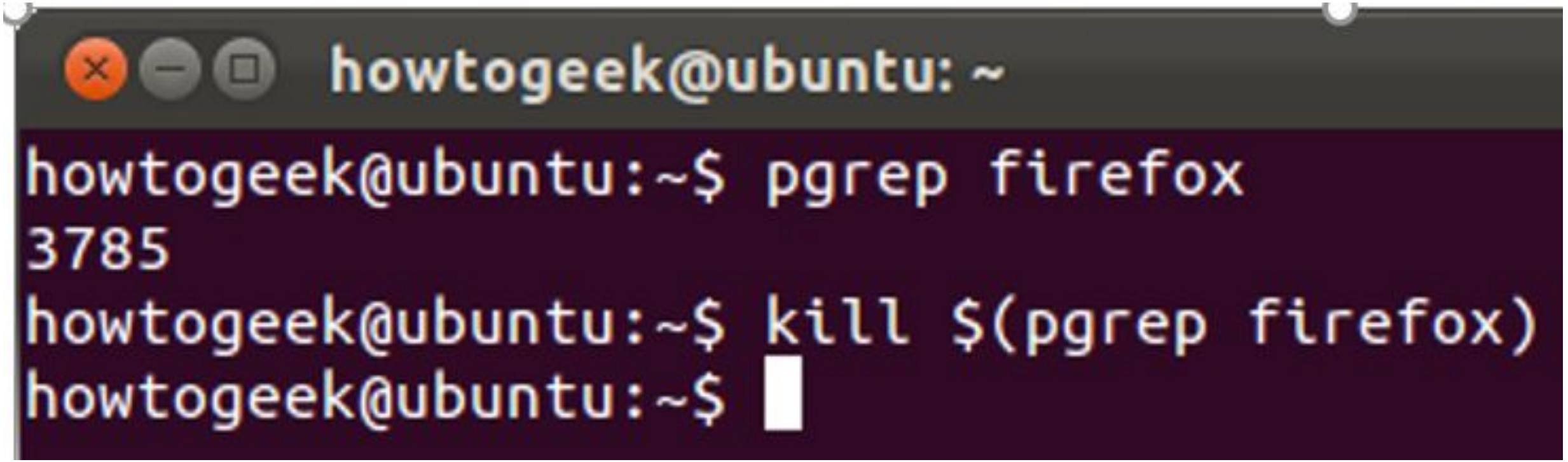


- To find the PID of a process simply type

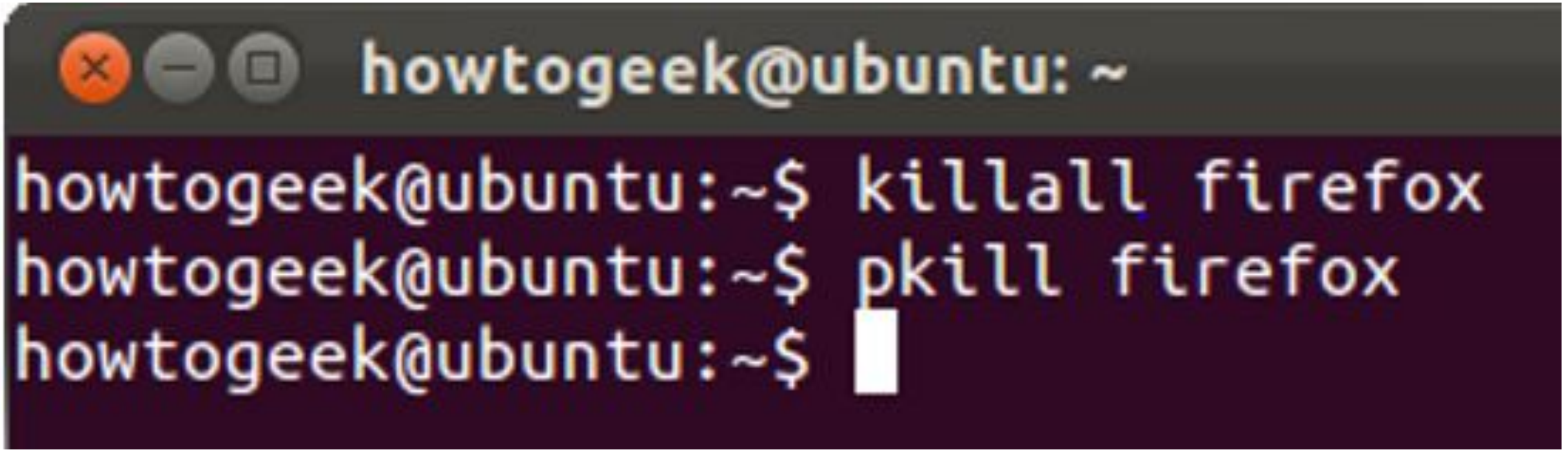
  pidof Process Name



- **pgrep**
- Given a search term, **pgrep** returns the process IDs that match it. For example, you could use the following command to find Firefox's PID:
- pgrep firefox
- You can also combine this command with kill to kill a specific process. Using pkill or killall is simpler, though.



- **pkill & killall**
- The **pkill** and **killall** commands can kill a process, given its name. Use either command to kill Firefox:
- pkill firefox
  killall firefox



NICE

- Linux can run a lot of processes at a time, which can slow down the speed of some high priority processes and result in poor performance.
- To avoid this, you can tell your machine to prioritize processes as per your requirements.
- This priority is called Niceness in Linux, and it has a value between -20 to 19. The lower the Niceness index, the higher would be a priority given to that task.
- The default value of all the processes is 0.
- To start a process with a niceness value other than the default value use the following syntax

Syntax is: nice –n ‘Nice Value’ process Name

```
home@VirtualBox:~$ nice -n 19 banshee
```

- If there is some process already running on the system, then you can 'Renice' its value using syntax.

  renice ‘nice value’ –p ‘PID’

- To change Niceness, you can use the 'top' command to determine the PID (process id) and its Nice value. Later use the renice command to change the value.

Let us understand this by an example.

Checking the niceness value of the process 'banshee'

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 3293 home      20   0  277m   64m   35m S  96.4   6.4   9:56.72 banshee
```

Renicing the value to -20

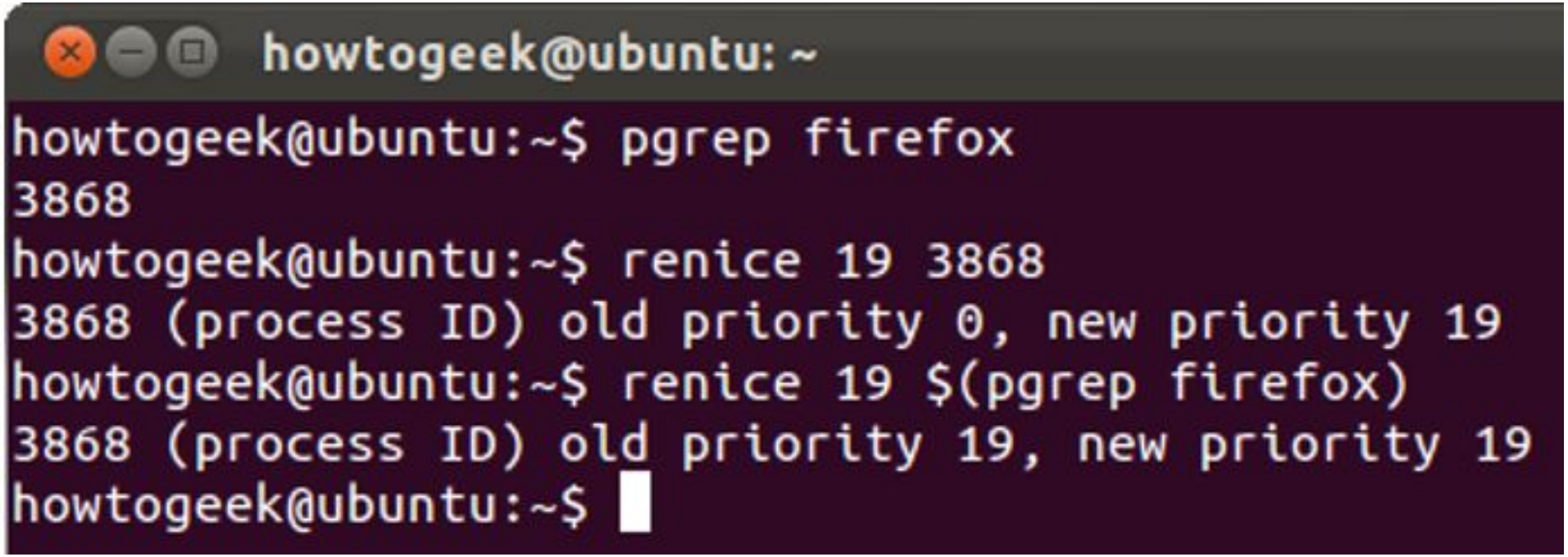
```
home@VirtualBox:~$ sudo renice -20 -p 3293
[sudo] password for home:
3293 (process ID) old priority 0, new priority -20
```

The value changed to -20

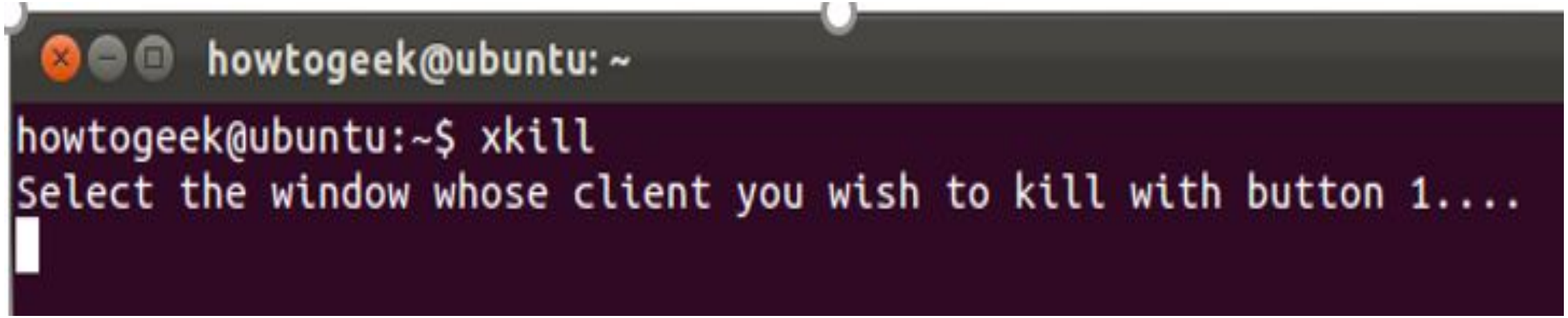
```
 3293 home      0 -20  277m   64m   35m S  95.2   6.4   3:32.95 banshee
```

Renice: The **renice** command changes the nice value of an already running process. The nice value determines what priority the process runs with. A value of **-19** is very high priority, while a value of **19** is very low priority. A value of **0** is the default priority.

- The renice command requires a process's PID. The following command makes a process run with very low priority:
- renice 19 *PID*
- You can use the **pgrep** trick above with renice, too.
- If you're making a process run at a higher priority, you'll require root permissions. On Ubuntu, use **sudo** for that:
- sudo renice -19 #



- **xkill**
- The **xkill** command is a way of easily killing graphical programs. Run it and your cursor will turn into an **x** sign. Click a program's window to kill that program. If you don't want to kill a program, you can back out of xkill by right-clicking instead.
- You don't have to run this command from a terminal — you can also press Alt-F2, type **xkill** and press Enter to use it from a graphical desktop.



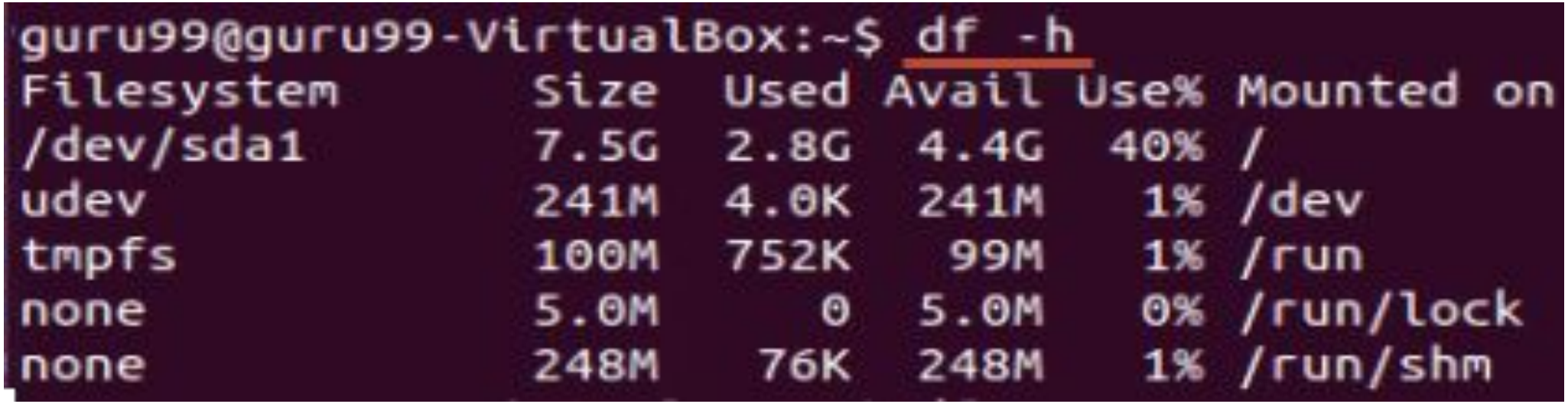
DF

- This utility reports the free disk space(Hard Disk) on all the file systems.

```
guru99@guru99-VirtualBox:~$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/sda1        7837756 2921376   4523216  40% /
udev              246488       4    246484   1% /dev
tmpfs             101512     752    100760   1% /run
none                5120       0      5120   0% /run/lock
none              253776      76    253700   1% /run/shm
```

If you want the above information in a readable format, then use the command

df –h



Free

- This command shows the free and used memory (RAM) on the Linux system.

```
home@VirtualBox:~$ free
             total       used       free     shared    buffers     cached
Mem:       1026080     803604     222476          0      36312     343376
-/+ buffers/cache:     423916     602164
Swap:      1046524      35832    1010692
```

- You can use the arguments
- free -m to display output in MB
- free -g to display output in GB

Summary:

- Any running program or a command given to a Linux system is called a process
- A process could run in foreground or background
- The priority index of a process is called Nice in Linux. Its default value is 0, and it can vary between 20 to -19
- The lower the Niceness index, the higher would be priority given to that task

Command	Description
bg	To send a process to the background
fg	To run a stopped process in the foreground
top	Details on all Active Processes
ps	Give the status of processes running for a user
ps PID	Gives the status of a particular process
pidof	Gives the Process ID (PID) of a process
kill PID	Kills a process
nice	Starts a process with a given priority
renice	Changes priority of an already running process
df	Gives free hard disk space on your system
free	Gives free RAM on your system

Kill Command

- Use the kill command to send a signal to each process specified by a pid (process identifier). The default signal is SIGTERM (terminate the process).
- Terminate process with given signal. Falls under process management category and Root privileges are not required.
- It removes background processes or jobs.

- Syntax kill PID

  kill –s SignalName PID

  kill –SignalNumber PID

  kill –SignalName PID

- signalNumber : A non-negative decimal integer, specifying the signal to be sent instead of the default TERM.
- signalName : A symbolic signal name specifying the signal to be sent instead of the default TERM.
- PID : Specify the list of processes that kill should signal. Each PID can be any one of the following:
  - n: If PID is a positive value, the kill command sends the process whose process ID is equal to the PID
  - 0: All processes in the current process group are signaled.
  - -1: All processes with pid larger than 1 will be signaled i.e. the kill command sends the signal to all processes owned by the effective user of the sender.

Common Linux and UNIX signal names and numbers (Total 0-31)

Number	Name (short name)	Description	Used for
0	SIGNULL (NULL)	Null	Check access to pid
1	SIGHUP (HUP)	Hangup	Terminate; can be trapped
2	SIGINT (INT)	Interrupt	Terminate; can be trapped
3	SIGQUIT (QUIT)	Quit	Terminate with core dump; can be trapped
9	SIGKILL (KILL)	Kill	Forced termination; cannot be trapped
15	SIGTERM (TERM)	Terminate	Terminate; can be trapped
24	SIGSTOP (STOP)	Stop	Pause the process; cannot be trapped. This is default if signal not provided to kill command.
25	SIGTSTP (STP)	Terminal	Stop/pause the process; can be trapped
26	SIGCONT (CONT)	Continue	Run a stopped process