Linux System Administration

The Boot Process

- What happens when we turn on our workstation and try to boot into Linux?
  - The BIOS checks the system.
  - The Boot loader finds the kernel image, loads it into memory, and starts it.
  - The kernel initializes devices and their drivers.
  - The kernel mounts the root filesystem.
  - The kernel starts the `init` program.
  - `init` gets the rest of the processes started
  - The last process that `init` starts will allow you to login.

Linux System Administration

The Boot Process

- What is the Master Boot Record (MBR)?
  - It is the first 512 bytes located on the first sector of the media.
  - The MBR has enough information to determine four primary partitions:
    - The start cylinder for each partition
    - The number of cylinders for each partition
    - The id or type of each partition
    - Is the partition bootable?

Linux System Administration

The Boot Process

- What is the Boot Loader?
  - First stage: The boot loader locates and reads into memory the first stage of an operating system.

    Second Stage: The boot loader then transfers control to the rest of the operating system.
  - In order for a medium to be bootable, the boot loader must be on one of the following:
    - The boot sector of a floppy disk
    - The MBR of the first hard disk
    - The MBR of the first CD-ROM device
    - The boot sector of a Linux filesystem partition on the first hard drive
    - The boot sector of an extended partition on the first hard drive
    - Many Linux distributions are using GRUB (GRand Unified Boot loader)

Linux System Administration

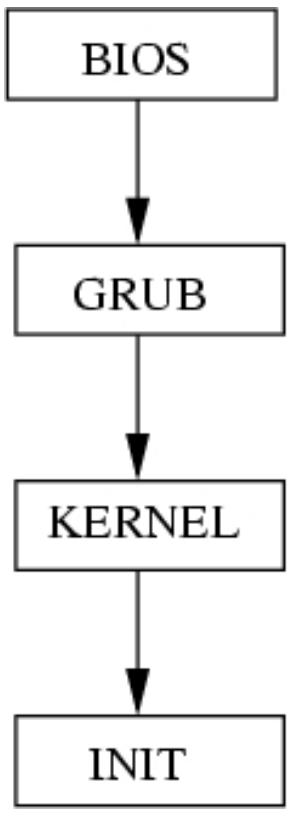
The Boot Process

- What happens when we turn on our workstation and try to boot into Linux?
  - The BIOS checks the system.
  - The Boot loader finds the kernel image, loads it into memory, and starts it.
    - initrd is a file system loaded in at boot time that loads drivers to get the kernel going.
  - The kernel initializes devices and their drivers.
  - The kernel mounts the root filesystem.
  - The kernel starts the `init` program.
  - `init` gets the rest of the processes started
  - The last process that `init` starts will allow you to login.

Linux System Administration

The Boot Process

- Order of the boot procedure:



Linux System Administration

The Boot Process

GRUB

- Fedora automatically installs GRUB
- GRUB files are located in /boot/grub
  - `/boot/grub.conf` is also a link in `/etc/grub.conf` and also in `/boot/menu.lst`
- When you add a new kernel or OS, you need to edit `/boot/grub.conf`
- GRUB will boot the default OS.
  - It is possible to control GRUB by pressing e (for edit) at the GRUB prompt.

Linux System Administration

The Boot Process

GRUB Editor Commands

B	Boot the currently selected operating system
E	Edit the currently selected GRUB command
C	Open a screen for interactively entering and executing GRUB commands
O	Enter a new command before/after the currently selected command
D	Delete the currently selected command
Esc	Return to the main GRUB menu.

From : *Learning Red Hat Enterprise Linux & Fedora*

Linux System Administration

The Boot Process

GRUB Commands

`chainloader`	Used to load Microsoft operating systems
`initrd`	Specifies the file containing the initial RAM disk.
`kernel`	Specifies the file containing the Linux kernel to be booted
`root` `rootnoverify`	Specifies the partition to be mounted as the root partition. The root command causes the filesystem to be verified before the partition is mounted.

From : *Learning Red Hat Enterprise Linux & Fedora*

Linux System Administration

The Boot Process

/boot/grub.conf

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd1,0)
#          kernel /vmlinuz-version ro root=/dev/hdb3
#          initrd /initrd-version.img
#boot=/dev/hda
default=0
timeout=10
splashimage=(hd1,0)/grub/splash.xpm.gz
title Fedora Core (2.6.9-1.667)
        root (hd1,0)
        kernel /vmlinuz-2.6.9-1.667 ro root=LABEL=/ rhgb quiet
        initrd /initrd-2.6.9-1.667.img
title Other
        rootnoverify (hd0,0)
        chainloader +1
```

Linux System Administration

The Boot Process

GRUB Problems

- What happens if we accidentally write over the MBR or if the MBR becomes corrupted?
  - GRUB is gone, and we have no boot loader!
- `/sbin/grub-install /dev/hda`

Linux System Administration

Basic Commands

---

ls (Listing)

- This command will show you the contents of a directory.
  - `ls` --> will show you the contents of the current directory.
  - `ls /dir/name` --> will show you the contents of a specified directory.
  - `ls -l` --> will show you a *long* listing containing ownership, permissions, time last modified, and size.
  - `ls -a` --> will show you *all* of the files in the directory, including those starting with a .
  - `ls -al` --> What do you think?

```
[mlevan@localhost BasicCommands]$ ls -al
total 24
drwxrwxr-x   2 mlevan mlevan 4096 Apr 30 17:43 .
drwxr-xr-x  10 mlevan mlevan 4096 Apr 30 17:36 ..
-rw-rw-r--   1 mlevan mlevan 1828 Apr 30 17:57
Basic_page1.html
-rwxr-xr-x   1 mlevan mlevan 4542 Apr 30 17:37 logo2.gif
-rw-rw-r--   1 mlevan mlevan 1409 Apr 30 17:36
template.html
```

  - Note that . stands for the current directory and .. refers to the parent directory

Linux System Administration

Basic Commands

---

cd (Change Directory)

- This command will change your current working directory.
- `cd` --> If you just type in `cd`, then you will be sent to your home directory. For example, `/home/mlevan/`

- `cd /dir/name` --> This command will send you directly into the desired directory.
  - `cd /var/log/` --> This will send us to the /var/log directory.

- What about these commands :
  - `cd .`
  - `cd ..`

Prev Page 2 Next

Linux System Administration

Basic Commands

---

cp (CoPy)

- `cp filename1 filename2` --> This command will copy the first file into the second file.
- `cp Amy.txt Garret.txt`
- Note that if Garret.txt is already a file, then it will be overwritten !! Be careful with this command.
- `cp -i Amy.txt Garret.txt`
  - If Garret.txt exists, then this command will inquire if you want to overwrite the file.
  - If Garret.txt does not exist, then you will not be asked.
- Note that you can also add directory names to this:
- `cp /home/mlevan/Amy.txt /home/guest/Garret.txt`
- You can also copy files to a directory :
  - `cp file1 file2 fileN directory_name`
  - `cp Amy.txt Garret.txt temp/`
- Note that ~ can also represent your home directory. For example, say I want to copy a file from /home/guest1/booty to the temp directory in my account:
  - `cp /home/guest1/booty/blah.txt ~/temp/`

Linux System Administration

Basic Commands

---

rm (ReMove)

- The `rm` command will remove a file.
  - `rm filename`
- If you type in `rm -i filename`, then you will be asked if you really want to remove the file.
- It is virtually impossible to regain a file after it has been removed in this fashion.

Linux System Administration

Basic Commands

---

mv (MoVe)

- This is the "rename" command used in DOS.
- This command moves one filename into another filename.
  - `mv filename1 filename2`
- The above command automatically writes over filename2 with whatever was in filename1
  - `mv -i filename1 filename2`
- The above command will inquire if you really want to move the file.
- You can also move directories with this command,
  - `mv dir_name1 dir_name2`

Linux System Administration

Basic Commands

---

touch

- This command will create a file.
    - `touch filename`
- If the file already exists, then touch will update the timestamp of the file.

Linux System Administration

Basic Commands

---

echo

- This is a command that will print to the screen whatever is after the word echo.
  - `echo text text text ...  text`
- When can this be useful?

Linux System Administration

Basic Commands

---

mkdir (MaKe DIRectory)

- This command will create a directory in your current working directory:
  - mkdir dir_name
- You can create a directory anywhere using the full pathname... if you have permission:
  - mkdir /var/log/class

Prev Page 8 Next

Linux System Administration

Basic Commands

---

rmdir (ReMove DIRectory)

- This command will remove an empty directory.
  - `rmdir temp`

- If the directory is not empty, then pass the parameters r (recursive) and f (force) to the rm command.
  - The f parameter will force the removal, never inquiring if you want to remove any subsequent files or directories.
  - The r parameter will remove travel down any directories within the directory and remove all the files.

. `rm -rf dir_name`

Linux System Administration

Basic Commands

---

cat

- This command will print out a text file.
  - `cat filename`
- What happens if we pass two files to this command?
  - `cat filename1 filename2`
- What happens if we don't pass any files to this command?
  - `cat`
- Control-D or Control-C ?
  - Control-C terminates a program.
  - Control-D stops the current input. (Admittedly, this can also end a program)

Linux System Administration

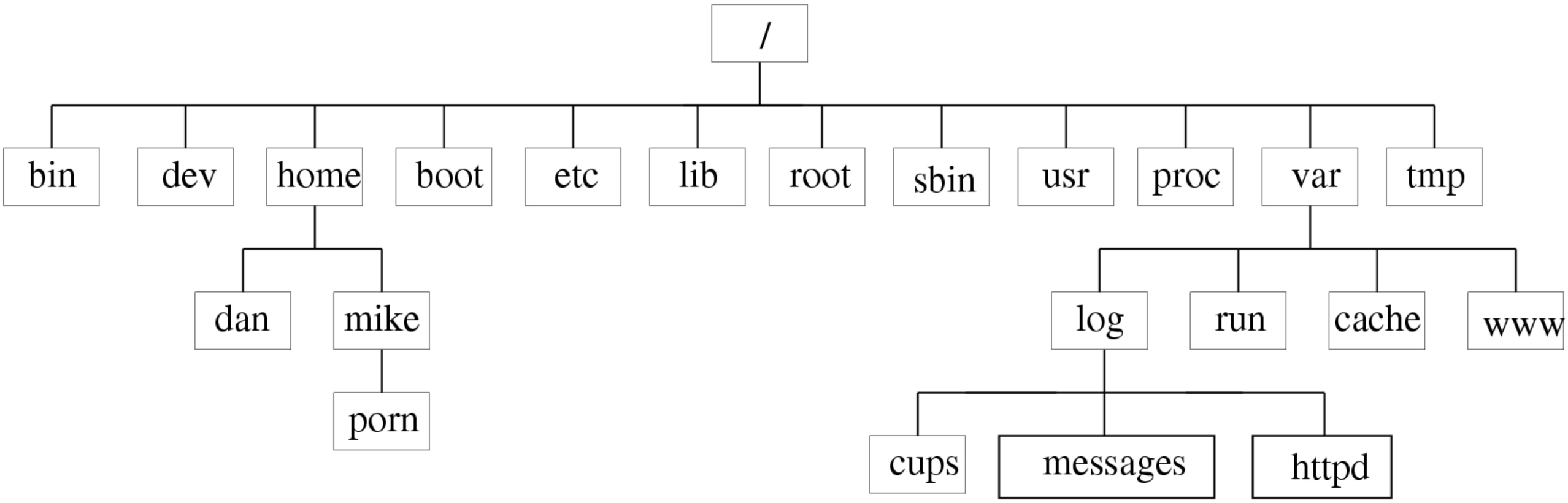
Directory Structure

- Much of the information from this section can be found at the Filesystem Heirarchy Standard webpage.
  - http://www.pathname.com/fhs/
- Or through this PDF file:
  - http://www.pathname.com/fhs/pub/fhs-2.3.pdf
- Please review this PDF file for information about all of the directories and their purpose. Today, we shall try to highlight a few of them. Almost all of the information from this lesson was shamefully pilfered from here.

Linux System Administration

Directory Structure

- Here is an example of a directory structure. Note that not all directories are listed.



Linux System Administration

Directory Structure

/bin

- This directory contains commands that may be used by both the superuser and the regular users.
- These commands are required when no other file systems are mounted (e.g. single user mode)
- Some examples of commands:
  - ls, cp, rm, etc.
- Note that there can be no subdirectories in /bin ???

Prev Page 3 Next

Linux System Administration

Directory Structure

/sbin

- This directory contains utilities to be used by the system administrator.
- These binaries are essential for booting, restoring, recovering, and/or repairing the system.
- You may also find binaries in `/usr/sbin` and `/usr/local/sbin`.
  - The binaries are placed in `/usr/sbin` if it is needed after /usr has been mounted.
  - The binaries are placed in `/usr/local/sbin` if it is a utility installed locally.

Prev Page 4 Next

Linux System Administration

Directory Structure

/dev

- This is the location of special or device files
- This could be an IDE hard drive : `/dev/hda` `/dev/hdb` `/dev/hd*`
- This could be a SCSI hard drive : `/dev/sda` `/dev/sdb/` `/dev/sd*`
- This could be an audio device : `/dev/audio` `/dev/dsp` `/dev/mixer`
- This could be a terminal : `/dev/tty0` `/dev/tty1` `/dev/tty*` `/dev/pts/*` `/dev/tty`
- This could be a serial port : `/dev/ttyS*`
  - COM port1 --> `/dev/tty0` COM port 2 --> `/dev/tty1`
- This could be a floppy disk : `/dev/fd*`
- This could be a parallel port : `/dev/lp0` `/dev/lp1`

Linux System Administration

Directory Structure

/home

- This directory holds personal files for normal users on the system.
- This setup can vary slightly from system to system, so no program should rely on this directory.

Linux System Administration

Directory Structure

/boot

- This directory contains everything required for the boot process except configuration files not needed at boot time and the map installer.
- The operating system kernel must either be in / or /boot.

Linux System Administration

Directory Structure

/etc

- This directory contains configuration files.
  - A configuration file is a local file used to control the operation of a program.
  - It must be static.
- No binary files should be located in /etc.
- User password, boot, device, networking, and other setup files are here.
- Many items in /etc are specific to the hardware.
  - `/etc/X11` directory contains the graphics card configuration.

Linux System Administration

Directory Structure

/lib

- This directory contains shared library code needed to boot the system and run the commands in the root filesystem.
- The files in this directory should be static.
- Other library directories (`/usr/lib`) could contain static and shared libraries.

Linux System Administration

Directory Structure

/root

- This is the root account's home directory.
- Some feel this directory is optional.

Linux System Administration

Directory Structure

/usr

- This is a large directory that looks a little like `/` (the top level directory)
- Much of the Linux system resides in `/usr`
- There are many subdirectories, including :
  - /local : Where administrators can install their own software.
  - /bin : Most user commands
  - /include : header files included by C programs
  - /man : this contains the man pages
  - many others

Linux System Administration

Directory Structure

/proc

- The standard Linux method for handling process and system information.
- Provides system statistics through a directory and file interface.

```
[mlevan@localhost proc]$ more swaps
Filename                                Type            Size    Used
Priority
/dev/hdb5                               partition       2152668 8       -1
```

- This directory has many directories that are just numbers. These represent processes that are running.

Linux System Administration

Directory Structure

/var

- This directory contains variable data files.
  - This includes spool directories, administrative and logging files, and transient or temporary files.
- Programs record runtime information.
- It is often good to make `/var` a separate partition during installation.

Linux System Administration

Directory Structure

/tmp

- This directory is the place to put smaller temporary files that you don't care much about.
- Any user may read to and write from /tpm, but they can't access other's files in this directory.
- Most distributions clear /tmp when booting.
  - Don't put anything important in /tmp.
- Some programs use this directory as a workspace.

Prev Page 14 Next

Linux System Administration

Directory Structure

/media

- This directory contains mount points for removable media.
  - /media/cdrom
  - /media/floppy
  - /media/cdrecorder

Linux System Administration

Filesystems

- What is a filesystem?
  - A filesystem is a database of files and directories that you can attach to a Unix system at the root (/) or some other directory (like /usr) in a currently attached filesystem.

  - In other words, Linux places all the partitions under the root (/) directory.
  - The partitions are mounted (loaded) under certain directories.
  - Unless a partition is mounted, Linux does not know it exists.

- How does Linux know which partitions to mount ?
  - /etc/fstab

Prev Page 1 Next

Linux System Administration

Filesystems

- Example:

```
[mlevan@localhost Filesystems]$ more /etc/fstab
# This file is edited by fstab-sync - see 'man fstab-sync' for details
/dev/hdb3               /                       ext3    defaults        1 1
/dev/hdb1               /boot                   ext3    defaults        1 2
none                    /dev/pts                devpts  gid=5,mode=620  0 0
none                    /dev/shm                tmpfs   defaults        0 0
/dev/hdb2               /home                   ext3    defaults        1 2
none                    /proc                   proc    defaults        0 0
none                    /sys                    sysfs   defaults        0 0
/dev/hdb4               /usr                    ext3    defaults        1 2
/dev/hdb5               swap                    swap    defaults        0 0
/dev/hdd                /media/cdrecorder       auto
pamconsole,exec,noauto,managed 0 0
/dev/hdc                /media/dvd              auto
pamconsole,exec,noauto,managed 0 0
/dev/fd0                /media/floppy           auto
pamconsole,exec,noauto,managed 0 0
```

Linux System Administration

Filesystems

/etc/fstab

- This is a text file that shows the following:
  - The partition or device to mount
  - Where to mount the partition
  - The type of filesystem
  - Options
  - Backup information for the dump command
  - The filesystem integrity check during boot.
    - 0 means do not check
    - Usually used for devices such as CD or DVD players.
    - All other numbers represent the order checked during boot up.
    - / should always be set to 1
    - Others to be checked get set to 2

Linux System Administration

Filesystems

- How can we mount a filesystem by hand?
- Use the "mount" command with the filesystem type, device, and desired mount point.
  - `mount -t type device mountpoint options`
  - `mount -t ext2  /dev/hdb3  /home/extra`
    - Why is this a bad idea?
      - `. mount -t ext2  /dev/hdb3  /usr`
  - How can you unmount a filesystem?
    - `. umount  mountpoint`

Linux System Administration

Filesystems

Filesystem Types

- ext2
  - The second extended filesystem.
  - This is native to Linux.
  - Nearly every Linux system uses ext2 or .....

- ext3
  - This is the ext2 filesystem with journal support.
    - This journal contains changes not yet written to to regular filesystem database.
  - The journaling system can make recovery from an abrupt system reboot or system failure quicker and less painful.
    - Journaling modes:
      - data=writeback : smaller journals, faster speed
      - data=journal : larger journals, usually slower speeds
      - data=ordered : balanced journal and speed; default option.
      - These are options you can use in /etc/fstab.

Linux System Administration

Filesystems

- ISO9660
  - This is a CD-ROM standard. Most CD-ROMs use some variety of the ISO9660 extension.

- FAT
  - FAT filesystems (msdos, vfat, umsdos) pertain to Microsoft systems.
    - msdos supports older MS-DOS and Windows (3.11 and older) systems.
    - vfat supports Windows systems (95, 98, ME)
    - umsdos is an uncommon type that supports Unix features of an MS-DOS filesystem.

- NTFS
  - Windows NT, 2000, XP systems.
  - This is not always included in a default kernel, so you might need to add this module.

- Reiser
  - Relatively new.
  - Supports a journal and is optimized for fairly small files which is common in Unix systems.

Linux System Administration

Filesystems

Special Purpose Filesystems

- proc
  - Short for "process", mounted on /proc
  - Each numbered directory represents a current process.
    - The files in these directories represent various aspects of the processes.
  - This filesystem contains a great deal of additional kernel and hardware information.

- usbdevfs
  - mounted on /proc/bus/usb
  - Programs that interact with the USB interface and its devices often need the files here.
  - The files contain information on the bus status.

- tmpfs
  - mounted on /dev/shm
  - This allows you to use physical memory and swap space as temporary storage.
  - Be careful here as you don't want to overload this filesystem as you will quickly run out of system resources.

Linux System Administration

Filesystems

Special Purpose Filesystems

- devpts
  - The *devpts* file system provides an interface to pseudo terminal (*pty*) devices.
  - A pseudo-terminal device is a terminal device that does not have a physical terminal associated with it.
    - consoles, shells
  - At mount time, a user identity, group identity, and mode can be specified for all pty files in the *devpts* file system.
  - Typically, this feature is used to set the group and mode to allow write access by programs that are setgid to the *tty* group.

Linux System Administration

Filesystems

Special Purpose Filesystems

- sysfs
  - The sysfs virtual filesystem is one of the many additions to the 2.6 kernel.
  - it is used by the `udev` utility to create device nodes for hardware and, eventually, numerous other purposes.
  - There is a lot of information about the system available under sysfs; it may, eventually, replace many of the files currently found under `/proc`
  - This has led to a smaller footprint in memory.

Linux System Administration

Filesystems

Filesystem Capacity

- How can you tell how much space is left on your filesystems?
- With the df (disk full) command.
  - df
  - df -h
  - df -i
- Note that each filesystem has reserved blocks that only the super-user can access.
  - This keeps system servers from failing if the partitions run out of disk space.

Linux System Administration

Filesystems

Adding A New Filesystem

- Let's say we just bought a new hard drive or have some free space left on our current system and we want to add a new partition to our current system. What do we need to do?

  1. Use fdisk to create a new partition.
  2. The kernel needs to be aware of the new partition table, so either reboot or run `partprobe`.
     - Note that if `partprobe` hangs, then you must reboot.
  3. We need to add a filesystem to the new partition.
     - `mke2fs <options> device`
     - `mke2fs -j /dev/hda6`
  4. We now need to mount the partition. Let's call the partition "NEW"
     - `mount /dev/dha6 /NEW`
  5. Create any needed mount points.
  6. Add a line to /etc/fstab to mount this new partition at boot time.

```
[mlevan@localhost Filesystems]$ more /etc/fstab
# This file is edited by fstab-sync - see 'man fstab-
sync' for details
/dev/hdb3               /                       ext3
defaults        1 1
/dev/hdb1               /boot                   ext3
defaults        1 2
none                    /dev/pts                devpts
gid=5,mode=620  0 0
none                    /dev/shm                tmpfs
defaults        0 0
/dev/hdb2               /home                   ext3
defaults        1 2
none                    /proc                   proc
defaults        0 0
none                    /sys                    sysfs
defaults        0 0
/dev/hdb4               /usr                    ext3
defaults        1 2
/dev/hdb5               swap                    swap
```

```
defaults        0 0
/dev/hdd                    /media/cdrecorder       auto
pamconsole,exec,noauto,managed 0 0
/dev/hdc                    /media/dvd              auto
pamconsole,exec,noauto,managed 0 0
/dev/fd0                    /media/floppy           auto
pamconsole,exec,noauto,managed 0 0
/dev/hda6                   /NEW                    ext3
defaults        1 2
```

Linux System Administration

Filesystems

Ooops!

- Oh no! I meant to make the filesystem ext3, but I forgot the -j option. What can I do?
- In order to convert an ext2 partition into an ext3 partition, use the tune2fs command.

1. Make sure you change the filesystem type in /etc/fstab
2. Unmount the filesystem to be changed.
3. Use tune2fs to convert the filesystem:
   - tune2fs -j partition
   - tune2fs -j /dev/hda6
4. Mount the filesystem

Prev Page 12 Next

Linux System Administration

Filesystems

Ooops! I did it again!

- What if I forgot to install a swap space when I installed Linux?
  - Swap space is not needed, but it is useful.
- What if I want to add another swap space to my system?
  - It's OK. You can have up to 8.
- How do I do it?

1. Use fdisk (or some similar utility) to make a new partition
2. Set the partition type to swap (82 in fdisk).
3. Use the mkswap command:
   - mkswap <options> partition
   - `mkswap -v1 /dev/hda7`
4. Edit /etc/fstab to reflect the new swap space
5. Turn on the swap (or reboot if you editted /etc/fstab) with the swapon command
   - swapon <options>
   - `swapon -a` (reads fstab and turns on all the swap partitions listed)
6. Check the status (optional)
   - `swapon -s`
7. Limitations to swap space?

Linux System Administration

Filesystems (Continued)

Partitioning Scheme

- The original partitioning scheme for PC hard disks allowed only four partitions.
  - This quickly turned into a bad idea.

/dev/hda

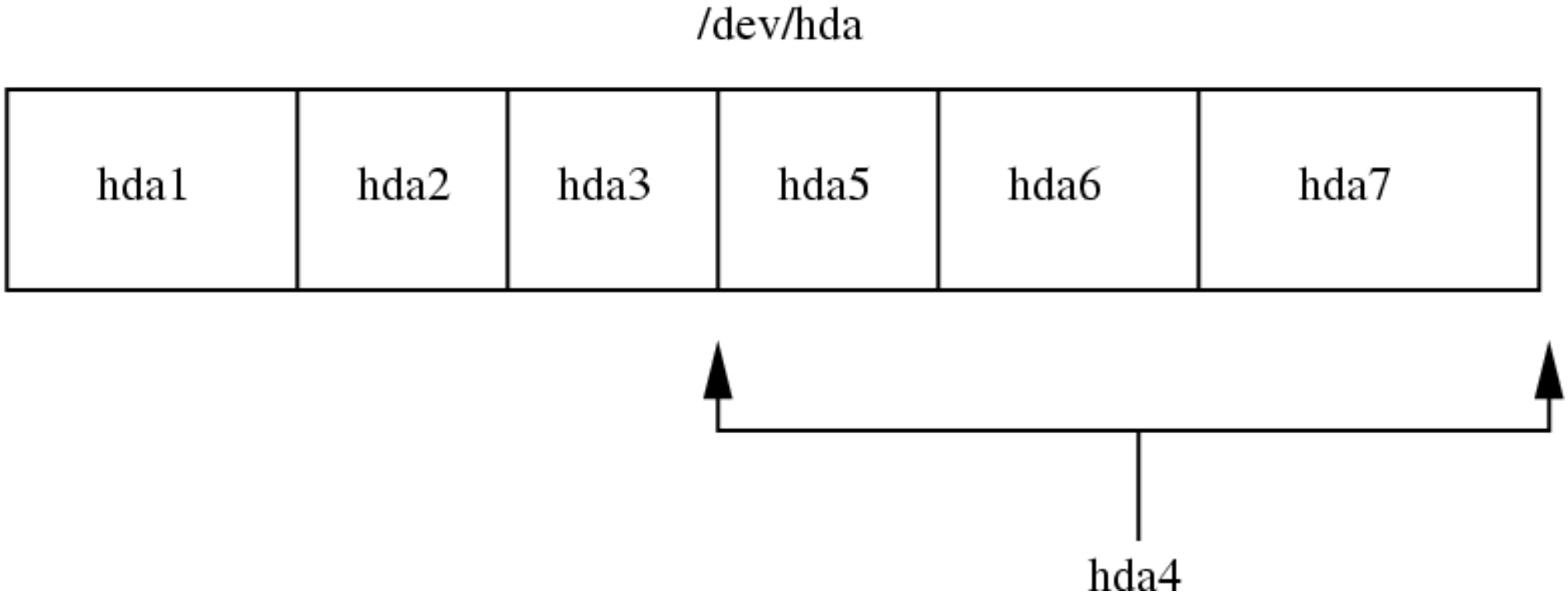
hda1	hda2	hda3	hda4

- The partitioning scheme is not built into the hardware, or even into the BIOS. It is only a convention that many operating systems follow.

Linux System Administration

Filesystems (Continued)

- In order to overcome the problem of only four partitions being allowed, *extended partitions* were created.
- This allows you to break up a primary partition into smaller partitions.
  - This primary partition is called the extended partition
  - These smaller partitions contained within the extended partition are called *logical partitions*.



- In this setup,
  - hda1, hda2, hda3 are primary partitions
  - hda4 is an extended partition
  - hda5m hda6, hda7 are logical partitions.

Linux System Administration

Filesystems (Continued)

- How many partitions can we have?
  - SCSI : 15
  - IDE : 63
- Why would we want multiple partitions and not just / ?
  - Protection from Attacks
  - Protection from Corrupted Filesystems
    - fsck can help repair corrupted filesystesm.
    - Unmount the partition, run fsck, and then mount the partition again.
    - If any files are missing, look in the **`lost & found`** directory in each filesystem.

Linux System Administration

Filesystems (Continued)

RAID
(Redundant Array of Inexpensive/Independent Disks)

- RAID uses two or more hard disk drives or partitions in combination with one another.
- The purpose is to improve fault tolerance and/or performance.
  - RAID can provide data redundancy.
  - RAID can increase read/write speed and throughput.
- Applications and utilitites see the multiple drives or partitions as a single logical device.
- RAID can be implemented in either hardware or software.

Linux System Administration

Filesystems (Continued)

- RAID Level 0 (Striping)
  - Improves performance, but offers no redundancy.
  - The storage capacity of the RAID device is equal to the sum of the partitions in the RAID.
  - The data is "striped" across the partitions.
  - This increases the performance by spreading the hits on the filesystems across multiple partitions, which are usually spread across several hard drives.

Linux System Administration

Filesystems (Continued)

- RAID Level 1 (Mirroring)
  - Provides simple redundancy.
  - Improves data reliability
  - Can improve the performance of read-intensive applications.
  - The storage capacity is equal to one of the partitions in the RAID.
  - This allows for data recovery if one of the disks fails. There is a copy on another disk.
  - This is continuously maintained, so one partition is a mirror image of another.

Linux System Administration

Filesystems (Continued)

- RAID Level 5 (Disk Striping with Parity)
  - Provides redundancy and improves performance.
  - The storage capacity of the RAID device is equal to that of the member partitions, minus one of the partitions.
  - If there are n partitions being used, then (n - 1) are used for striping, and 1 is used for parity. That way, if one disk fails, the backup disk can help recover lost data.
  - You need at least three disks for this RAID level.

Linux System Administration

Filesystems (Continued)

RAID Summary

- RAID Level 0 (Disk Striping with Parity)
  - Use RAID 0 to combine smaller drives into one large virtual drive.
  - Best Read/Write performance of all the schemes listed here.
  - No protection from drive failure.
  - ADVICE: Buy very reliable hard disk drives if you plan to use this scheme.

- RAID Level 1 (Mirroring)
  - Good read/write performance
  - Inefficient use of storage space (half the total space available for data)
  - Best protection from drive failure.

- RAID Level 5 (Disk Striping with Parity)
  - Protection against single drive failure.
  - Use RAID 5 if you need to make the best use of your available storage space while gaining protection against single drive failure.

Linux System Administration

Filesystems (Continued)

RAID Configuration

- If you know that you want a RAID setup when you are doing an installation, this can be done much easier during the installation process.
- There is a nice web page detailing this process during installation.

http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/custom-guide/ch-software-raid.html

Chapter 10. Software RAID Configuration

Read Chapter 3 *Redundant Array of Independent Disks (RAID)* first to learn about RAID, the differences between Hardware and Software RAID, and the differences between RAID 0, 1, and 5.

Software RAID can be configured during the graphical installation of Red Hat Linux or during a kickstart installation. This chapter discusses how to configure software RAID during installation, using the **Disk Druid** interface.

Before you can create a RAID device, you must first create RAID partitions, using the following step-by-step instructions:

1. On the **Disk Partitioning Setup** screen, select **Manually partition with Disk Druid**.
2. In **Disk Druid**, choose **New** to create a new partition.
3. You will not be able to enter a mount point (you will be able to do that once you have created your RAID device).
4. Choose **software RAID** from the **File System Type** pulldown menu as shown in Figure 10-1.

   Figure 10-1. Creating a New RAID Partition
5. For **Allowable Drives**, select the drive(s) on which RAID will be created. If you have multiple drives, all drives will be selected here and you must deselect those drives which will *not* have the RAID array on them.
6. Enter the size that you want the partition to be.
7. Select **Fixed size** to make the partition the specified size, select **Fill all space up to (MB)** and enter a size in MBs to give range for the partition size, or select **Fill to maximum allowable size** to make it grow to fill all available space on the hard disk. If you make more than one partition growable, they will share the available free space on the disk.
8. Select **Force to be a primary partition** if you want the partition to be a primary partition.
9. Select **Check for bad blocks** if you want the installation program to check for bad blocks on the hard drive before formatting it.
10. Click **OK** to return to the main screen.

Repeat these steps to create as many partitions as needed for your RAID setup. Notice that all the partitions do not have to be RAID partitions. For example, you can configure only the `/home` partition as a software RAID device.

Once you have all of your partitions created as **software RAID** partitions, follow these steps:

1. Select the **RAID** button on the **Disk Druid** main partitioning screen (see Figure 10-3).
2. Next, Figure 10-2 will appear, where you can make a RAID device.

Figure 10-2. Making a RAID Device

3. Enter a mount point.
4. Choose the file system type for the partition.
5. Select a device name such as **md0** for the RAID device.
6. Choose your RAID level. You can choose from **RAID 0**, **RAID 1**, and **RAID 5**.

> Note **Note**
>
> If you are making a RAID partition of `/boot`, you must choose RAID level 1, and it must use one of the first two drives (IDE first, SCSI second). If you are not creating a RAID partition of `/boot`, and you are making a RAID partition of `/`, it must be RAID level 1 and it must use one of the first two drives (IDE first, SCSI second).

7. The RAID partitions you just created appear in the **RAID Members** list. Select which partitions of these partitions should be used to create the RAID device.
8. If configuring RAID 1 or RAID 5, specify the number of spare partitions. If a software RAID partition fails, the spare will automatically be used as a replacement. For each spare you want to specify, you must create an additional software RAID partition (in addition to the partitions for the RAID device). In the previous step, select the partitions for the RAID device and the partition(s) for the spare(s).
9. After clicking **OK**, the RAID device will appear in the **Drive Summary** list as shown in Figure 10-3. At this point, you can continue with your installation process. Refer to the ***Red Hat Linux Installation Guide*** for further instructions.

Figure 10-3. RAID Array Created

Linux System Administration

Filesystems (Continued)

RAID Configuration - MDADM

- **mdadm** is a program that can be used to create, manage, and monitor MD devices.
- **mdadm** is a single program and not a collection of programs.
- **mdadm** can perform (almost) all of its functions without having a configuration file and does not use one by default.
- **mdadm** helps with management of the configuration file.
- **mdadm** can provide information about your arrays (through Query, Detail, and Examine)
- See the man pages for details as to the options available.

Linux System Administration

Filesystems (Continued)

RAID Configuration - MDADM
http://www.networknewz.com/2003/0113.html
Derek Vadala - Managing RAID on Linux

- **mdadm** has five major modes of operation:
  - **Create** and **Assemble,** are used to configure and activate arrays.
  - **Manage** mode is used to manipulate devices in an active array.
  - **Follow** (or Monitor) mode allows administrators to configure event notification and actions for arrays.
  - **Build** mode is used when working with legacy arrays that use an old version of the md driver.

- **mdadm** commands take the format:

mdadm [mode] [options]

Linux System Administration

Filesystems (Continued)

RAID Configuration - Creating An Array

- Create (mdadm --create) mode is used to create a new array.
- In this example we will use mdadm to create a RAID-0 at /dev/md0 made up of /dev/hdb1 and /dev/hdc1

```
mdadm --create --verbose /dev/md0 --level=0  --raid-
            devices=2 /dev/hdb1 /dev/hdc1
```

```
mdadm -Cv /dev/md0 -l0 -n2 -c128 /dev/hdb1 /dev/hdc1
```

Note : Default chunk size is 64kb

Linux System Administration

Filesystems (Continued)

RAID Configuration - Creating An Array

- Here is another example that will create a RAID-5 at /dev/md0 made up of /dev/hda6, /dev/hda7, /dev/hda8, /dev/hda9, and /dev/hda10

```
mdadm -Cv /dev/md0 -l5 -n5 -c128 /dev/hda{6,7,8,9,10}
```

- The command to stop a running array:

```
mdadm -S /dev/md0
```

Linux System Administration

Filesystems (Continued)

RAID Configuration - mdadm.conf

- `/etc/mdadm.conf` is mdadms' primary configuration file.
- `mdadm` does not rely on `/etc/mdadm.conf` to create or manage arrays.
  - mdadm.conf is simply an extra way of keeping track of software RAIDs.
  - Using a configuration file with `mdadm` is useful, but not required.
  - Having a configuration file means you can quickly manage arrays without spending extra time figuring out what array properties are and where disks belong.
- `mdadm.conf` is concise and simply lists disks and arrays.

Linux System Administration

Filesystems (Continued)

RAID Configuration - mdadm.conf

- The configuration file can contain two types of lines each starting with either the DEVICE or ARRAY keyword.
- Whitespace separates the keyword from the configuration information.
  - DEVICE lines specify a list of devices that are potential member disks.
  - ARRAY lines specify device entries for arrays as well as identifier information.
    - This information can include lists of one or more UUIDs, md device minor numbers, or a listing of member devices.

- Sample:

```
DEVICE      /dev/sda1 /dev/sdb1 /dev/sdc1 /dev/sdd1
ARRAY       /dev/md0 devices=/dev/sda1,/dev/sdb1
ARRAY       /dev/md1 devices=/dev/sdc1,/dev/sdd1
```

Linux System Administration

Filesystems (Continued)

RAID Configuration - mdadm.conf

- In general, it's best to create an `/etc/mdadm.conf` file after you have created an array and update the file when new arrays are created.
- Without an `/etc/mdadm.conf` file you'd need to specify more detailed information about an array on the command in order to activate it.
- `mdadm` even provides an easy way to generate ARRAY lines:

```
$  mdadm --detail --scan
   ARRAY /dev/md0 level=raid0 num-devices=2
UUID=410a299e:4cdd535e:169d3df4:48b7144a
```

- So after you're done building arrays you could copy the output of `mdadm --detail --scan` to `/etc/mdadm.conf`.
  - You manually create a DEVICE entry as well.
- Based on the example above, here is what the /etc/mdadm.conf file might look like:

```
DEVICE    /dev/sdb1 /dev/sdc1

ARRAY     /dev/md0 level=raid0 num-devices=2 UUID=410a299e:4cdd535e:169d3df4:48b7144a
```

Linux System Administration

Filesystems (Continued)

RAID Configuration - Starting an Array

- Assemble mode is used to start an array that already exists.
- If you created an `/etc/mdadm.conf` you can automatically start an array listed there with the following command:

  $ mdadm -As /dev/md0

  mdadm: /dev/md0 has been started with 2 drives.

- The -A option denotes **assemble** mode. You can also use --assemble.
- The -s or --scan option tells `mdadm` to look in `/etc/mdadm.conf` for information about arrays and devices.
- If you want to start every array listed in `/etc/mdadm.conf`, don't specify an md device on the command line.

Linux System Administration

Filesystems (Continued)

RAID Configuration - Starting an Array

- If you didn't create an `/etc/mdadm.conf` file, you will need to specify additional information on the command line in order to start an array.
- This command attempts to start /dev/md0 using the devices listed on the command line:

```
$ mdadm -A /dev/md0 /dev/sdb1 /dev/sdc1
```

- Using mdadm -A in this way assumes you have some prior knowledge about how arrays are arranged.
  - It might not be useful on systems that have arrays that were created by someone else.
- You may wish to examine some devices to gain a better picture about how arrays should be assembled.
  - The examine options (-E or --examine) allows you to print the md superblock (if present) from a block device that could be an array component.

Linux System Administration

Filesystems (Continued)

RAID Configuration - Starting an Array

$ mdadm -E /dev/sdc1

/dev/sdc1:

Magic : a92b4efc

Version : 00.90.00

UUID : 84788b68:1bb79088:9a73ebcc:2ab430da

Creation Time : Mon Sep 23 16:02:33 2002

Raid Level : raid0

Device Size : 17920384 (17.09 GiB 18.40 GB)

Raid Devices : 4

Total Devices : 4

Preferred Minor : 0

Update Time : Mon Sep 23 16:14:52 2002

State : clean, no-errors

Active Devices : 4

Working Devices : 4

Failed Devices : 0

Spare Devices : 0

Checksum : 8ab5e437 - correct

Events : 0.10

Chunk Size : 128K

Number Major Minor RaidDevice State

1 8 33 1 active sync /dev/sdc1

0 0 8 17 0 active sync /dev/sdb1

1 1 8 33 1 active sync /dev/sdc1

2 2 8 49 2 active sync /dev/sdd1

3 3 8 65 3 active sync /dev/sde1

- In this case we can tell that `/dev/sdc1` belongs to a RAID-0 made up of a total of four member disks.
- A UUID is a 128-bit number that is guaranteed to be reasonably unique on both the local system and across other systems.
    - It is a randomly generated using system hardware and timestamps as part of its seed.
- When an array is created, the md driver generates a UUID for the array and stores it in the md superblock.
- You can use the UUID as criteria for array assembly.

Linux System Administration

Filesystems (Continued)

RAID Configuration - Starting an Array

- Here is how you can activate the array to which `/dev/sdc1` belongs using its UUID.

```
  $ mdadm -Av /dev/md0 --
uuid=84788b68:1bb79088:9a73ebcc:2ab430da /dev/sd*
```

- This command scans every SCSI disk (/dev/sd*) to see if it's a member of the array with the UUID 84788b68:1bb79088:9a73ebcc:2ab430da and then starts the array, assuming it found each component device.

- `mdadm` will produce a lot of output each time it tries to scan a device that does not exist. You can safely ignore such warnings.

Linux System Administration

Filesystems (Continued)

RAID Configuration - Managing Arrays

- Using Manage mode you can add and remove disks to a running array.
- This is useful for removing failed disks, adding spare disks, or adding replacement disks.
- Manage mode can also be used to mark a member disk as failed.
- For example, to add a disk to an active array:

```
$ mdadm /dev/md0 --add /dev/sdc1
```

- To remove /dev/sdc1 from /dev/md0 :

```
$ mdadm /dev/md0 --fail /dev/sdc1 --remove /dev/sdc1
```

- You can combine commands on one line as above.
  - Make sure the order of the commands makes sense.
  - For instance, you have to fail a disk before removing it.

Linux System Administration

Filesystems (Continued)

RAID Configuration - Monitoring Arrays

- Using Follow/Monitor mode you can `daemonize mdadm` and configure it to send email alerts to system administrators when arrays encounter errors or fail.
- You can also use Follow mode to arbitrarily execute commands when a disk fails.
    - For example, you might want to try removing and reinserting a failed disk in an attempt to correct a non-fatal failure without user intervention.
- The following command will monitor /dev/md0 (polling every 300 seconds) for critical events.
    - When a fatal error occurs, mdadm will send an email to sysadmin.
    - You can tailor the polling interval and email address to meet your needs.

```
$ mdadm --monitor --mail=sysadmin --delay=300 /dev/md0
```

- Note that when using monitor mode, `mdadm` will not exit, so you might want to wrap it around `nohup` and `ampersand`:

```
$ nohup mdadm --monitor --mail=sysadmin --delay=300 /dev/md0 &
```

Linux System Administration

Filesystems (Continued)

RAID Configuration - Monitoring Arrays

- Follow/Monitor mode also allows arrays to share spare disks.
  - That means you only need to provide one spare disk for a group of arrays or for all arrays.
  - It also means that system administrators don't have to manually intervene to shuffle around spare disks when arrays fail.
- When Follow/Monitor mode is invoked, it polls arrays at regular intervals.
- When a disk failure is detected on an array without a spare disk, `mdadm` will remove an available spare disk from another array and insert it into the array with the failed disk.
- To facilitate this process, each ARRAY line in `/etc/mdadm.conf` needs to have a spare-group defined.

```
DEVICE    /dev/sd*
ARRAY     /dev/md0 level=raid1 num-devices=3 spare-group=database
 UUID=410a299e:4cdd535e:169d3df4:48b7144a
ARRAY    /dev/md1 level=raid1 num-device=2 spare-group=database
 UUID=59b6e564:739d4d28:ae0aa308:71147fe7
```

- Note that both `/dev/md0` and `/dev/md1` are part of the spare group: database.
  - The name does not have to be database, it could be anything.
  - Only groups with the same spare group name will switch out disks.
- Just assume that `/dev/md0` is a two-disk RAID-1 with a single spare disk.
  - If `mdadm` is running in monitor mode, and a disk in `/dev/md1` fails, `mdadm` will remove the spare disk from `/dev/md0` and insert it into `/dev/md1`

Linux System Administration

Logical Volume Manager (LVM)

- The Logical Volume Manager (LVM) enables you to resize your partitions without having to modify the partition tables on your hard disk.
- This is most useful when you find yourself running out of space on a filesystem and want to expand into a new disk partition versus migrating all or a part of the filesystem to a new disk.

Linux System Administration

Logical Volume Manager (LVM)

Terminology

- Physical Volume: A physical volume (PV) is another name for a regular physical disk partition that is used or will be used by LVM.

- Volume Group: Any number of physical volumes (PVs) on different disk drives can be lumped together into a volume group (VG). Under LVM, volume groups are analogous to a virtual disk drive.

- Logical Volumes: Volume groups must then be subdivided into logical volumes. Each logical volume can be individually formatted as if it were a regular Linux partition. A logical volume is, therefore, like a virtual partition on your virtual disk drive.

- Physical Extent: Real disk partitions are divided into chunks of data called physical extents (PEs) when you add them to a logical volume. PEs are important as you usually have to specify the size of your volume group not in gigabytes, but as a number of physical extents.

Linux System Administration

Logical Volume Manager (LVM)

- Since we do not have more than one drive, we need to create partitions that we will join together to form the volume group. Each of the partitions are part of the physical volume.

- Use fdisk to create the partitions. Use type 8e to represent a logical volume.

```
[root@localhost ~]# fdisk -l

Disk /dev/hda: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id
System
/dev/hda1   *           1        4865    39078081    c  W95
FAT32 (LBA)

Disk /dev/hdb: 61.4 GB, 61492838400 bytes
255 heads, 63 sectors/track, 7476 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id
System
/dev/hdb1   *           1          13      104391   83
Linux
/dev/hdb2              14        1925    15358140   83
Linux
/dev/hdb3            1926        2690     6144862+  83
Linux
/dev/hdb4            2691        7476    38443545    f  W95
Ext'd (LBA)
/dev/hdb5            2691        2958     2152678+  82
Linux swap
/dev/hdb6            2959        7476    36290803+  8e
Linux LVM
```

- Make sure you reboot or use partprobe after writing the new partition table.

Linux System Administration

Logical Volume Manager (LVM)

- Imagine we are going to try to combine partitions /dev/hda7, /dev/hda8, and /dev/hda9 into our volume group.
- We need to first create each physical volume.
  - This is done with the `pvcreate` command.

```
sh-2.05b# pvcreate /dev/hda7

pvcreate -- physical volume "/dev/hda7" successfully
created

sh-2.05b# pvcreate /dev/hda8

pvcreate -- physical volume "/dev/hda7" successfully
created

sh-2.05b# pvcreate /dev/hda9

pvcreate -- physical volume "/dev/hda7" successfully
created
```

Linux System Administration

Logical Volume Manager (LVM)

- The next step is to make Linux scan for any new LVM disk partitions and automatically create the LVM configuration files in the /etc directory.
- To do this, use the `vgscan` command.

```
sh-2.05b# vgscan

vgscan -- reading all physical volumes (this may take a while...)

sh-2.05b#
```

- We have now finished creating the Physical Volumes.

Linux System Administration

Logical Volume Manager (LVM)

- We now need to create the Volume Group.
- Use the `vgcreate` command to combine the physical volumes into a single unit called a volume group.
- The LVM software effectively tricks the operating system into thinking the volume group is a new hard disk.
- In the example, the volume group is called lvm-hdb.

```
sh-2.05b# vgcreate lvm-hdb /dev/hda7
/dev/hda8 /dev/hda9

                Volume group "lvm-hdb" successfully created

                sh-2.05b#
```

- Therefore, the `vgcreate` syntax uses the name of the volume group as the first argument followed by the partitions that it will be comprised of as all subsequent arguments.

Linux System Administration

Logical Volume Manager (LVM)

- The next step is to create a Logical Volume from the Volume Group.
- We can partition the volume group into logical volumes with the `lvcreate` command.
- While hard disks are divided into blocks of data, logical volumes are divided into units called *physical extents* (PEs).
- You'll have to know the number of available PEs before creating the logical volume.
- This is done with the `vgdisplay` command using the new `lvm-hdb` volume group as the argument.

```
sh-2.05b# vgdisplay lvm-hdb
--- Volume group ---
VG Name               lvm-hdb
VG Access             read/write
VG Status             available/resizable
VG #                  0
MAX LV                256
Cur LV                0
Open LV               0
MAX LV Size           255.99 GB
Max PV                256
Cur PV                2
Act PV                2
VG Size               848 MB
PE Size               4 MB
Total PE              212
Alloc PE / Size       0 / 0
Free  PE / Size       212 / 848 MB
VG UUID               W7bgLB-lAFW-wtKi-wZET-jDJF-8VYD-snUaSZ

sh-2.05b#
```

- As you can see, 212 PEs are available as free.
- We can now use all 212 of them to create a logical volume named `lvm0` from volume group `lvm-hdb`.

```
sh-2.05b# lvcreate -l 212 lvm-hdb -n lvm0

Logical volume "lvm0" created

sh-2.05b#
```

Prev Page 7 Next

Linux System Administration

Logical Volume Manager (LVM)

- After the logical volume is created, you can format it as if it were a regular partition.

```
sh-2.05b# mke2fs -j /dev/lvm-hdb/lvm0

mke2fs 1.32 (09-Nov-2002)

Filesystem label=

OS type: Linux

Block size=4096 (log=2)

Fragment size=4096 (log=2)

108640 inodes, 217088 blocks

10854 blocks (5.00%) reserved for the super user

First data block=0

7 block groups

32768 blocks per group, 32768 fragments per group

15520 inodes per group

Superblock backups stored on blocks:

        32768, 98304, 163840

Writing inode tables: done

Creating journal (4096 blocks): done

Writing superblocks and filesystem accounting information: done


This filesystem will be automatically checked every 38 mounts
or

180 days, whichever comes first.  Use tune2fs -c or -i to
override.

sh-2.05b#
```

Linux System Administration

Logical Volume Manager (LVM)

- We need to create a mount point.

```
sh-2.05b# mkdir /NEW
```

- Don't forget to update `/etc/fstab`.

```
/dev/lvm-hdb/lvm0   /NEW       ext3
defaults         1 2
```

- The `/dev/hda7, /dev/hda8` and `/dev/hda9` partitions are replaced by the combined /lvm0 logical volume.
- You, therefore, don't want the old partitions to be mounted again.
- Make sure that any reference to them in this file has either been commented a # character at the beginning of each line or deleted entirely.

- We can now mount the directory.
- The `mount -a` command reads the `/etc/fstab` file and mounts all the devices that haven't been mounted already.

```
sh-2.05b# mount -a

sh-2.05b# ls /home

lost+found

sh-2.05b#
```

Linux System Administration

Logical Volume Manager (LVM)

- We now have our LVM set up, so how do we resize the partitions?
- We can either declare how much space to add or we could declare how big we want the drive to be.
  - For instance, if the drive is set at 8GB, we could say:
    - add 2GB to the drive or
    - grow the drive to 10GB.

```
# lvextend -L10G /dev/lvm-hdb/lvm0
lvextend -- extending logical volume
"/dev/lvm-hdb/lvm0" to 12 GB
                lvextend -- doing automatic backup of
volume group "lvm-hdb"
                lvextend -- logical volume "/dev/lvm-
hdb/lvm0" successfully extended
```

    - The command above will extend the drive to 10GB
    - The command below will add 2GB to the drive:

```
                # lvextend -L+2G /dev/lvm-
hdb/lvm0
                lvextend -- extending logical
volume "/dev/lvm-hdb/lvm0" to 10GB
                lvextend -- doing automatic
backup of volume group "lvm-hdb"
                lvextend -- logical volume
"/dev/lvm-hdb/lvm0" successfully extended
```

Linux System Administration

Logical Volume Manager (LVM)

- After you have extended the logical volume it is necessary to increase the file system size to match.
- By default, most file system resizing tools will increase the size of the file system to be the size of the underlying logical volume so you don't need to worry about specifying the same size for each of the two commands.
- Unless you have patched your kernel with the `ext2online` patch it is necessary to unmount the file system before resizing it.

```
# umount /dev/lvm-hdb/lvm0
# resize2fs /dev/lvm-hdb/lvm0
# mount /dev/lvm-hdb/lvm0 /NEW
```

Linux System Administration

Logical Volume Manager (LVM)

- Similarly logical volumes can be reduced by using the following command:

```
            # lvreduce -L-1G /dev/lvm-hdb/lvm0
            lvreduce -- -Warning: reducing active
logical volume to 2GB
            lvreduce- -- This may destroy your data
(filesystem etc.)
            lvreduce -- -do you really want to reduce
"/dev/lvm-hdb/lvm0"? [y/n]: y
            lvreduce- -- doing automatic backup of
volume group "test_lvm"
            lvreduce- -- logical volume "/dev/lvm-
hdb/lvm0" successfully reduced
```

- It is very important to remember to reduce the size of the file system or whatever is residing in the volume before shrinking the volume itself

Linux System Administration

Intermediate Commands

`more`

- `more` will allow you to see the contents of a file, one screenful at a time.
- Press the space bar to go forward one screen.
- Press the return key to go forward one line.
- Press the '`b`' key to move back one screenful.
- Press the '`q`' key to quit `more`.

Linux System Administration

Intermediate Commands

`less`

- The `less` command performs the same as the more command, but it is more powerful and more widely used.
  - You can move forward and backwards by a line, window, or half-window.
  - `less` also has pattern matching.
  - See `less --help` for the options.

Linux System Administration

Intermediate Commands

`grep`

- grep prints the lines from a file or input stream that match an expression.

```
grep sshd /var/log/messages
```

  - This command will look for the phrase sshd in the file /var/log/messages
- You can use regular expressions with `grep`.
- grep options:
  - `grep -i` --> case-insensitive matches
  - `grep -v` --> inverts search; shows lines that don't match.
  - see the man pages for more options.

Linux System Administration

Intermediate Commands

---

pwd

- pwd --> Print Working Directory
- This command will remind you which directory you are currently in.

Prev Page 4 Next

Linux System Administration

Intermediate Commands

`diff`

- This command shows the differences between two text files.

```
diff  file1 file2
```

- There are several options that can control the format of the output.
- The default option is often most comprehensible by human beings.

Linux System Administration

Intermediate Commands

`file`

- The file command will give you the best guess as to the type (format) of file you are investigating.

`file  filename`

- While this command may not seem too useful, it can be quite handy if you encounter a file that you are unsure of.
- Remember that the extension of the file is not always indicative of the type of the file, especially if your system has been compromised.

Linux System Administration

Intermediate Commands

`find`

- This command will look for a filename in a directory and any subsequent child directories.

```
find  dir  -name  filename
```

  - `find /home/mlevan -name "*.html"`
  - `find . -name "*.*"`
  - `find . -name "*"`

- See the man pages for more options.

Prev Page 7 Next

Linux System Administration

Intermediate Commands

`head` and `tail`

- The head command shows the first ten lines of a text file.

```
head /etc/inittab
```

- The tail command shows the last ten lines of a text file.

```
tail /var/log/messages
```

- You can change the number of lines to print with the -n option:

```
head -n filename
```

```
head -20 /etc/inittab
```

- You can keep a viewing the changes with the -f (follow) option:

```
tail -f /var/log/messages
```

Linux System Administration

Intermediate Commands

`sort`

- The `sort` command quickly puts the lines of a text file in alphanumeric order.
- If the file's lines start with numbers, and you want to sort in numeric order, use the `-n` option.
- Use the `-r` option if you want to reverse the order of the sort.

Linux System Administration

Intermediate Commands

`passwd`

- Use the `passwd` command to change your password.
  - The command asks you for your old password and then prompts you for the new password (twice).
  - If you are the super-user, then
    - You can change any password using the form : `passwd username`
    - You are not prompted for the old password.
  - Passwords are stored in `/etc/shadow` if shadow passwords are enabled (and they should be!).
  - Passwords can contain uppercase and lowercase letters, the digits 0 through 9, and punctuation marks.
  - Passwords are case sensitive.

Linux System Administration

Intermediate Commands

---

ps

- The `ps` command will show you a current snapshot of all the processes.
- `ps --help` will show you the different options you can use for this command.
- My favorite : ps -aux
  - a --> all processes
  - u --> user oriented
  - x --> processes without controlling ttys
- `pstree`

Linux System Administration

Intermediate Commands

---

kill

- The kill command will kill a process.

kill pid

  - This kill command will try to stop the program normally.
  - The kill -9 pid will force the process to stop.

Linux System Administration

Intermediate Commands

`top`

- The `top` command provides a dynamic real-time view of a running system.
- It can display system summary information as well as a list of tasks currently being managed by the Linux kernel.
- `top` displays PID, User, CPU usage, memory usage, and more.
- You can kill a process through top with the k command and the PID.

Linux System Administration

Intermediate Commands

Virtual Consoles

- You can get 5 virtual consoles by typing CNRTL+ALT+F2, CNTRL+ALT+F3, ... , CNTRL+ALT+F6
  - These consoles will be tty2, tty3, tty4, tty5, tty6
- CNTRL+ALT+F7 will bring you back to your original X session.
  - This console is tty1
- You can not start a different X session from these extra virtual consoles.
- If you are not sure of your console, the command fgconsole will let you know.

Linux System Administration

Intermediate Commands

---

clear

- `clear` will clear the current console by placing the prompt at the top of the console.

Linux System Administration

Intermediate Commands

`alias`

- The `alias` command will allow you to create "shortcuts" for types commands.

`alias  name=value`

  - alias cp='cp -i'
  - alias dir='ls -l'

- `alias` with no parameters will print out all the current aliases.

- You can place these in your ~/.bash_profile directory if you want to use an alias every time you open a console.

Linux System Administration

Intermediate Commands

`who` and `w`

- The `who` command will show you who is logged on to the system.

```
[mlevan@localhost IntermediateCommands]$ who
mlevan   tty1         Apr 26 14:13
mlevan   pts/0        Apr 26 14:13
mlevan   pts/1        May  8 16:41
root     pts/6        May  3 12:16
```

- The `w` command will show you who is logged on to the system and what they are doing.

```
[mlevan@localhost IntermediateCommands]$ w
 20:29:55 up 12 days,  6:17,  4 users,  load average:
2.02, 2.05, 2.07
USER     TTY      FROM              LOGIN@   IDLE
JCPU   PCPU WHAT
mlevan   tty1     -                 26Apr05 12days
0.79s  0.00s /bin/sh /usr/X11R6/bin/startx
mlevan   pts/0    -                 26Apr05 12days  0.00s
13.29s kdeinit: kded
mlevan   pts/1    -                 16:41    0.00s
0.15s  0.01s w
root     pts/6    -                 Tue12    5days
0.02s  0.02s -bash
```

Linux System Administration

Intermediate Commands

`su  or  su -`

- Run a shell with a substitute user.

```
su  username

su - username

su

su -
```

- Note that if you are a user trying to su to another user, then you will need to provide a password.
- If you are root trying to su into another user, then you do not need to provide a password.

Linux System Administration

Intermediate Commands

`whoami`

- This command will print the user name associated with the current effective user id.
  - Note that if the prompt begins with a #, then you are currently root.
  - If the prompt begins with a $, then you are a regular user.

Linux System Administration

Intermediate Commands

---

history

- This command will display the command history list with line numbers.
  - You can repeat commands using "!"
    - To run the 398th command in the list, type: !398

- You can combine this with grep to find certain commands:

```
history | grep ssh
```

Linux System Administration

Intermediate Commands

`switchdesk`

- This command will allow you to switch between desktop managers.
- Note that you will have to stop and restart you X-windows session in order for the change to take place.

Linux System Administration

User Management

- What does a user need in order to log on to and use the system?
  - An entry in /etc/passwd
  - An entry in /etc/shadow
  - A home directory.

- You can use the GUI `system-config-users` to set up users.

Linux System Administration

User Management

/etc/passwd

- The /etc/passwd file maps login names to user IDs.
- It has seven fields:
  - username, encrypted password or an x or a * or blank, User ID, Group ID, User's real name, User's home directory, User's shell.

```
[root@localhost UserManagement]# more /etc/passwd
   root:x:0:0:root:/root:/bin/bash
        bin:x:1:1:bin:/bin:/sbin/nologin
        daemon:x:2:2:daemon:/sbin:/sbin/nologin
        adm:x:3:4:adm:/var/adm:/sbin/nologin
        .
        .
        .
        xfs:x:43:43:X Font
Server:/etc/X11/fs:/sbin/nologin
        named:x:25:25:Named:/var/named:/sbin/nologin
        ntp:x:38:38::/etc/ntp:/sbin/nologin
        gdm:x:42:42::/var/gdm:/sbin/nologin
        mlevan:x:500:500:Mike
LeVan:/home/mlevan:/bin/bash
```

- If the second field is an 'x' then the password is encrypted in /etc/shadow.
- If the second file is a '*', then the user can not login.
- If the second field is blank, then there is no password on the account.
- No comment or blank lines are allowed in this file.
- Any user can view this file.

Linux System Administration

User Management

/etc/shadow

- As with the passwd file, each field in the shadow file is also separated with ":" colon characters, and are as follows:
  - Username, up to 8 characters. Case-sensitive, usually all lowercase. A direct match to the username in the /etc/passwd file.
  - Password, 13 character encrypted. A blank entry (eg. ::) indicates a password is not required to log in (usually a bad idea), and a "*" entry (eg. :*:) indicates the account has been disabled.
  - The number of days (since January 1, 1970) since the password was last changed.
  - The number of days before password may be changed (0 indicates it may be changed at any time)
  - The number of days after which password must be changed (99999 indicates user can keep his or her password unchanged for many, many years)
  - The number of days to warn user of an expiring password (7 for a full week)
  - The number of days after password expires that account is disabled
  - The number of days since January 1, 1970 that an account has been disabled
  - A reserved field for possible future use
- This file is only readable by root, so it is more secure than /etc/passwd

```
[root@localhost ~]# more /etc/shadow

root:$1$OpnHwjeB$xNCb/VqY9PQjyZoFp/eA11:12646:0:99999:7:::
bin:*:12646:0:99999:7:::
daemon:*:12646:0:99999:7:::
gdm:!!:12646:0:99999:7:::

mlevan:$1$oCxnR7Mg$3HHb/KZ9Kytr0eLLW0HF50:12646:0:99999:7:::
```

Linux System Administration

User Management

```
[root@Hamming ~]# system-config-users &
```

system-config-users

Prev Page 4 Next

Linux System Administration

User Management

- Adding a user:
  - Click on the **Add User** button on the toolbar.
  - Enter the information and click **OK**.

- Modifying a user:
  - Highlight the user in the User Manager window and click **Properties** on the toolbar.
  - The User properties window has four tabs : User Data, Account Info, Password Info, and Groups.
  - The **User Data** tab holds basic user information such as name and Groups.
  - The **Account Info** tab allows you to specify an expiration date for the account.
  - The **Account Info** tab also allows you to lock the account so the user can not log in.
  - The **Password Info** tab allows you to turn on password expiration and specify various parameters.
  - In the **Groups** tab, you can specify the groups that the user is a member of.

Linux System Administration

User Management

Working with Groups

- Click the **Groups** tab in the User Manager window to work with groups.
- To create a group, click **Add Group** on the toolbar and specify the name of the group.
- To change the name of the group or add or remove users from a group:
  - Highlight the group and click **Properties** on the toolbar.
  - Click the appropriate tab, make the changes you want, and click **OK**.

Linux System Administration

User Management

Command Line

- `useradd` : Adds a user account.
  - The useradd utility adds new users accounts to the system.
  - By default, useradd assigns the next highest unused user ID
  - useradd also specifies the bash as the user's login shell.

```
# useradd -g 500 -c "Mike LeVan" mlevan
```

  - This command will:
    - Create the user's home directory (in /home).
    - Specify the user's group ID.
    - Puts the user's full name in the comment field.

Linux System Administration

User Management

---

useradd

- Based on /etc/login.defs, the system creates a home directory for the new user.
- The contents of /etc/skel is copied to the home directory.
  - This contains bash and other startup files.
- Once you have added a user, use passwd to give the user a password.

Linux System Administration

User Management

`userdel`

- The following command will remove a user's account:

```
userdel -r username
```

- If appropriate, make a backup copy of the files belonging to the user before deleting the account.
- The userdel command will remove the account, the user's home directory, and all the files in the directory.

Linux System Administration

User Management

---

usermod

- This command can temporarily turn off a user's account.
  - You can change the expiration date for the account.

```
usermod -e "12/31/03" username
```

- This command will prevent the user from logging in.

Linux System Administration

User Management

---

groupadd

- This command will add a new group to **/etc/group**.
- The following will create a new group:

```
# groupadd -g 1024 groupname
```

- The **-g** option allows you to pick the group ID number.
- If you do not use this option, then the system picks the next available sequential number greater than 500.
- The **-o** option allows the group ID to be nonunique.
  - This allows multiple names for the same group ID.

Linux System Administration

User Management

/etc/group

- The /etc/group file has four fields in the following format:

group-name : password : group-ID : login-name-list

- group-name : The name of the group.
- password : Optional encrypted password. This is rarely used, and is usually an 'x'
- group-ID : A number, with 1 - 499 reserved for system accounts.
- login-name-list : a comma-separated list of users that belong to that group.

```
[root@localhost temp]# more /etc/group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
xfs:x:43:
named:x:25:
ntp:x:38:
gdm:x:42:
mlevan:x:500:
```

Linux System Administration

User Management

---

groupdel

- This command is analogous to userdel.
- This command will take a group name as an argument and remove the group.

```
# groupdel groupname
```

groupmod

- This command can change the name of the group or the group ID.

```
# groupmod -g 1025 groupname
# groupmod -n newgroupname oldgroupname
```

- This command does not change group numbers in /etc/passwd when you renumber a group.
  - You must edit /etc/passwd by hand and change the entry yourself.

- Note that if a file belongs to group X and you change group X to group Y, then that file either belongs to no group, or to another group with the old ID number.

Linux System Administration

User Management

Permissions

- Three types of users could access a file :
  - The owner of a file (*owner*)
  - a member of a group to which the owner belongs (*group*)
  - and everyone else (*other*)

- A user can attempt to access a file in one of three ways:
  - A user could *read* the file (*r*)
  - A user could *write* to the file (*w*)
  - A user could *execute* the file (*e*)

- This gives us nine possible ways to access an ordinary file. We need to determine how many of these nine possibilities we are going to allow to a specific file or directory.

Linux System Administration

User Management

Displaying Permissions

- Use `ls -l` to display the permissions of a file or directory.

```
[mlevan@localhost rhct]$ ls -l
total 84
drwxr-xr-x  3 mlevan mlevan  4096 Aug 13  2004 ks-floppy
-rwxr-xr-x  1 mlevan mlevan 77701 Aug 13  2004
Mike_LeVan-RHCT.pdf
```

- Consider the first 10 dashes :
  - The first dash tells you the type of file (directory, block, character, file)
  - The next three dash's represent the permissions for the owner.
    - The *w* represents write, the *x* represents execute, and the *r* means read.
  - The next three dash's represent the permissions for the group.
  - The next three dash's represent the permissions for all others.

Linux System Administration

User Management

Changing permissions

- The owner of a file controls which users have permission to access the file and how they can access it.
- You can use the chmod (change mode) command to change the permissions for that file.
- Some examples:

```
chmod o+rx filename

chmod g+w filename

chmod a+rwx filename

chmod o-x filename

chmod u-rx filename
```

- Note that o stands for others, g stands for group, and u stands for user (owner).
- Note that r stands for read, w stands for write, and x stands for execute.
- Note that the + symbol adds the permissions and the - symbol removes the permissions.

Linux System Administration

User Management

Changing Permissions

- This can also be done numerically:

Value	Meaning
0	---
1	--x
2	-w-
3	-wx
4	r--
5	r-x
6	rw-
7	rwx

- Consider the following file:

```
-rwxr-xr-x  1 mlevan mlevan 77701 Aug 13  2004 blah.txt
```

- This means the following two statements are equivalent:

```
chmod 775 blah.txt          chmod g+w blah.txt
```

Linux System Administration

User Management

Changing Ownership and Group of a File

- The command you need to change the ownership of a file is chown:

```
chown  newowner  filename
```

- The command you need to change the group of a file is chgrp:

```
chgrp  newgroup  filename
```

- Note that the owner of the file can use these commands.

Linux System Administration

User Management

Changing permissions, owners, and groups on Directories

- The commands we have been using

  chmod, chown, chgrp

  are also applicable to directories.

Linux System Administration

User Management

UMASK

- When a file is created, what are it's default permissions?
  - The default is 777 (!)

- This can be changed by setting the umask.
  - The umask decreases the permission number by the value stored in umask.
  - For example, if you wanted `-rwxr-xr-x` to be the default you would set the umask to be 022.
  - To get the new default, you subtract the umask from 777.

- The umask is set in `/etc/bashrc` for the system

- The umask can be set locally in ~/.bash_profile

Prev Page 20 Next

umask 044 = 7 7 7 RWX for Everyone
0 2 2 Subtract
---
7 5 5 RWX for User, RX for Groups, RX for Everyone

NOTE: This is assuming that your default permissions for a newly created file is 777.

Linux File System Quotas

Disk Quotas: This feature of Linux allows the system administrator to allocate a maximum amount of disk space a user or group may use. It can be flexible in its adherence to the rules assigned and is applied per filesystem. The default Linux Kernel which comes with Redhat and Fedora Core comes with quota support compiled in.

Two versions of quotas have been released. Version 2 is used by the Linux 2.4 and 2.6 kernel. Quotas version 1 is used by the Linux 2.2 kernel. Both are discussed in this tutorial.

Configuration:

Configuration of disk usage quotas on Linux - Perform the following as root:

1. Edit file `/etc/fstab` to add qualifier "usrquota" or "grpquota" to the partition. The following file system mounting options can be specified in `/etc/fstab`: grpquota, noquota, quota and usrquota. (These options are also accepted by the `mount` command but ignored.) The filesystem when mounted will show up in the file `/etc/mtab`, the list of all currently mounted filesystems.)

   - To enable user quota support on a file system, add "usrquota" to the fourth field containing the word "defaults".

```
...
/dev/hda2     /home   ext3    defaults,usrquota              1    1
...
```

   - Replace "usrquota" with "grpquota", should you need group quota support on a file system.

```
...
/dev/hda2     /home   ext3    defaults,grpquota              1    1
...
```

   - Need both user quota and group quota support on a file system?

```
...
/dev/hda2     /home   ext3    defaults,usrquota,grpquota     1    1
...
```

   This enables user and group quotas support on the `/home` file system.

2. `touch /partition/aquota.user`
   where the partition might be `/home` or some partition defined in `/etc/fstab`.
   then
   `chmod 600 /partition/aquota.user`

   The file should be owned by root. Quotas may also be set for groups by using the file `aquota.group`

   Quota file names:

   - Quota Version 2 (Linux 2.4/2.6 kernel: Red Hat 7.1+/8/9,FC 1-3): `aquota.user, aquota.group`
   - Quota Version 1 (Linux 2.2 kernel: Red Hat 6, 7.0): `quota.user, quota.group`

   The files can be converted/upgraded using the `convertquota` command.

3. Re-boot or re-mount file partition with quotas.
   - Re-boot: `shutdown -r now`
   - Re-mount partition: `mount -o remount /partition`

   After re-booting or re-mounting the file system, the partition will show up in the list of mounted filesystems as having quotas.
   Check `/etc/mtab`:

```
...
/dev/hda5 / ext3 rw,usrquota 0 0
...
```

4. `quotacheck -vgum /partition`
   or
   `quotacheck -vguma`
   - For example (Linux kernel 2.4+: Red Hat 7.1+, Fedora): `quotacheck -vguma`

```
quotacheck: WARNING -  Quotafile //aquota.user was probably truncated. ...
quotacheck: Scanning /dev/hda5 [/] done
```

```
quotacheck: Checked 9998 directories and 179487 files
```

- For example (Linux kernel 2.2: Red Hat 6/7.0): `quotacheck -v /dev/hda6`
  System response:

```
Scanning /dev/hda6 [/home] done
Checked 444 directories and 3136 files
Using quotafile /home/quota.user
```

Quotacheck is used to scan a file system for disk usages, and updates the quota record file "quota.user/aquota.user" to the most recent state. It is recommended thet quotacheck be run at bootup (part of Redhat default installation)

Man page: quotacheck - scan a filesystem for disk usage, create, check and repair quota files

5. `quotaon -av`
   System Response: `/dev/hda6: user quotas turned on`

   quotaon - enable disk quotas on a file system.
   quotaoff - turn off disk quotas for a file system.

   Man page: quotaon - turn filesystem quotas on and off

6. `edquota -u` *`user_id`*
   Edit directly using vi editor commands. (See below for more info.)

For example: `edquota -u user1`

- System Response (RH 7+):

```
Disk quotas for user user1 (uid 501):
  Filesystem                   blocks       soft       hard     inodes     soft
hard
  /dev/hda5                      1944          0          0        120        0
0
```

  - blocks: 1k blocks
  - inodes: Number of entries in directory file
  - soft: Max number of blocks/inodes user may have on partition before warning is issued and grace persiod countdown begins.
    If set to "0" (zero) then no limit is enforced.
  - hard: Max number of blocks/inodes user may have on partition.
    If set to "0" (zero) then no limit is enforced.

- System Response (RH 6):

```
Quotas for user user1:
/dev/sdb6: blocks in use: 56, limits (soft = 0, hard = 0)
        inodes in use: 50, limits (soft = 0, hard = 0)
```

Something failed if you get the response:

```
Quotas for user user1:
/dev/sdb6: blocks in use: 0, limits (soft = 0, hard = 0)
        inodes in use: 0, limits (soft = 0, hard = 0)
```

Edit limits:

```
Quotas for user user1:
/dev/hda6: blocks in use: 992, limits (soft = 50000, hard = 55000)
        inodes in use: 71, limits (soft = 10000, hard = 11000)
```

If editing group quotas: `edquota -g` *`group_name`*

Man page: edquota - edit user quotas

7. List quotas:
   `quota -u` *`user_id`*

   For example: `quota -u user1`
   System response:

```
Disk quotas for user user1 (uid 501):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/hda6     992   50000   55000              71   10000   11000
```

If this does not respond similar to the above, then restart the computer: `shutdown -r now`

Man page: quota - display disk usage and limits

Quota Reports:

- Report on all users over quota limits: `quota -q`
- Quota summary report: `repquota -a`

```
*** Report for user quotas on device /dev/hda5
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      -- 4335200       0       0         181502     0     0
bin       --   15644       0       0            101     0     0
...
user1     --    1944       0       0            120     0     0
```

No limits shown with this user as limits are set to 0.

Man page: repquota - summarize quotas for a filesystem.

Cron:

Quotacheck should scan the file system via cronjob periodically (say, every week?). Add a script to the /etc/cron.weekly/ directory.
File: /etc/cron.weekly/runQuotacheck

- Linux Kernel 2.4: Red Hat 7.1 - Fedora Core 3:

```
#!/bin/bash
/sbin/quotacheck -vguma
```

- Linux Kernel 2.2: Red Hat 6/7.0:

```
#!/bin/bash
/sbin/quotacheck -v -a
```

(Remember to `chmod +x /etc/cron.weekly/runQuotacheck`)

Edquota Note:

The "edquota" command puts you into a "vi" editing mode so knowledge of the "vi" editor is necessary. Another editor may be specified with the **EDITOR** environment variable. You are **NOT** editing the quota.user file directly. The */partition*/quota.user or quota.group file is a binary file which you do not edit directly. The command edquota gives you an ascii interface with the text prepared for you. When you ":wq" to save the file from the vi session, it is converted to binary by the edquota command and stored in the quota.user file.

Assigning quota for a bunch of users with the same value. To rapidly set quotas for all users, on my system to the same value as user user1, I would first edit user `user1`'s quota information by hand, then execute:

```
edquota -p user1 `awk -F: '$3 > 499 {print $1}' /etc/passwd`
```

This assumes that the user uid's start from 500 and increment upwards. "blocks in use" is the total number of blocks (in kilobytes) a user has comsumed on a partition. "inodes in use" is the total number of files a user has on a partition.

edquota options:

Option	Description
-r -m	Edit quotas on remote server using RPC. Remote server must be configured with the daemon `rpc.rquotad`
-u	Edit user quota
-g	Edit group quota
-p *user-id*	Duplicate the quotas based on existing prototype user
-F *format* -F vfsold -F vfsv0 -F rpc -F xfs	Format: vfsold - version 1 vfsv0 - version 2 rpc - quotas over NFS xfs - quotas for XFS filesystem
-f */file-system*	Perform on specified filesystem. Default is to apply on all filesystems with quotas

-t	Edit the soft time limits for each filesystem.
-T	Edit time for user/group when softlimit is enforced. Specify number and unit or "unset"

Soft Limit and Hard Limits:

Soft limit indicates the maximum amount of disk usage a quota user has on a partition. When combined with "grace period", it acts as the border line, which a quota user is issued warnings about his impending quota violation when passed. Hard limit works only when "grace period" is set. It specifies the absolute limit on the disk usage, which a quota user can't go beyond his "hard limit".

Grace Period:

"Grace Period" is configured with the command "`edquota -t`", "grace period" is a time limit before the "soft limit" is enforced for a file system with quota enabled. Time units of sec(onds), min(utes), hour(s), day(s), week(s), and month(s) can be used. This is what you'll see with the command "`edquota -t`":

System response:

- Linux Kernel 2.4+: Red Hat 7.1+/Fedora:

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem             Block grace period     Inode grace period
  /dev/hda5                     7days                  7days
```

- Linux Kernel 2.2: Red Hat 6/7.0:

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/dev/hda2: block grace period: 0 days, file grace period: 0 days
```

Change the 0 days part to any length of time you feel reasonable. A good choice might be 7 days (or 1 week).

Quota files: (non-XFS file systems)

The edquota command will create/edit the quota file at the root of the file system. (See `/etc/mtab` for the list of the currently mounted filesystems.)

- Version 2: `aquota.user, aquota.group`
- Version 1: `quota.user, quota.group`

The Linux Kernel:

The default Red Hat/Fedora Core Linux kernel is shipped quota ready. If you have streamlined your kernel by rebuilding it with fewer options, make sure it has been configured with quotas support. When using the tools xconfig or menuconfig be sure to reply y to:
`Quota support (CONFIG_QUOTA) [n] y`

Fedora Core 3: `grep CONFIG_QUOTA /usr/src/redhat/SOURCES/kernel-2.6.9-x86_64.config`
Response:

```
CONFIG_QUOTA=y
CONFIG_QUOTACTL=y
```

The Redhat default init script `/etc/rc.d/rc.sysinit` will also contain a point in the script to run quotacheck:

- Red Hat 6, 7.0:

```
if [ -x /sbin/quotacheck ]; then
        echo "Checking root filesystem quotas"
        /sbin/quotacheck -v -a
fi
```

And turn quota checking on:

```
if [ -x /usr/sbin/quotaon ] then
   echo "Turning on quota."
   /usr/sbin/quotaon -v -a
fi
```

Links/Information:

Also note that system limits may be set in the configuration file: `/etc/security/limits.conf`. Here file size limits may be set for core dumps and data files as well as resource limits such as max cpu time and number of processes.

More Quota Info:

- http://www.freebsd.org/handbook/quotas.html

Exploits:

http://listweb.bilkent.edu.tr/linux/06/0653.html

Software Available From:

ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/subsystems/quota/all.tar.gz

Linux man pages:

- quota - display disk usage and limits
- rquota - implement quotas on remote machines
- fstab - static information about the filesystems
- edquota - edit user quotas
- setquota - set disk quotas (Command line editor)
- quotacheck - scan a filesystem for disk usage, create, check and repair quota files
- quotaon - turn filesystem quotas on
- quotaoff - turn filesystem quotas off
- repquota - produce a summary of quota information for a file system
- convertquota - convert quota from old file format to new one. Convert quota.user to aquota.user
- quotactl - manipulate disk quotas (C programmer interface)

Return to http://YoLinux.com for more Linux links, information and tutorials
Return to YoLinux Tutorial Index

16.14 File System Quotas

Quotas are an optional feature of the operating system that allow you to limit the amount of disk space and/or the number of files a user or members of a group may allocate on a per-file system basis. This is used most often on timesharing systems where it is desirable to limit the amount of resources any one user or group of users may allocate. This will prevent one user or group of users from consuming all of the available disk space.

16.14.1 Configuring Your System to Enable Disk Quotas

Before attempting to use disk quotas, it is necessary to make sure that quotas are configured in your kernel. This is done by adding the following line to your kernel configuration file:

```
options QUOTA
```

The stock `GENERIC` kernel does not have this enabled by default, so you will have to configure, build and install a custom kernel in order to use disk quotas. Please refer to Chapter 8 for more information on kernel configuration.

Next you will need to enable disk quotas in `/etc/rc.conf`. This is done by adding the line:

```
enable_quotas="YES"
```

For finer control over your quota startup, there is an additional configuration variable available. Normally on bootup, the quota integrity of each file system is checked by the quotacheck(8) program. The quotacheck(8) facility insures that the data in the quota database properly reflects the data on the file system. This is a very time consuming process that will significantly affect the time your system takes to boot. If you would like to skip this step, a variable in `/etc/rc.conf` is made available for the purpose:

```
check_quotas="NO"
```

If you are running FreeBSD prior to 3.2-RELEASE, the configuration is simpler, and consists of only one variable. Set the following in your `/etc/rc.conf`:

```
check_quotas="YES"
```

Finally you will need to edit `/etc/fstab` to enable disk quotas on a per-file system basis. This is where you can either enable user or group quotas or both for all of your file systems.

To enable per-user quotas on a file system, add the *userquota* option to the options field in the `/etc/fstab` entry for the file system you want to enable quotas on. For example:

```
/dev/da1s2g   /home    ufs rw,userquota 1 2
```

Similarly, to enable group quotas, use the *groupquota* option instead of *userquota*. To enable both user and group quotas, change the entry as follows:

```
/dev/da1s2g    /home    ufs rw,userquota,groupquota 1 2
```

By default, the quota files are stored in the root directory of the file system with the names `quota.user` and `quota.group` for user and group quotas respectively. See fstab(5) for more information. Even though the fstab(5) manual page says that you can specify an alternate location for the quota files, this is not recommended because the various quota utilities do not seem to handle this properly.

At this point you should reboot your system with your new kernel. `/etc/rc` will automatically run the appropriate commands to create the initial quota files for all of the quotas you enabled in `/etc/fstab`, so there is no need to manually create any zero length quota files.

In the normal course of operations you should not be required to run the quotacheck(8), quotaon(8), or quotaoff(8) commands manually. However, you may want to read their manual pages just to be familiar with their operation.

16.14.2 Setting Quota Limits

Once you have configured your system to enable quotas, verify that they really are enabled. An easy way to do this is to run:

```
# quota -v
```

You should see a one line summary of disk usage and current quota limits for each file system that quotas are enabled on.

You are now ready to start assigning quota limits with the edquota(8) command.

You have several options on how to enforce limits on the amount of disk space a user or group may allocate, and how many files they may create. You may limit allocations based on disk space (block quotas) or number of files (inode quotas) or a combination of both. Each of these limits are further broken down into two categories: hard and soft limits.

A hard limit may not be exceeded. Once a user reaches his hard limit he may not make any further allocations on the file system in question. For example, if the user has a hard limit of 500 kbytes on a file system and is currently using 490 kbytes, the user can only allocate an additional 10 kbytes. Attempting to allocate an additional 11 kbytes will fail.

Soft limits, on the other hand, can be exceeded for a limited amount of time. This period of time is known as the grace period, which is one week by default. If a user stays over his or her soft limit longer than the grace period, the soft limit will turn into a hard limit and no further allocations will be allowed. When the user drops back below the soft limit, the grace period will be reset.

The following is an example of what you might see when you run the edquota(8) command. When the edquota(8) command is invoked, you are placed into the editor specified by the `EDITOR` environment variable, or in the **vi** editor if the `EDITOR` variable is not set, to allow you to edit the quota limits.

```
# edquota -u test

Quotas for user test:
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
        inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: kbytes in use: 0, limits (soft = 50, hard = 75)
        inodes in use: 0, limits (soft = 50, hard = 60)
```

You will normally see two lines for each file system that has quotas enabled. One line for the block limits, and one line for inode limits. Simply change the value you want updated to modify the quota limit. For example, to raise this user's block limit from a soft limit of 50 and a hard limit of 75 to a soft limit of 500 and a hard limit of 600, change:

```
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
```

to:

```
/usr: kbytes in use: 65, limits (soft = 500, hard = 600)
```

The new quota limits will be in place when you exit the editor.

Sometimes it is desirable to set quota limits on a range of UIDs. This can be done by use of the *-p* option on the edquota(8) command. First, assign the desired quota limit to a user, and then run `edquota -p protouser startuid-enduid`. For example, if user `test` has the desired quota limits, the following command can be used to duplicate those quota limits for UIDs 10,000 through 19,999:

```
# edquota -p test 10000-19999
```

For more information see edquota(8) manual page.

16.14.3 Checking Quota Limits and Disk Usage

You can use either the quota(1) or the repquota(8) commands to check quota limits and disk usage. The quota(1) command can be used to check individual user or group quotas and disk usage. A user may only examine his own quota, and the quota of a group he is a member of. Only the super-user may view all user and group quotas. The

repquota(8) command can be used to get a summary of all quotas and disk usage for file systems with quotas enabled.

The following is some sample output from the `quota -v` command for a user that has quota limits on two file systems.

```
Disk quotas for user test (uid 1002):
     Filesystem  usage    quota   limit   grace   files   quota   limit
grace
           /usr      65*     50      75   5days       7      50      60
       /usr/var       0      50      75                0      50      60
```

On the `/usr` file system in the above example, this user is currently 15 kbytes over the soft limit of 50 kbytes and has 5 days of the grace period left. Note the asterisk * which indicates that the user is currently over his quota limit.

Normally file systems that the user is not using any disk space on will not show up in the output from the quota(1) command, even if he has a quota limit assigned for that file system. The *-v* option will display those file systems, such as the `/usr/var` file system in the above example.

16.14.4 Quotas over NFS

Quotas are enforced by the quota subsystem on the NFS server. The rpc.rquotad(8) daemon makes quota information available to the quota(1) command on NFS clients, allowing users on those machines to see their quota statistics.

Enable `rpc.rquotad` in `/etc/inetd.conf` like so:

```
rquotad/1      dgram rpc/udp wait root /usr/libexec/rpc.rquotad rpc.rquotad
```

Now restart `inetd`:

```
# kill -HUP `cat /var/run/inetd.pid`
```

Prev: File System Snapshots | Home | Up | Next: Encrypting Disk Partitions

edquota prb.

Murat Arslan (*arslanm at arslanm dot linux-tr dot EU dot org*)
Fri, 17 Apr 1998 19:04:45 +0300 (EEST)

- **Messages sorted by:** [date][thread][subject][author]
- **Next message:** HorneT: "Re: [LINUX:1849] Apache server problemi"
- **Previous message:** Murat Arslan: "Re: [LINUX:1845] Re: quota problemi"

---------- Forwarded message ----------
Date: Sat, 21 Mar 1998 09:37:47 -0300
From: Solar Designer <solar at FALSE dot COM>
To: BUGTRAQ at NETSPACE dot ORG
Subject: edquota(8) feature

Hello,

Okay, at least two different bugs today, but let me start with a tiny FAQ:

Q: How do I crash a Linux-based shell provider?
A: Register with username "67108864".

Q: How do I just bypass the quota? My admin uses BSD-derived edquota(8).
A: Register as "65535".

Q: How do I consume some hours of their CPU time, as root?
A: Register as "12345678". The next quotacheck run (usually at reboot)
will take hours to complete.

Q: How do I reduce someone else's increased quota to the default?
A: Register with their UID as your username.
Q: How do I corrupt their quota.user file?
A: They have to allow 9 character long usernames for that. Read below.

Of these, only the first scenario is Linux-specific. Others apply to many
systems: BSD 4.3, BSDI 2.0, FreeBSD 2.2.5, SunOS 4.1.4, Solaris 2.5 seem to
be affected -- at least their edquota(8) got the "feature", too. I didn't
actually find that many victims yet, so feedback is welcome. ;-)

In general, only some setups are affected: [free] shell providers mostly.
Users should be allowed to pick all-digit usernames for these exploits to
work. However, the reason I was investigating this is that our quota.user
file grew 449 Megs large one day, so this _can_ happen.

Now, to the edquota feature (yes, this was meant to be a feature): it has
"special support" for all-digit usernames. Simply, it treats them as UIDs,
and I was unable to find a mention of this in the manpages I have. Other
user-level quota utilities have the same feature, but that doesn't seem to
be a security problem there. However, a typical (I think) ISP setup would
use edquota in a script, running as root, to set the quota for every new
user created.

While this feature by itself is a security problem (see the 2nd and 4th
questions above), things are even worse in reality. Only some versions of
edquota check and disallow negative UIDs, and none of those I've seen do
any check for UIDs past 65535.

Now, everything depends on the way quota file is updated. There're several
approaches here. Some versions of edquota will only work when the quota is
on at the moment, and use quotactl(2). Others first try to use quotactl(),
and, if that fails, assume the quota is off (some are wise enough to check
errno though), and write to the file directly. (Of those, many don't care
to check return value from lseek(), which brings a reliability problem, but
I won't go into that now.)

If our version of edquota supports direct quota file access, _and_ it is

run while the quota is off, then the attacker is probably lucky, since it will happily lseek() to whatever UID it got from the username.

Otherwise, everything depends on how well the kernel checks if the values passed to quotactl() are valid. Again, many systems seem to let the attacker succeed, perhaps thinking that they did the super-user check already. Some check for negative UIDs only, which is definitely not enough.

Let's assume the attacker succeeded in making the quota file really huge. What's the problem with this, it's just the filesize, and doesn't take that much of physical storage anyway? Still, there're several problems. First, some versions of quotacheck(8), which typically runs at reboot, got the following code:

```
if (fstat(fd, &st) == 0) {
max_id = st.st_size / sizeof(struct dqblk);
[...]
for (id = 1; id <= max_id; id++) {
```

That is, its execution time will increase with the file size. For 449 Megs, this was over 8 hours of CPU time.

Then, there's a problem when 9 character long usernames are allowed, _and_ sizeof(struct dqblk) is not a power of 2. Nine decimal digits are enough to cause an integer overflow when edquota (or the kernel) multiplies UID by sizeof(struct dqblk). This can be used to write a block not at a block boundary, corrupting the quota file.

Finally, there's a Linux kernel bug (might be present on some other systems also, I just didn't have a chance to check; the impact will likely differ though). There's no check whether the UID supplied via quotactl() is valid, so that it is possible to get negative file offsets. Now, if it used lseek() the way it is accessible via the syscall, everything would be fine. However, the kernel simply does:

```
filp->f_pos = dqoff(dquot->dq_id);
```

The system stops responding, and the console gets flooded with ext2 warning messages. Hopefully there's someone around to hit that reset button. The username from first FAQ question exploits exactly this bug (combined with the edquota feature, of course). Here's another exploit, just to show this specific problem:

```
#include <stdio.h>
#include <unistd.h>
#include <linux/quota.h>

#define DEVICE "/dev/hda3"

int main()
{
struct dqblk block;

if (quotactl(QCMD(Q_SETQUOTA, USRQUOTA), DEVICE,
(unsigned int)~0 / sizeof(block), (caddr_t)&block))
perror("quotactl");

return 0;
}
```

It should be run as root, and is mainly for checking whether the bug got fixed -- it's not a real exploit. Be sure to run it with quota enabled, and don't forget to set DEVICE correctly. This crashes my 2.0.33 just fine.

Well, probably it's the time for fixes. If you don't need the edquota feature, you can just disable it (patch for Linux quota utils, v1.51):

```
--- edquota.c.orig Fri Mar 20 18:20:54 1998
```

```
+++ edquota.c Fri Mar 20 18:23:30 1998
@@ -173,8 +173,6 @@
struct passwd *pw;
struct group *gr;

- if (alldigits(name))
- return (atoi(name));
switch (quotatype) {
case USRQUOTA:
if (pw = getpwnam(name))
```

A real fix should probably either add an extra option (like '-n') for numeric UIDs, or at least check getpwnam() _before_ alldigits(). (The latter is still a bit dangerous though.)

Another obvious workaround for a particular site would be to disallow all-digit usernames.

And finally, here's the Linux kernel patch, for 2.0.33:

```
--- linux/fs/dquot.c.orig Sat Mar 21 06:37:47 1998
+++ linux/fs/dquot.c Sat Mar 21 06:40:02 1998
@@ -1075,6 +1075,9 @@
return(-EINVAL);
}

+ if (id & ~0xFFFF)
+ return(-EINVAL);
+
flags |= QUOTA_SYSCALL;
if (has_quota_enabled(dev, type))
return(set_dqblk(dev, id, type, flags, (struct dqblk *) addr));
```

Signed,
Solar Designer

—
Murat Arslan
PGP KeyID : 2047/673351F1
For PGPkey: finger arslanm at gate dot marketweb dot net dot tr
Key FPrint: F1C6 E3F2 91C2 CD98 440B 4073 DFBC 532F

- **Next message:** HorneT: "Re: [LINUX:1849] Apache server problemi"
- **Previous message:** Murat Arslan: "Re: [LINUX:1845] Re: quota problemi"

Linux System Administration

Quotas

---

du

- This is a command that will allow you to see how much disk space is being used in a directory and related subdirectories.
- Handy forms:

```
du -c  /directory
```

```
du -cs  /directory
```

```
du -csh  /directory
```

- The first will show you a summary for each file and subdirectory.
- The second will summarize the total for you.
- The third will put the summary in a more readable form.

Prev Page 1 Next

Linux System Administration

Quotas

- It is quite possible that a user or group could use all the space on a partition, possibly causing the system to come to a (crashing) halt.
- If we set up quotas for each user or group, we can limit the amount of resources a user or group can use.
- We can limit the amount of disk space by limiting the amount of blocks each user or group can use.
- We can limit the amount of files a user creates by limiting the amount of inodes a user or group can consume.

Prev Page 2 Next

Linux System Administration

Quotas

- Here are the steps one needs to take in order to set up quotas:

    1. Edit the /etc/fstab file
    2. Create the quota files
    3. Create quota rules
    4. Check quotas

Linux System Administration

Quotas

---

(1) Editing `/etc/fstab`

- We need to add quota support to the file system.
- Add `usrquota` to field four of the partition for which we want to set quotas.

```
    /dev/hda2          /home          ext3
defaults,usrquota          1  2

    /dev/hda2          /home          ext3
defaults,grpquota          1  2

    /dev/hda2          /home          ext3
defaults,usrquota,grpquota          1  2
```

- This will allow the `/home` file system to allow disk quotas for all users' home directories under the `/home` directory.

- The filesystem will have to be remounted in order for the change to take effect.

- The following command will allow you to unmount and remount this directory:

```
[root@localhost etc]# mount -o remount /home
```

  - If this command does not work, then you will have to reboot.

  - Please be kind and let your users know that you are going to be doing this step. They may lose data if they are logged on at the time.

  - Remember that if you type `init 1`, you will be in single user mode. `/home` is not mounted in this mode.

Linux System Administration

Quotas

---

(2) Creating quota files

- You need to have `aquota.user` and/or `aquota.group` files in the root (upper most) directory of the partition on which you want to establish disk quotas.

- `aquota.user` allows quotas for individual users.
- `aquota.group` allows quotas based on groups.

- You can create the files with the following command:

```
# quotacheck -c /home
```

- This will create the file /home/aquota.user
  - The -c option means don't read existing quota files. Just perform a new scan and save it to disk.
  - This file needs to be readable by root only, so (if necesary) do the following : `chmod 600 aquota.user`

- To create an initial aquota.group file, type : `touch /home/aquota.group`
  - Check the permissions on this file, too.

Linux System Administration

Quotas

---

(2) Creating quota files

- Next, you must create the disk usage table for the partition.

```
# quotacheck  -vug  /home
```

- The command looks at the filesystem partition mounted on /home and builds a table of disk usage.
  - The -v option produces verbose output.
  - The -u option causes user quotas to be checked.
  - The -g options causes group quotas to be checked.

Prev Page 6 Next

Linux System Administration

Quotas

---

(3) Creating quota rules

- We can use the `edquota` command to create quota rules for a particular user or group.

```
#edquota -u mlevan
```

- This will bring up a file for you to edit.
  - The default editor is vi. If you want emacs, enter the following before the edquoate command :

```
#export EDITOR=emacs
```

- The edquota command looks in `/etc/password` for valid users and `/etc/group` for valid groups.

Linux System Administration

Quotas

---

(3) Creating quota rules

```
#edquota -u mlevan

Filesystem                   blocks       soft       hard     inodes
   soft     hard
/dev/hda7                       988          0          0        123
      0        0
```

(Note : `edquota -g groupname` will create the rules for group quotas.)

- This says that mlevan has currently used 988 blocks.
- This says that 123 files have been created by mlevan. (Represented by the inodes column)
- To change the limits, we can change the zeros in the columns.
  - The first soft and hard refer to block limits.
  - The second soft and hard refer to inode limits.
- If the soft and hard limits are set to 0, then no limit is enforced.

Linux System Administration

Quotas

---

(3) Creating quota rules

- Soft limits set limits that you don't want a user or group to exceed.
  - It is possible to exceed these limits.
  - You can set a grace period for exceeding a soft limit.
    - Once that limit is exceeded, then the limit becomes a hard limit.
  - Type the following to check and change the grace periods:

```
# edquota -t
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem             Block grace period     Inode grace period
  /dev/hda7                     7days                  7days
```

- Hard limits will not allow users to exceed their quota.

Linux System Administration

Quotas

---

(3) Creating quota rules

- Here is an example of how the quotas could be changed:

```
#edquota -u mlevan

Filesystem       blocks       soft       hard       inodes
   soft      hard
/dev/hda7           988      25000      30000        123
      800      1000
```

- In this example, the soft limit on the number of blocks that the user mlevan could consume on the device /dev/hda7 (or the /home) partition is 25000 blocks (or 25MB).
- The hard limit is 30000 blocks (or 30MB).
- Soft and hard limits on inodes are 800 and 1000, respectively.
- If mlevan passes these soft limits, then he has 7 days to get back under the limit, or he will be blocked from using any more disk space or inodes.
- After you have changed the settings for a user, run `repquota` to see if the settings are to your liking.

```
# repquota -a
# repquota /home
```

Linux System Administration

Quotas

---

(3) Creating quota rules

- If further attempts to write to a partition after the hard limit has been reached results in a failure to write to the disk.
- When this occurs, you may see a message such as:

```
Disk quota exceeded
```

- In Fedora Core 3, quotas are enabled at boot time. You should see the following near the beginning of a reboot:

```
                    Enabling local filesystem
quotas                          [   OK   ]
```

- If you have changed your quotas, you can use the quotaoff and quotaon commands to stop and start the quota service.

```
# quotaoff  /home

# quotaon  /home
```

Linux System Administration

Quotas

(4) Checking quotas

- Here is a check you can try:
  - Set the inodes soft limit to 50 more than the current usage and the hard limit to 60 more than the current usage.
  - Run the following script:

```
for i in $(seq 1 65); do
echo -n "Touching file${i}"; touch
file${i} 2>&1; done | less
```

- Notice you get two types of error messages. One when you pass the soft limit and another when you hit the hard limit.
- Use the `repquota` command to see that you have filled up your inode allocation.

Linux System Administration

Shells

- The shell is one of the most important parts of a Unix system.
- A shell is a program that runs commands.
- There are many different shells, but all derive many of their features from the *Bourne shell*, or /bin/sh.
- Every Unix system needs the Bourne shell to function correctly.
- `bash` is the default shell for most Linux distributions.
- `/bin/sh` is usually a link to `bash` on a Linux system.

Prev Page 1 Next

Linux System Administration

Shells

Startup Files

- When a user is created, many startup files are placed in their home directory.

```
[root@localhost ~]# useradd guest
[root@localhost ~]# cd /home/guest
[root@localhost guest]# ls -al
total 36
drwx------  3 guest guest 4096 May 11 20:29 .
drwxr-xr-x  5 root  root  4096 May 11 20:29 ..
-rw-r--r--  1 guest guest  302 May 11 20:29
.bash_logout
-rw-r--r--  1 guest guest  191 May 11 20:29
.bash_profile
-rw-r--r--  1 guest guest  124 May 11 20:29
.bashrc
-rw-r--r--  1 guest guest  383 May 11 20:29
.emacs
-rw-r--r--  1 guest guest  120 May 11 20:29
.gtkrc
drwxr-xr-x  3 guest guest 4096 May 11 20:29 .kde
```

- Remember these files come from /etc/skel/

Linux System Administration

Shells

Shell Startup File Elements

- What goes into a shell startup file?
    - The Path
        - What should be in the path?
    - The Prompt
        - What does a reasonable prompt look like?
    - Aliases?
        - What makes a good alias?

- These are the types of questions you need to think about when thinking about putting anything in your startup files.

Linux System Administration

Shells

The Command Path

- The most important part of any shell startup file is the command path.
  - The command path tells the shell which directories to look into when you try to run an application.
  - The path should cover all the directories that contain the applications of interest to the user.
  - To see the current path, type echo $PATH

```
[guest@localhost ~]$ echo $PATH

/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/guest/bin
```

  - Suppose I like to run the application `ifconfig`. This is a file in `/sbin`. Here are its permissions.

```
[guest@localhost ~]$ ls -l /sbin/ifconfig
-rwxr-xr-x  1 root root 56492 Jan 31 05:24 /sbin/ifconfig
```

Linux System Administration

Shells

The Command Path

- Notice that I have permission to run this application. In order to run it, I would have to use the full pathname:

```
[guest@localhost ~]$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr
00:50:BF:50:B3:DA
          inet addr:192.168.1.150
Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr:
fe80::250:bfff:fe50:b3da/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST
MTU:1500  Metric:1
          RX packets:40824 errors:0 dropped:0
overruns:0 frame:0
          TX packets:31382 errors:0 dropped:0
overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:30254304 (28.8 MiB)  TX
bytes:5336892 (5.0 MiB)
          Interrupt:11 Base address:0xe000
```

- If /sbin were in my path, then all I would have to type is the following:

```
[guest@localhost ~]$ ifconfig
```

Linux System Administration

Shells

The Command Path

- You do want to be careful and (probably) not add the . (dot) directory to your path.
  - If you want to run a program in your current directory that is not in your path, you would have to type the following:

```
[guest@localhost ~]$ ./application
```

  - If the . (dot) directory is in your path, then all you have to do is type:

```
[guest@localhost ~]$ application
```

  - This may seem convenient, but it could cause a couple of problems:
    1. It can lead to security problems.
    2. A command's behavior can change according to the current directory.

Linux System Administration

Shells

The Command Path

- How can we change the PATH ?
  - In a shell, type the following:

```
[guest@localhost ~]$ echo $PATH
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/guest/bin
[guest@localhost ~]$ PATH=/sbin:$PATH
[guest@localhost ~]$ echo $PATH
/sbin:/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/guest/bin
```

  - This changes the PATH for this shell. If you open a new shell, then PATH will be the default.
  - Note that the shell looks for a command from left to right along the PATH.
    - The command above will look in /sbin first. If you want to look in /sbin last, do the following:

```
[guest@localhost ~]$ echo $PATH
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/guest/bin
[guest@localhost ~]$ PATH=$PATH:/sbin
[guest@localhost ~]$ echo $PATH
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/guest/bin:/sbin
```

Linux System Administration

Shells

The Command Path

- How can we change the PATH for every shell we open?
- Change your .bash_profile
  - Edit the following line near the end of the file:

```
PATH=$PATH:$HOME/bin
```

  - Change it to the following:

```
PATH=$PATH:$HOME/bin:/sbin
```

- This will change the PATH for every shell you now open.

Linux System Administration

Shells

The Prompt

- Avoid long, complicated prompts.
  - Just because you can place a lot of information in a prompt, doesn't mean you should.

```
                \a     an ASCII bell character (07)
                 \d     the  date in "Weekday Month Date"
format (e.g., "Tue May
                        26")
                \D{format}
                       the format is passed to strftime(3)
and  the  result  is
                       inserted into the prompt string; an
empty format results
                       in a locale-specific time
representation.   The  braces
                       are required
                \e     an ASCII escape character (033)
                \h     the hostname up to the first ‘.’
                \H     the hostname
                \j     the number of jobs currently
managed by the shell
                \l     the basename of the shell’s
terminal device name
                \n     newline
                \r     carriage return
                \s     the  name  of the shell, the
basename of $0 (the portion
                       following the final slash)
                \t     the current time in 24-hour
HH:MM:SS format
                \T     the current time in 12-hour
HH:MM:SS format
                \@     the current time in 12-hour am/pm
format
                \A     the current time in 24-hour HH:MM
format
                \u     the username of the current user
                \v     the version of bash (e.g., 2.00)
```

```
                \V     the release  of  bash,  version  +
patch  level  (e.g.,
                       2.00.0)
                \w     the  current  working  directory,
with $HOME abbreviated
                       with a tilde
                \W     the basename of  the  current
working  directory,  with
                       $HOME abbreviated with a tilde
                \!     the history number of this command
                \#     the command number of this command
                \$     if the effective UID is 0, a #,
otherwise a $
                \nnn   the character corresponding to the
octal number nnn
                \\     a backslash
                \[     begin a sequence of non-printing
characters, which could
                       be used to embed a terminal
control  sequence  into  the
                       prompt
                \]     end a sequence of non-printing
characters
```

- Some examples:

```
<>                  $ export PS1="> "
                   $ export PS1="This is my super prompt >
"
                   $ export PS1="\u@\H > "
```

Linux System Administration

Shells

The Prompt

- How can we make these changes permanent?
  - The prompt is originally set in /etc/bashrc
  - We can export the new prompt in ~/.bash_profile

`export PS1="  whatever  "`

where *whatever* is your choice of prompt.

current default : `PS1="[\u@\h \W]\\$ "`

for fun: `export PS1="\[\e[36;1m\]\u@\[\e[32;1m\]\H>
\[\e[0m\]"`

Linux System Administration

Shells

Default File Order

- OK, so we have several default files. In what order are they executed?

`/etc/profile`

- This file sets up user environment information for every user.
- This is executed when you first log in.
- This file provides values for your path, history files, and more.
- `/etc/profile` gathers information from configuration files in `/etc/profile.d` directory.

Linux System Administration

Shells

Default File Order

`/etc/bashrc`

- This file is executed for every user who runs the bash shell.
- This file is executed every time a bash shell is opened.
- It sets the default prompt and may add some aliases.
- Values can be overwritten in each user's `~/.bashrc` file.

Linux System Administration

Shells

Default File Order

`~/.bash_profile`

- This file is used by each user to enter information that is specific to his/her own use of the shell.
- It is executed only once, when the user logs in.
- By default, it sets a few environment variables and executes the user's `~/.bashrc` file.

Linux System Administration

Shells

Default File Order

`~/.bashrc`

- This file contains information that is specific to your bash shells.
- It is read when you log in and each time you open a bash shell.
- This is the best location to add environment variables and aliases so that your shell picks them up.

Linux System Administration

Shells

Default File Order

`~/.bash_logout`

- This file executes every time you log out (exit the bash shell).
- By default, it simply clears the screen.

Linux System Administration

Shells

Default Files

- In order to change the main default files , `/etc/profile` and `/etc/bashrc`, you must be root.
- Users control the information in the `~/.bash_profile`, `~/.bashrc`, and `~/.bash_logout` files.
- If a system administrator sees a good change in a user's account, they might want to consider changing the defaults for all users.

Linux System Administration

Help!

- Where can we go to look for help?
  - Google.com/linux
  - Linux Forums (free registration required)
  - LinuxSelfHelp.com
  - LinuxQuestions.org
  - The Linux Documentation Project
  - man pages, apropos, whatis
  - Man Pages Online
  - Help Facilities (From Linux Cookbook)

Linux System Administration

Software Management with RPM

Red Hat Package Manager (RPM)

- An RPM is a non-interactive method of installing software.
- The `rpm` utility works only for those software packages that have been built for processing by `rpm`.
- Red Hat released `rpm` under the GPL, and hence, `rpm` is used by several different distributions.
- The rpm utility keeps track of where software packages are installed, the versions of the software that you have installed, and the dependencies between the packages.

Prev Page 1 Next

Linux System Administration

Software Management with RPM

- The RPM systems consists of a local database, the `rpm` executable, and `rpm` package files.
- The local RPM database is kept in `/var/lib/rpm`.
- RPM packages have the following format:

```
name-version-release.architecture.rpm
```

  - The name is the name of the package.....
  - The version refers to the open source version of the project.
  - The release refers to Red Hat internal patches to the open source code.
  - The architecture refers to the the architectures supported by Red Hat:
    - `i386, i486, i586, i686` : Intel x86 Compatible
    - `x86_64` : AMD 64-bit
    - `ppc64` : Power PC 64-bit
    - `s390x` : IBM Mainframe 64-bit
    - noarch : architecture independent code (scripts, documentation, images, etc.)
    - src : Source code and files required to build the binary RPM.

Linux System Administration

Software Management with RPM

- The RPM command can be used to install any packages that are in RPM format.
- Usually, an RPM installs a binary file.
- The rpm command comes with several options:
  - install : -i
  - upgrade : -U
  - freshen : -F
  - query : -q
  - verify : -V
  - signature check : --checksig
  - uninstall / erase : -e
  - rebuild database : --rebuilddb
  - fix permissions : --setperms
  - set owners/groups : --setugids
  - show RC : --showrc
- Note that you must be logged in as root in order to add or remove packages.
- We shall review some of these, and leave the rest as an exercise for the reader.

Linux System Administration

Software Management with RPM

RPM - Install

- Here is the command one would use to install an RPM:

```
# rpm -i package-name.rpm
```

- When installing an RPM, the rpm command will consult the local database to ensure that any dependencies are installed on the system, and that installing the RPM will not write over any needed pre-existing files.
- These checks can be omitted by adding `--nodeps` or `--replacefiles` to the command when installing.

```
# rpm -i --nodeps package-name.rpm
```

```
# rpm -i --replacefiles package-name.rpm
```

- If you want to do both of these commands, you can use the --force options.

```
# rpm -i --force package-name.rpm
```

Linux System Administration

Software Management with RPM

RPM - Install

- You can get some feedback with the verbose (-v) option and the hash (-h) option.
  - The verbose option will give you a little information as the package is being installed.
  - The hash option will print 50 hash marks as the files is installed.

```
# rpm -ivh package-name.rpm
```

- Note that if you are trying to install a newer version of an application with this command, you will *not* remove the old version.

Linux System Administration

Software Management with RPM

RPM - Upgrade

- If you have previously installed a package via RPM earlier, and you have found a later version of the software, then you can use the Upgrade (-U) option to install the newer version.

```
# rpm -Uvh package-name.rpm
```

- If you use the -U option as opposed to the -i option, the original package on the system will be removed, and the new package installed.
- If the package is not installed on the system and you use the -U option, the package will be installed anyway.
    - Many people generally use the -U option instead of the -i option.
- Note that a kernel should *not* be installed using this option! Make sure that if you are installing a new kernel via RPM that you do it with the Install option! If it goes well, you can remove the old kernel later.

Linux System Administration

Software Management with RPM

RPM - Freshen

- The freshen option is almost the Upgrade option. The only difference is that an older version of the package is not on the system, then the package will not be installed.

```
# rpm -Fvh package-name.rpm
```

- This can be a nice option if you can find a repository that contains all the RPM's that are installed on your system. You could freshen (upgrade) all the packages on your system with the following command:

```
# rpm -Fvh ftp://servername.com/current/en/os/i386/*rpm
```

Linux System Administration

Software Management with RPM

RPM - Uninstall / Erase

- The command to unistall/erase (-e) a package is as follows:

```
# rpm -e packagename
```

- For example, assume we installed the program Tux Paint with the following command:

```
# rpm -ivh tuxpaint-0.9.14-1.fc3.i386.rpm
```

- We would uninstall with the following command:

```
# rpm -e tuxpaint
```

- Notice that we only used the name of the package, and not the entire name of the RPM file.
- If we do use the entire name of the RPM, then we will be told that the file is not installed.
- If there are no dependencies on this package, then it is removed.

Linux System Administration

Software Management with RPM

RPM - Uninstall / Erase

- If you want to see the files that are being removed, you can use the -vv option.

```
# rpm -evv tuxpaint
```

- This can sometimes lead to a bunch of filenames flying down the screen. So, you might want to pipe the output to less or to a file for you to review later.
- You can also override some problems that might arise with a simple unistall.

- You may run into a dependcy problem if you try to remove a package that others are relying on.

```
# rpm -evv --nodeps tuxpaint
```

- The above option will uninstall the package without checking for dependencies.

```
# rpm -evv --noscipts tuxpaint
```

- The above command will unistall the package without running and preunistall or postunistall scripts.

```
# rpm -evv --notriggers tuxpaint
```

- The above command will uninstall the package without executing scripts that are triggered by removing the package.

Linux System Administration

Software Management with RPM

RPM - Query

- You can use the query (-q) option to get information about the package.
- Here are some options you can use with query:
  - -qa : lists all installed packages.
  - -qf file : Lists the packages that owns file.
  - -qi package : Lists lots of information about the package.
  - -qR package : Lists components (such as libraries and commands) that package depends on.
  - -ql package : Lists all the files contained in the package.
  - -qd package : Lists all documentation files that come in the package.
  - -qc package : Lists all configuration files that come in package.

Linux System Administration

Software Management with RPM

RPM - Query

- Here are some examples: (You might want to pipe these to a `more` command)

```
# rpm -qa

# rpm -qi tuxpaint

# rpm -ql tuxpaint

# rpm -qR tuxpaint
```

- Note that I am only using the package name here, and not the entire name of the RPM file.

Linux System Administration

Software Management with RPM

RPM - Verifying

- If you have a package that stops working, or if you suspect that your system might have been tampered with, then the verify option will compare the installed software versus its original software package.

- This has the -V option, as opposed to the -v (verbose) option.
- If everything is fine, then there is no output. If there is a conflict, then you may see some indicators:
  - **5 - MD5 Sum -** An MD5 ckecksum indicates a change to the file contents.
  - **S - File size -** The number of characters in the file has changed.
  - **L - Symlink -** The file has become a symbolic link to another file.
  - **T - Mtime -** The modification time of the file has changed.
  - **D - Device -** The file has become a device special file.
  - **U - User -** The username that owns the file has changed.
  - **G - Group -** The group assigned to the file has changed.
  - **M - Mode -** If the ownership or permission of the file has changed.
- These are the 8 fields that are checked.

Linux System Administration

Software Management with RPM

RPM - Verifying

- Example : Imagine I have changed the user and group of tuxracer from root to mlevan:

```
[root@localhost bin]# ls -l tux*
-rwxr-xr-x  1 root root 118668 Nov  9  2004
tuxpaint
-rwxr-xr-x  1 root root   1929 Nov  9  2004
tuxpaint-import
-rwxr-xr-x  1 root root 294948 Sep 21  2004
tuxracer

[root@localhost bin]# chown mlevan tuxracer
[root@localhost bin]# chgrp mlevan tuxracer
```

- When we now verify the package, we can see that these have changed from the install:

```
[root@localhost bin]# rpm -V tuxracer
.....UG.    /usr/bin/tuxracer
```

- Notice that fields 7 and 8 are marked as changed.

Linux System Administration

Software Management with RPM

RPM - Where To Find Them

- Where can you find RPM files to download?
  - rpmfind.net
  - rpm.pbone.net
  - freshrpms.net
  - rpmseek.com
  - fedoratracker.org

- Information about RPM:
  - http://www.rpm.org

Linux System Administration

Software Management with RPM

APT

- When you are trying to install a file via RPM, you might have several dependent packages that you will have to download and install first.
- APT (Advanced Package Tool) can help with the issues of dependencies.
- APT will try to download and install any dependent packages needed.
  - APT will look into all the RPM repositories that you have listed for the packages and dependencies.
  - If your repository does not contain your package or all the needed dependencies, then you can either add more repositories, or you can download the missing RPM files and install those yourself.
- APT will download a package, any needed dependencies, and will use RPM to install the packages.
  - This means APT will keep up the RPM database.
- APT originally started as part of the Debian distribution, but is so useful, it has since been ported to many other RPM based distributions.

Linux System Administration

Software Management with RPM

APT Commands

- You will first need to download the latest version of APT to install.

  `apt-0.5.15cnc7-1.i386.rpm` (I think this is the latest)

- The first command you need to use is update. This will update the local package list.

```
[root@localhost bin]# apt-get update
              Get:1 http://apt.sw.be fedora/3/en/i386
release [504B]
              Get:2 http://ayo.freshrpms.net
fedora/linux/3/i386 release [2135B]
              Get:3 http://newrpms.sunsite.dk
redhat/en/i386/fc3 release [496B]
              Reading Package Lists... Done
              Building Dependency Tree... Done
```

- Because the available packages changes regularly, it is a good idea to run this command often.

Linux System Administration

Software Management with RPM

APT - Commands

- APT will not run if there is a broken RPM dependency tree.
- To check the status of the tree, use the **check** option with apt:

```
[root@localhost bin]# apt-get check
Reading Package Lists... Done
Building Dependency Tree... Done
```

- If there is an error, you can uninstall the package that is breaking the dependencies, and re-install it using APT.

Linux System Administration

Software Management with RPM

APT - Commands

- The following command will update all the RPM-based applications on the system that depend on the software that is already installed:

```
[root@localhost bin]# apt-get upgrade
```

- If there are packages that are dependent on packages that are not installed, then the following will also install dependencies:

```
[root@localhost bin]# apt-get dist-upgrade
```

Linux System Administration

Software Management with RPM

APT - Installing Packages

- The command to install a package via APT is as follows:

```
[root@localhost bin]# apt-get install package
```

For example:

```
[root@localhost bin]# apt-get install tuxpaint
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  tuxpaint
0 upgraded, 1 newly installed, 0 removed and 166 not upgraded.
Need to get 0B/2922kB of archives.
After unpacking 5752kB of additional disk space will be used.
Committing changes...
Preparing...
########################################### [100%]
   1:tuxpaint
########################################### [100%]
Done.
```

Linux System Administration

Software Management with RPM

APT - Removing Packages

- The command to remove a package via APT is as follows:

```
[root@localhost bin]# apt-get remove package
```

For example:

```
[root@localhost bin]# apt-get remove
tuxpaint
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  tuxpaint
0 upgraded, 0 newly installed, 1 removed and
166 not upgraded.
Need to get 0B of archives.
After unpacking 5752kB disk space will be
freed.
Do you want to continue? [Y/n] Y
Committing changes...
Preparing...
########################################## [100%]
Done.
```

Linux System Administration

Software Management with RPM

APT - Removing Packages

- Note that `apt-get` remove will not remove configuration files from `/etc`.
  - This allows you to reinstall the package at a later date and still have the same configuration.
- You can use the --purge option to remove all files, including configuration files.
- Another option would be to move the files to another location and archive them.
  - How can you know which files to move: `rpm -ql package`

Linux System Administration

Software Management with RPM

APT - Configuration

- Which file configures the options for APT?
  - `/etc/apt/apt.conf`

- This file comes in three sections :
  - Apt : This contains the settings for the APT tools
  - Acquire : This contains settings related to the package-fetching mechanism
  - RPM : This contains RPM specific settings.

- Review this file to see the setup. Most parts are fairly self-explanatory.

Prev Page 22 Next

Linux System Administration

Software Management with RPM

APT - Repositories

- APT repositories are locations that store RPM files for you to download.
  - Different repositories may tend towards different packages. There may be an "official" repository for your distribution, or there may be a repository with more multimedia applications.

- These are kept in the file : `/etc/apt/sources.list.rpmsave`

- To add a new repository, add following on one line to the end of the file:
  - **Type** : rpm, rpm-src, rpm-dir, rpm-src-dir
  - **URI** : location of the RPM's
  - **Distribution** : The distribution to use
  - **Sections** : core updates, os updates, many others. You can have more than one listed here.

- For example, you could add the following to the end of `/etc/apt/sources.list.rpmsave`

```
rpm http://rpm.livna.org/ fedora/3/i386 stable unstable testing
```

or

```
rpm http://apt.sw.be/ fedora/3/en/i386 dag
```

- Many repositories may be found here:

[Fedora Core 3 Repositories]

Linux System Administration

Software Management with RPM

APT - GUI

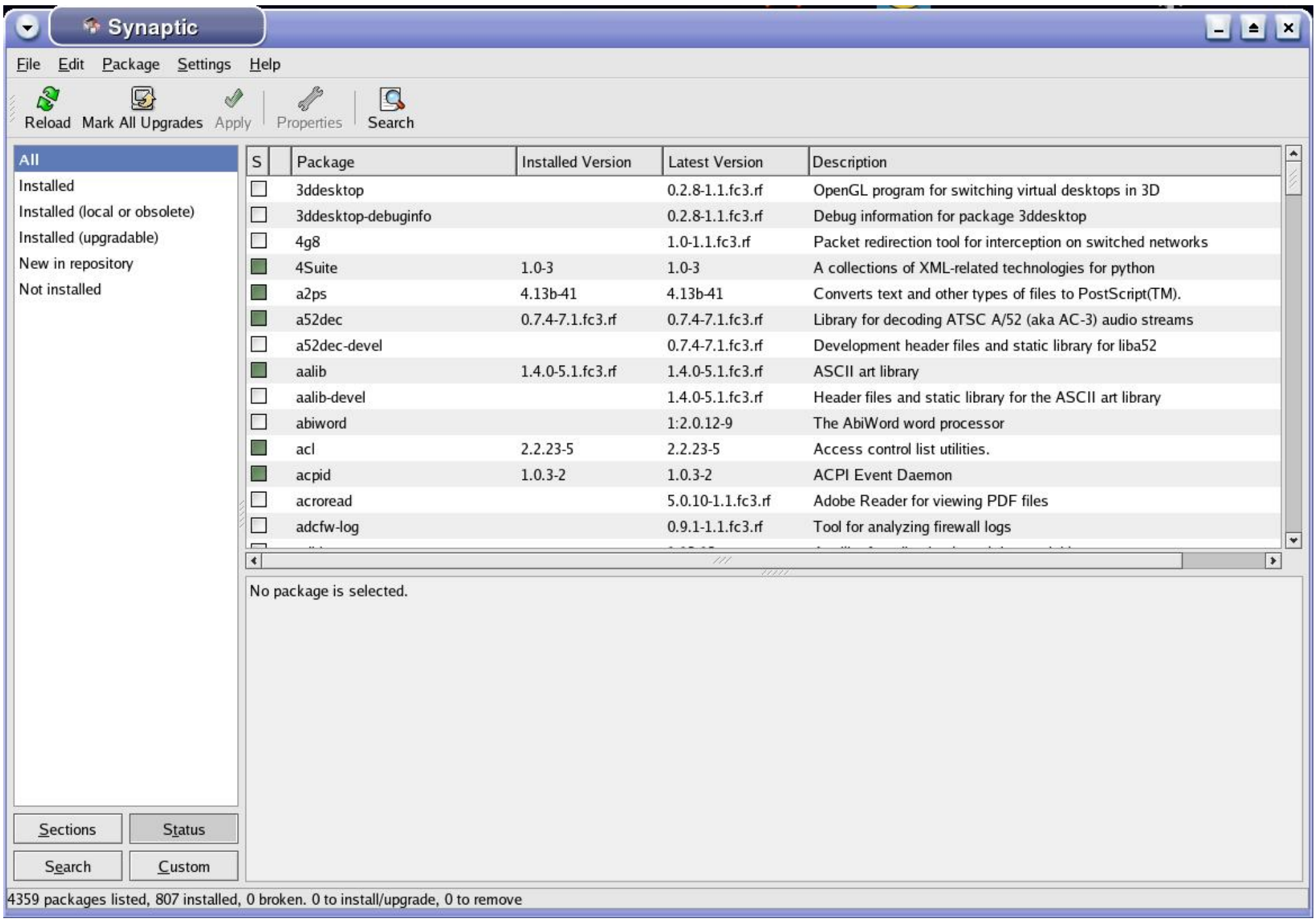
- Synaptic is the GUI for APT.
- Each time you start up synaptic, it creates and checks the dependency tree.
- Remember to click on the update (not upgrade!) button when you start synaptic.
- Synaptic lists the packages hierarchically, providing a nice way of browsing the packages available in the repositories.
  - You can select packages to upgrade, install, and remove.

Linux System Administration

Software Management with RPM

APT - Synaptic

- Synaptic should look a little like this:

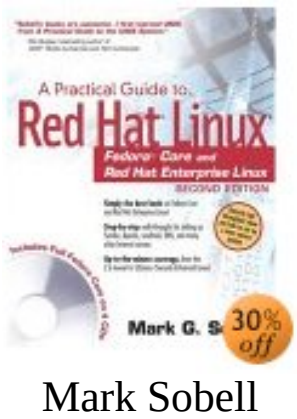


Linux System Administration

Software Management with RPM

Sources

- Some outside sources come from the following:



Mark Sobell

Christopher Negus

Linux System Administration

SSH - Secure Shell

SSH

- The secure shell (SSH) is now the de facto standard for remots logins to other machines.
- It has replaced old, insecure programs such as `telnet` and `rlogin`.
- SSH offers the following:
  - All data is encrypted, including the password.
  - You can tunnel other network connections.
    - Tunnelling is the process of packaging and transporting a network connection using another network connection.
    - I.e., we can tunnel X Window System connections and the data will be encrypted.
  - SSH has clients for almost every operating system.
  - SSH uses keys for host authentication.

Linux System Administration

SSH - Secure Shell

SSH

- Linux uses the package OpenSSH
  - The client is `ssh`.
  - The server is `sshd`.

- You can use the network services to turn sshd on and off:

```
[root@Hamming ~]# service sshd stop
Stopping
sshd:                                                      [  OK
]

           [root@Hamming ~]# chkconfig sshd off
           [root@Hamming ~]# chkconfig --list | grep
ssh
           sshd                0:off   1:off   2:off
3:off   4:off   5:off   6:off
```

- OpenSSH supports two protocols : 1 and 2.
  - Version 2 is usually the default.

Linux System Administration

SSH - Secure Shell

SSHD Server

- To run `sshd`, you need a configuration file and host keys in the configuration directory.
- This file is located in `/etc/ssh/sshd_config`
- You should not need to change anything in this file, but take a look at it to see what is going on.
- Here are a few entries in the file:
  - **HostKey** *file* : Uses *file* as a host key
  - **SyslogFacility** *name* : Logs messages with the syslog facility *name*
  - **LogLevel** *level* : Logs messages with the syslog level *level*
  - **PermitRootLogin** *value* : Permits the superuser to log in with SSH if *value* is set to yes; set *value* to no if you do not want to allow this.
  - **X11Forwarding** *value* : Enables X Wind System client tunneling if *value* is set to yes
  - **XAuthLication** *path* : Provides *a path* for xauth; X11 tunneling does not work without xauth. If xauth isn't in /usr/X11R6/bin, set *path* to the full pathname for xauth.
- Do not confuse this file with `ssh_config`!

Linux System Administration

SSH - Secure Shell

SSH - Host Keys

- OpenSSH has three different host key sets: one for protocol version 1, and two for protocol 2.
  - Each set has a *public key*. This has a .pub file extension.
  - Each set has a *private key*. This has no extension.

- If someone gets your host's private key, SSH provides no protection against password snooping.
- SSH version 1 has RSA keys.
- SSH version 2 has RSA and DSA keys.
  - RSA and DSA are public ke cryptography algorithms.
  - SSH version 2 supplies both sets, as there is a debate as to which version is better.

Linux System Administration

SSH - Secure Shell

SSH - Host Keys

- The key filenames are as follows:

ssh_host_rsa_key	Private RSA key (version 2)
ssh_host_rsa_key.pub	Public RSA key (version 2)
ssh_host_dsa_key	Private DSA key (version 2)
ssh_host_dsa_key.pub	Public DSA key (version 2)
ssh_host_key	Private RSA key (version 1)
ssh_host_key.pub	Public RSA key (version 1)

- You do not normally need to build the keys.
  - OpenSSH does this for you.

- However, if you want to use ssh-agent, you need to know how to create keys.

Linux System Administration

SSH - Secure Shell

SSH - Host Keys

- To create SSH version 2 keys, use the `ssh-keygen` program that comes with OpenSSH:

```
[mlevan@Hamming SSH]$ ssh-keygen -t rsa -N '' -f
ssh_host_rsa_key
Generating public/private rsa key pair.
Your identification has been saved in ssh_host_rsa_key.
Your public key has been saved in ssh_host_rsa_key.pub.
The key fingerprint is:
65:8a:a9:41:4c:ca:64:6a:a3:39:49:8e:cc:94:8d:f3
mlevan@Hamming.math.transy.edu

[mlevan@Hamming SSH]$ ssh-keygen -t dsa -N '' -f
ssh_host_dsa_key
Generating public/private dsa key pair.
Your identification has been saved in ssh_host_dsa_key.
Your public key has been saved in ssh_host_dsa_key.pub.
The key fingerprint is:
ef:7b:60:d7:f1:95:8c:9e:43:82:76:29:62:ec:18:65
mlevan@Hamming.math.transy.edu
```

- To create SSH version 1 keys, use this command:

```
[mlevan@Hamming SSH]$ ssh-keygen -t rsa1 -N '' -f
ssh_host_key
Generating public/private rsa1 key pair.
Your identification has been saved in ssh_host_key.
Your public key has been saved in ssh_host_key.pub.
The key fingerprint is:
e6:48:99:9b:7c:d8:8c:17:d0:9b:cb:a5:81:d9:48:29
mlevan@Hamming.math.transy.edu
```

Linux System Administration

SSH - Secure Shell

SSH - Host Keys

- The SSH server (and clients) also use another key file : `ssh_known_hosts`
  - This file contains public keys from other hosts.
  - This file must contain the host keys of all trusted clients.

  - When you use SSH the first time, you are asked if you want to accept the host public_key.

```
[root@Hamming etc]# ssh mlevan@12.222.236.168
The authenticity of host '12.222.236.168 (12.222.236.168)'
can't be established.
RSA key fingerprint is
f8:48:3f:ac:46:90:f3:38:31:65:f4:4a:eb:81:00:c9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '12.222.236.168' (RSA) to the
list of known hosts.
```

  - If that key changes, you will receive a warning :

```
dhcp-129-64-76-193:~ zshaw$ ssh harpo.unet.brandeis.edu
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-
middle attack)!
It is also possible that the RSA host key has just been
changed.
The fingerprint for the RSA key sent by the remote host is
f2:92:1d:da:81:2a:d7:16:0a:48:f0:43:20:1c:f4:b5.
Please contact your system administrator.
Add correct host key in /Users/zshaw/.ssh/known_hosts to
get rid of this message.
Offending key in /Users/zshaw/.ssh/known_hosts:5
Password authentication is disabled to avoid man-in-the-
middle attacks.
X11 forwarding is disabled to avoid man-in-the-middle
attacks.
Permission denied (publickey,password,keyboard-
interactive).
```

- If you get this message, contact the system administrator of the system you are trying to connect to and determine if the key has been changed. If it has, then it is safe to remove this key and get the new key. If not, then someone might be trying to deceive you.

Linux System Administration

SSH - Secure Shell

SSH Login Client

- To log in to a remote host, run this command:

```
ssh username@host
```

- If your local username is the same as the username on the host, then you could omit the username argument and run this command:

```
ssh host
```

- Note that you will be asked for a password before you are dropped into a shell.

Prev Page 8 Next

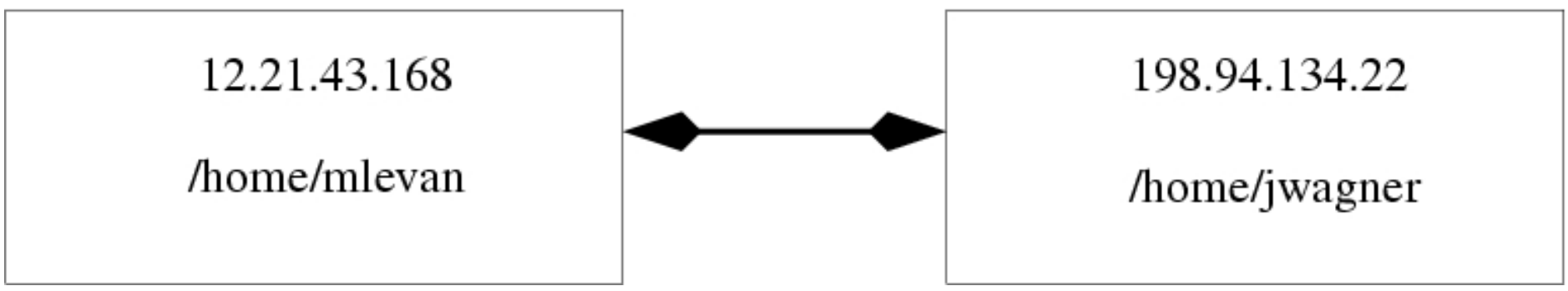
Linux System Administration

SSH - Secure Shell

SSH File Transfer Clients

- You can use scp to transfer files from a remote machine to your machine.
- The syntax is similar to the cp command.

```
scp file-to-be-copied-location location-to-copy-file
```



- Examples:
  - From local to 198.94.134.22:

```
scp test1.txt  jwagner@198.94.134.22:/home/jwagner/
```

This command will take the local file test1.txt and place it on the remote machine in the directory /home/jwagner

  - From 198.94.134.22 to local:

```
scp jwagner@198.94.134.22:/home/jwagner/report.txt .
```

This command will take the file from the remote machine and place it in the current directory .

  - You can even copy files from different machines:

```
scp user1@host1:file  user2@host2:dir
```

Linux System Administration

SSH - Secure Shell

SSH - SFTP

- This command can take the place of the `ftp` protocol. It provides a secure `ftp` session.
- This works much like ftp with get and put commands.
- The remote host must have a sftp-server program.
  - OpenSSH provides one of these.

```
[mlevan@Hamming SSH]$ sftp mlevan@12.222.238.166
Connecting to 12.222.238.166...
mlevan@12.222.238.166's password:
sftp>
```

Linux System Administration

SSH - Secure Shell

SSH-Agent

- It can be a pain to have to enter your password every time you use one of these commands.
- We can configure OpenSSH so we do not have to enter a password each time we try to connect to a remote system.
- Here is what we need to do:

    1. Generate a personal authentication key.
    2. Place the public part of the key on the remote server.
    3. Keep the private part of the key on the local client.

- When you try to connect to the remote system, the system issues a challenge based on the public key.
- The private part of the key is required to respond properly to the challenge.
- When the local system provides the proper response, the remote system logs you in.

Linux System Administration

SSH - Secure Shell

SSH-Agent

- We need to first generate the personal authentication keys.
- You can see if the keys exists by looking in **`~./ssh`** for either **id_dsa** and **id_dsa.pub** or **id_rsa** and **id_rsa.pub**.
- If these do not exist, you can do the following to create either the **dsa** or **rsa** keys:

```
[mlevan@localhost ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key
(/home/mlevan/.ssh/id_rsa):
/home/mlevan/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/mlevan/.ssh/id_rsa.
Your public key has been saved in
/home/mlevan/.ssh/id_rsa.pub.
The key fingerprint is:

34:63:9d:fc:04:26:ee:65:b3:8d:2d:3a:c0:8e:20:41
mlevan@localhost.localdomain
```

- This command creates two keys : **id_rsa** and **id_rsa.pub**.
- If you want to generate DSA keys, replace rsa with dsa.
- Be careful with the passphrase, as there is no way to recover it.

Linux System Administration

SSH - Secure Shell

SSH-Agent

- If you want to log on to a remote system, make sure you have a **.ssh** directory in the home directory.

- Next, copy the **id_rsa.pub** file on the local system to a file named **~/.ssh/authorized_keys** on the remote system.
  - If this file already exists, then append this file to **authorized_keys**.

- Make sure no one except the owner has permission to read or write to this file.

- We can now log in to the remote system using the passphrase we provided earlier.
  - This isn't really any better, so what can we do?

Linux System Administration

SSH - Secure Shell

SSH-Agent

- We can use *ssh-agent* to handle authentication requests.
  - You enter the passphrase once at the beginning of the session.
  - This will last until we log out from the session.
  - If we log out from the session and then try to log back in, we will have to re-enter the passphrase.

- In order to start *ssh-agent*, we need to pass along a shell as a parameter:

```
[mlevan@Hamming ~]$ ssh-agent bash
[mlevan@Hamming ~]$
```

- This will return you to a normal looking prompt.

Linux System Administration

SSH - Secure Shell

SSH-Add

- Next we need to run *ssh-add*.
  - This will load all of the keys in the `~/.ssh` directory.

```
[mlevan@Hamming ~]$ ssh-add
Enter passphrase for
/home/mlevan/.ssh/id_rsa:
Identity added: /home/mlevan/.ssh/id_rsa
(/home/mlevan/.ssh/id_rsa)
[mlevan@Hamming ~]$
```

- We can now log into any of the SSH hosts without entering a passphrase.

```
[mlevan@Hamming ~]$ ssh
mlevan@12.222.238.XXX
Last login: Mon May 16 22:59:35 2005 from
198.94.186.XXX
[mlevan@HomePC ~]$
```

- To shut down ssh-agent, simply exit out of the ssh-agent shell.

Linux System Administration

SSH - Secure Shell

SSH

- If you have *ssh-agent* running in one shell, when you open up another shell, you do not have *ssh-agent* running in the new shell. You will have to run the command for each shell you are planning to use with SSH if you want to use *ssh-agent*.

- You can see what keys are being used by ssh-agent with the following command:

```
[mlevan@Hamming ~]$ ssh-add -l
1024
c3:53:83:99:80:d8:ef:97:2d:bb:1d:14:39:b2:a9:8f
/home/mlevan/.ssh/id_rsa (RSA)
[mlevan@Hamming ~]$
```

- You need to be in the ssh-agent shell in order to use this command. This will not work if you are logged into your remote system.

Linux System Administration

SSH - Secure Shell

Tunneling X over SSH

- Put this line in /etc/ssh/sshd_config on the SSH server:

```
X11Forwarding yes
```

- Then connect to the server with the -*X* flag:

```
$ ssh -X 12.222.238.129
```

- Run this command to test that X forwarding is working:

```
# echo $DISPLAY
localhost:10.0
```

- If it weren't, it would return a blank line.
- Now you can run any X program installed on the server as though it were local.

```
$ xeyes
```

- For security purposes, make sure that ~./Xauthority has permissions 600.

Linux System Administration

SSH - Secure Shell

Tunneling X over SSH

- Be sure that these entries are in your local */etc/ssh/ssh_config* file, and and *~/.ssh/ssh_config* files on your system:

```
Host *
ForwardX11 no
ForwardAgent no
```

- This will ensure that X forwarding is turned off, except when you really need it.
- Note that X sessions over SSH can sometimes have some lag.

Linux System Administration

SSH - Secure Shell

SSH - Windows

- What if we wanted to access our Linux box from a Windows machine?
  - PuTTY us a free SSH client for Windows.
  - There is no server component, this is just a client.
  - All you do is download, install, and double click. The program comes up, and you can enter the hostname/IP and click Open.
  - WinSCP is a graphical `scp` protocol for Windows.
  - You can download from the following location:

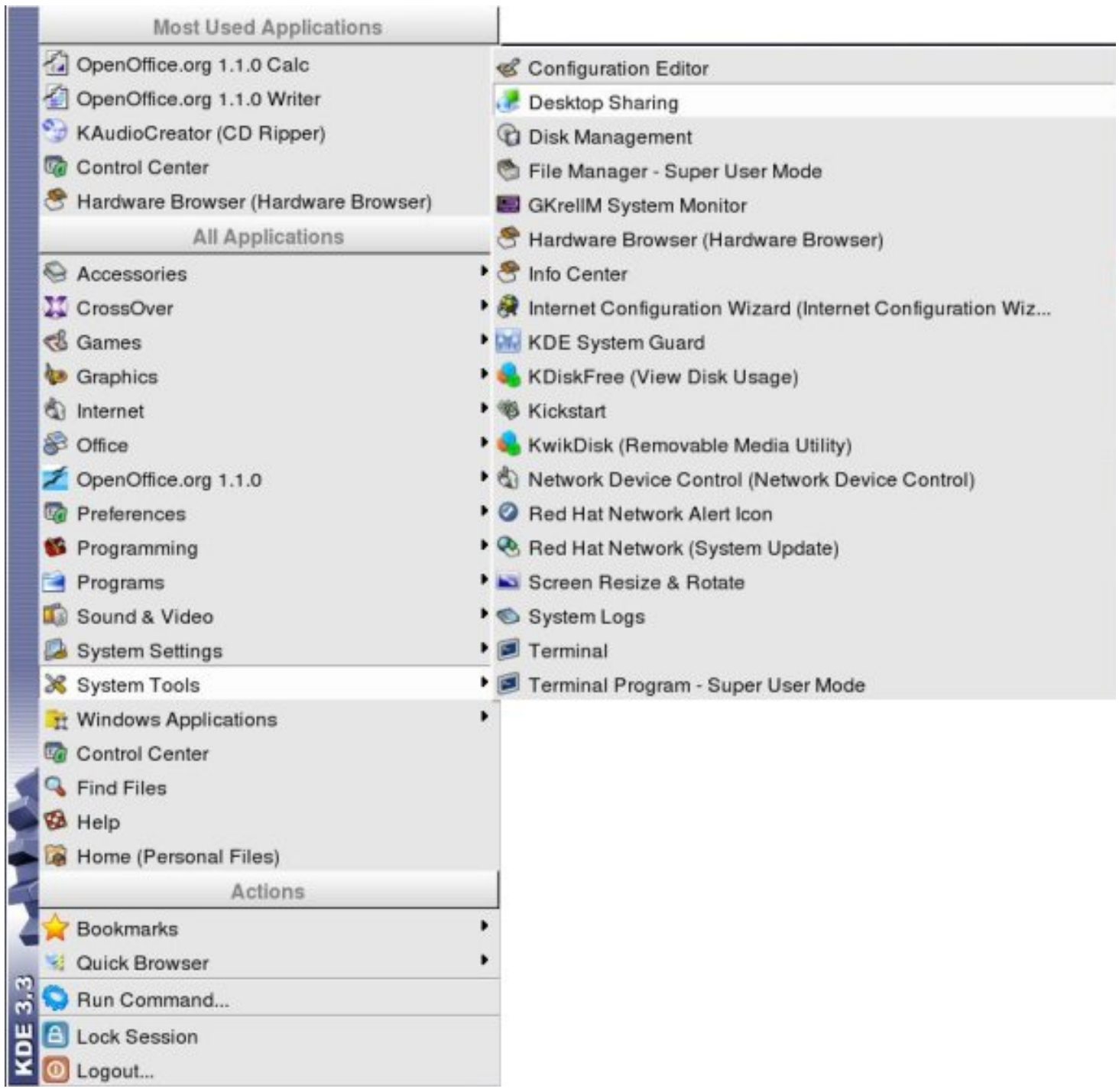
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Putty

Linux System Administration

Desktop Sharing

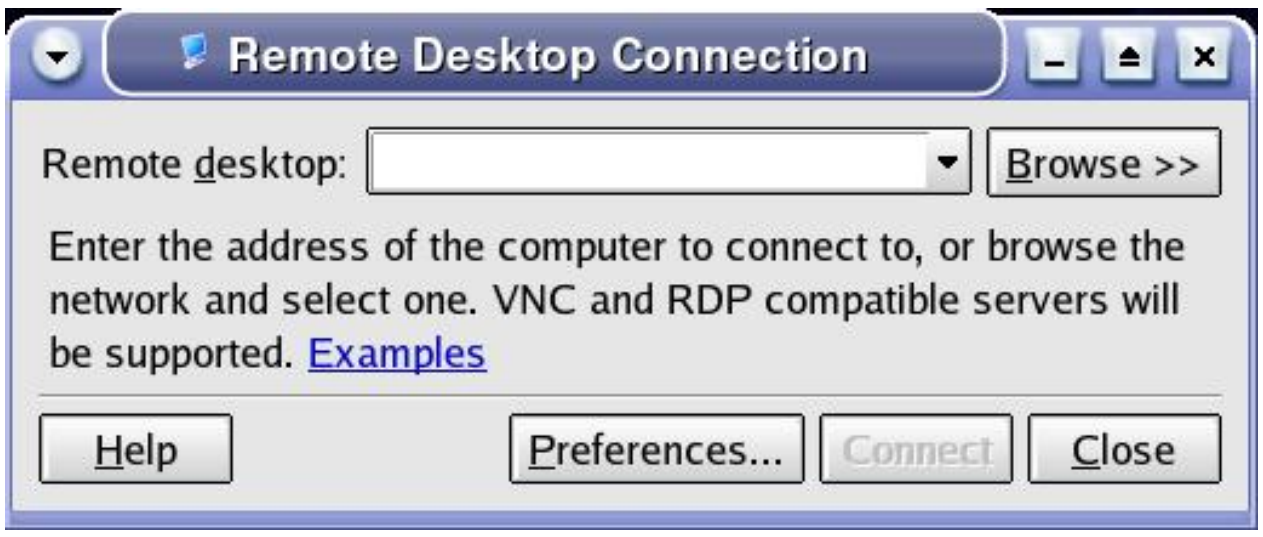
- You can access and administer to a remote system through Desktop Sharing.
- This allows you to literally take control of the desktop.
- From here you can work as if you are on the remote workstation.
- Here is where you can find the application on your taskbar:
    - System Tools --> Desktop Sharing



Linux System Administration

Desktop Sharing

- The client is named : **krdc** (KDE Remoted Desktop Connection)
  - You use the client if you want to connect to another machine.



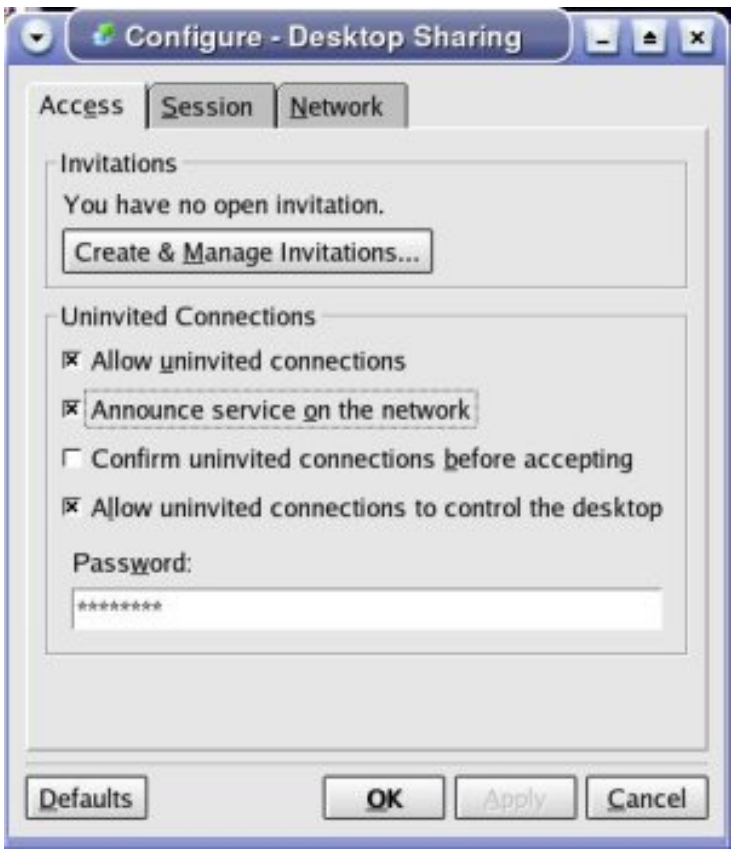
- The server is named : **krfb** (KDE Remote Frame Buffer)
  - The KRfb Server (KDE Desktop Sharing) is a small libvncserver-based app for sharing an X11 session via VNC.
  - You use the server if you want to allow other machines to connect to yours.

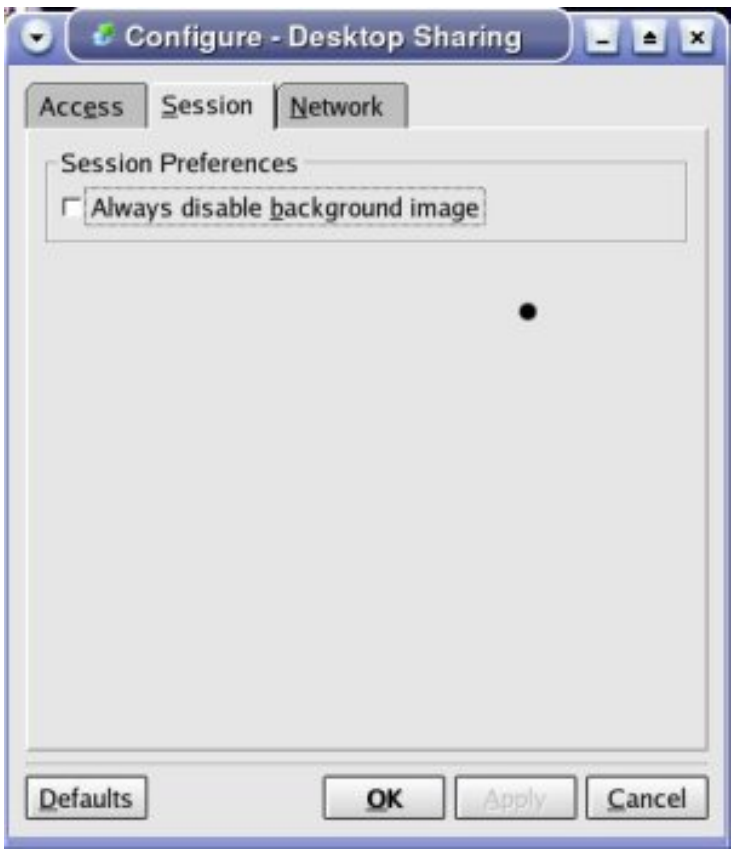
Linux System Administration

Desktop Sharing

The KRfb Server

- We first need to setup our server to accept clients. Click on the **Configure** button in the lower left hand corner to get the following dialog:





- I chose to do the following:
  - **ACCESS :** Allow uninvited connections
  - **ACCESS :** Allow uninvited connections to control the desktop
  - **ACCESS :** Require a password to access the desktop. You will need to know this in order to log on.
  - **Session :** I did not disable the background image
  - **Network :** I chose to pick the port. I chose port 5900

- You can customize these to your needs.

Linux System Administration

Desktop Sharing

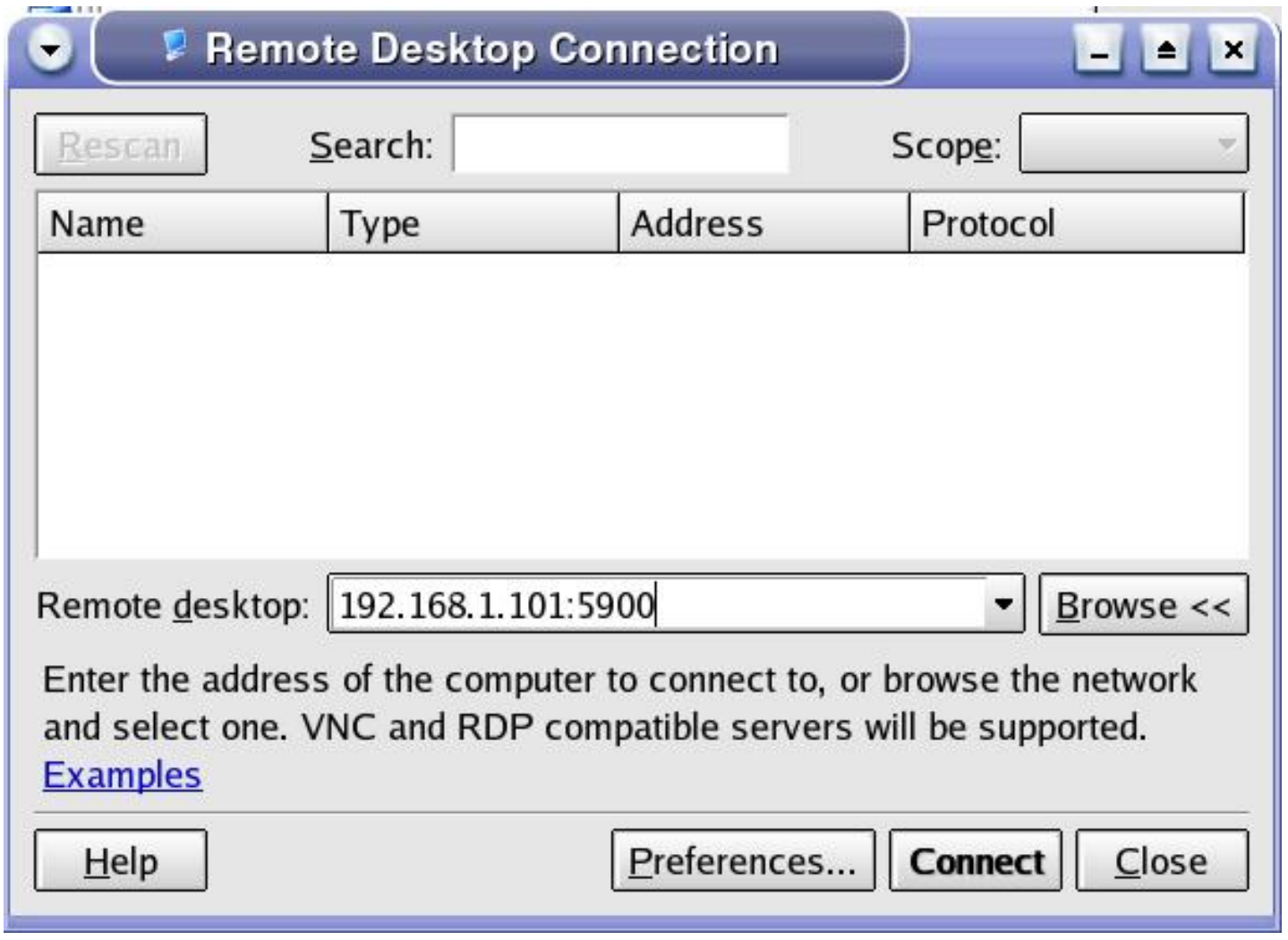
- We are now ready to connect to the server.
- Click on the following on the taskbar : Internet --> Remote Desktop Connection

- Remember you can call this file on the command line using the **krdc** command.

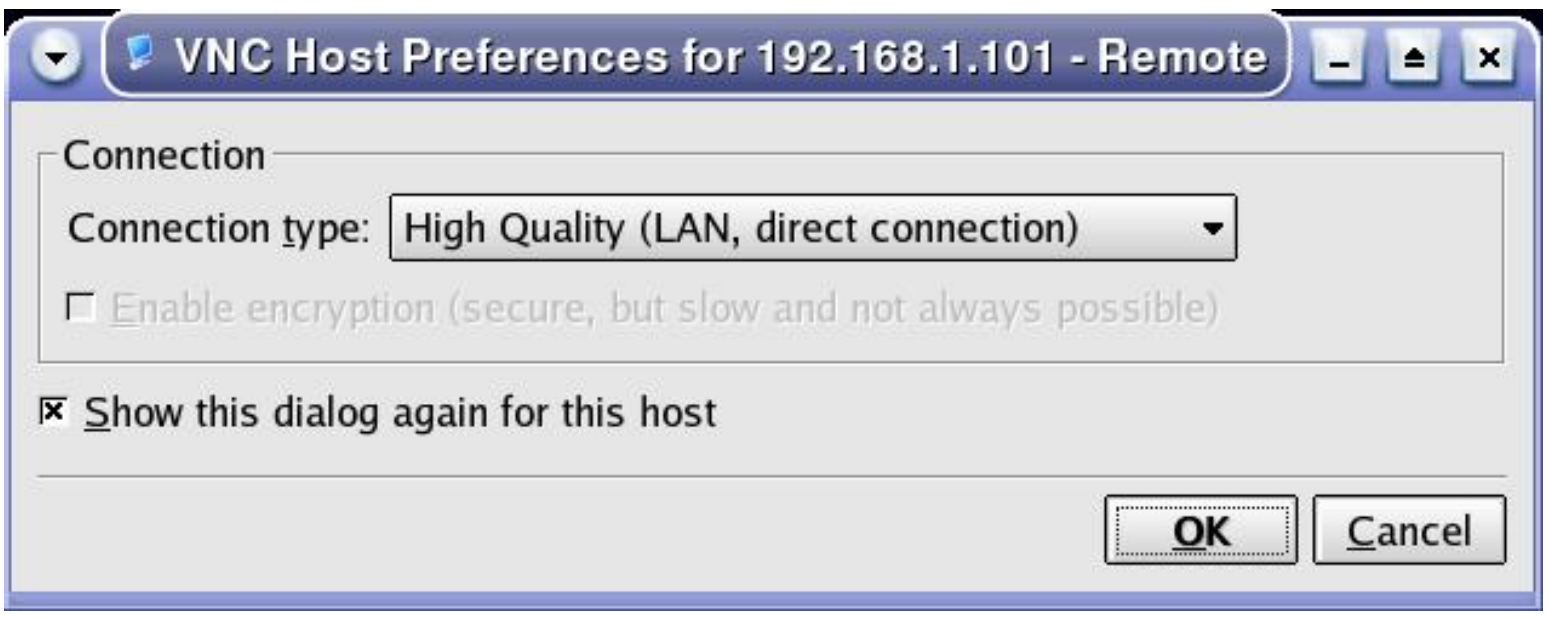
Linux System Administration

Desktop Sharing

- We can now enter the appropriate IP address and port.



- You will then be asked about the connection:

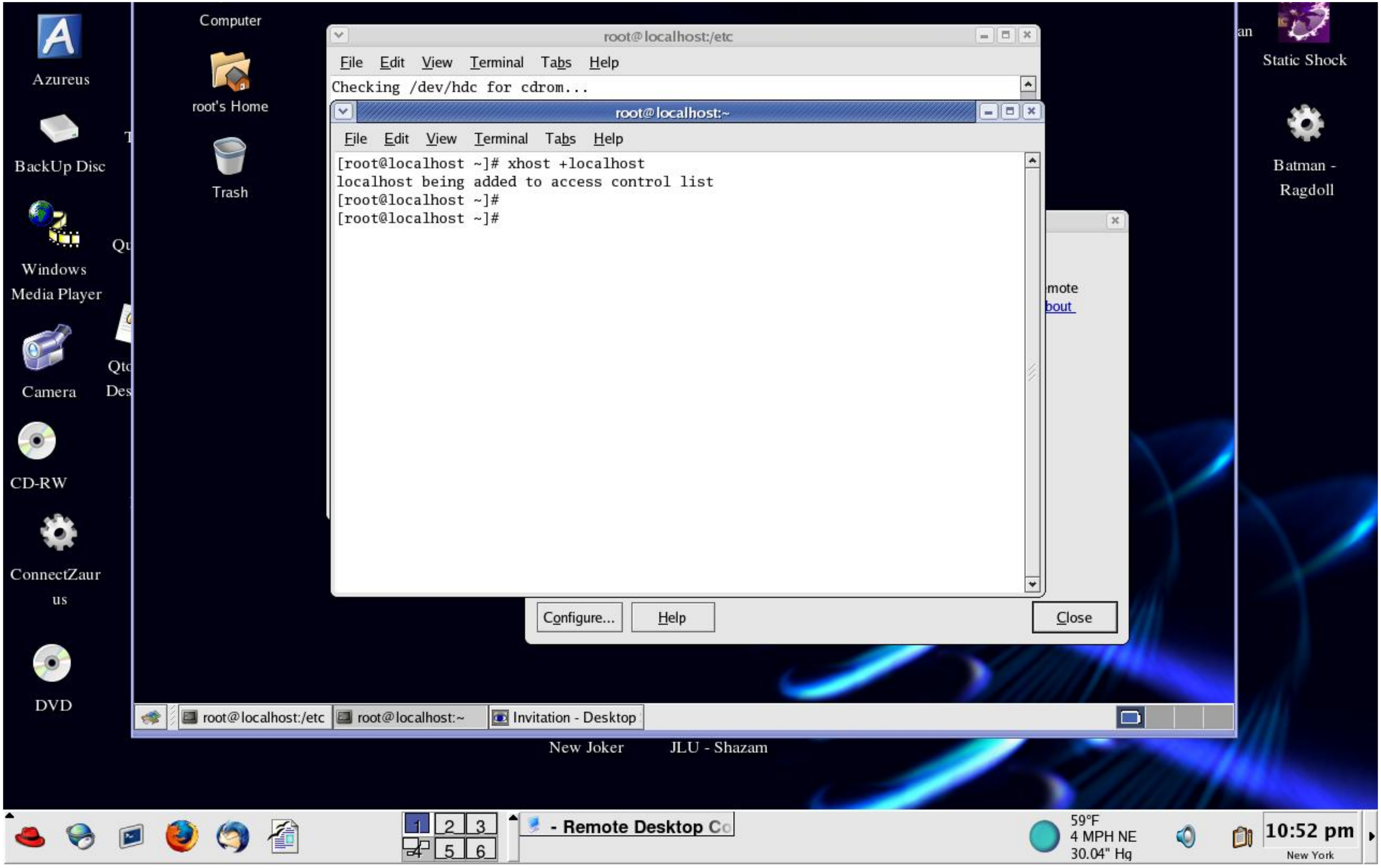


Linux System Administration

Desktop Sharing

- You will then be asked for the password :

- Note that this is not the password of your account on the machine. It is the password specified in the earlier configuration.
- You should now have access to the desktop.


Azureus
BackUp Disc
Windows Media Player
Camera
CD-RW
ConnectZaurus
DVD
Computer
root's Home
Trash
root@localhost:/etc
File Edit View Terminal Tabs Help
Checking /dev/hdc for cdrom...
root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# xhost +localhost
localhost being added to access control list
[root@localhost ~]#
[root@localhost ~]#
Configure...
Help
Close
root@localhost:/etc
root@localhost:~
Invitation - Desktop
New Joker
JLU - Shazam
Static Shock
Batman - Ragdoll
- Remote Desktop Cc
59°F
4 MPH NE
30.04" Hg
10:52 pm
New York

Prev Page 6 Next

Linux System Administration

NFS - Network File System

- It can be efficient for groups of people on a computer network to share common applications and directories needed to do their job.
- One way to store files centrally and share them on a netowrk is by setting up a file server.
- The Network File System (NFS) is a file-sharing protocol for Linux.

Linux System Administration

NFS - Network File System

- Goals of setting up a file server:
  - Centralized distribution - You can add files to one location and have them accessible to any authorized computer or user.
  - Transparency - You can connect remote filesystems to a local filesystem as if it existed on the local system.

Linux System Administration

NFS - Network File System

- How to set up NFS :

1. Set up the network - we already have this set.
2. Choose what to share - You pick a directory (and any subsequent files and subdirectories) that will be accessible to other computers.
3. Set up security on the server - This can be done on many different levels : fstab, /etc/hosts.allow, /etc/exports
4. Mount the file system on the client - The shared filesystem can be mounted anyplace you choose. Choose wisely.

Prev Page 3 Next

Linux System Administration

NFS - Network File System

Setting up an NFS server

- To share an NFS file system, we need to export the filesystem from the server system.

- We can let the NFS server know which directories to export by listing them in the `/etc/exports` file.
  - This file contains a list of entries; each entry indicates a volume that is shared and how it is shared.
  - Check the man pages (man exports) for a complete description of all the setup options for the file.

- An entry in `/etc/exports` will typically look like this:

```
        directory       machine1(option11,option12)
machine2(option21,option22)
```

directory

The directory that you want to share. It may be an entire volume though it need not be. If you share a directory, then all directories under it within the same file system will be shared as well.

Linux System Administration

NFS - Network File System

Setting up an NFS server

```
        directory        machine1(option11,option12)
machine2(option21,option22)
```

machine1 and machine2

These are the client machines that will have access to the directory. The machines may be listed by their DNS address or their IP address (e.g., *machine.company.com* or *192.168.0.8*). Using IP addresses is more reliable and more secure.

optionxx

the option listing for each machine will describe what kind of access that machine will have. Important options are:

- **ro**: The directory is shared read only; the client machine will not be able to write to it. This is the default.
- **rw**: The client machine will have read and write access to the directory.
- **no_root_squash**: By default, any file request made by user `root` on the client machine is treated as if it is made by user `nobody` on the server. (Excatly which UID the request is mapped to depends on the UID of user "nobody" on the server, not the client.) If **no_root_squash** is selected, then root on the client machine will have the same level of access to the files on the system as root on the server. This can have serious security implications, although it may be necessary if you want to perform any administrative work on the client machine that involves the exported directories. You should not specify this option without a good reason.
- **no_subtree_check**: If only part of a volume is exported, a routine called subtree checking verifies that a file that is requested from the client is in the appropriate part of the volume. If the entire volume is exported, disabling this check will speed up transfers.
- **sync**: By default, all but the most recent version (version 1.11) of the **exportfs** command will use **async** behavior, telling a client machine that a file write is complete - that is, has been written to stable storage - when NFS has finished handing the write over to the filesysytem. This behavior may cause data corruption if the server reboots, and the **sync** option

prevents this.

Prev Page 5 Next

Linux System Administration

NFS - Network File System

Setting up an NFS server

- Suppose we have two client machines, *slave1* and *slave2*, that have IP addresses *192.168.0.1* and *192.168.0.2*, respectively.

- We wish to share our software binaries and home directories with these machines. A typical setup for `/etc/exports` might look like this:

```
/usr/local   192.168.0.1(ro) 192.168.0.2(ro)
/home        192.168.0.1(rw) 192.168.0.2(rw)
```

- Here we are sharing `/usr/local` read-only to slave1 and slave2, because it probably contains our software and there may not be benefits to allowing slave1 and slave2 to write to it that outweigh security concerns.

- On the other hand, home directories need to be exported read-write if users are to save work on them.

Linux System Administration

NFS - Network File System

Setting up an NFS server

- If you have a large installation, you may find that you have a bunch of computers all on the same local network that require access to your server.
- There are a few ways of simplifying references to large numbers of machines.
  - First, you can give access to a range of machines at once by specifying a network and a netmask.
  - For example, if you wanted to allow access to all the machines with IP addresses between *192.168.0.0* and *192.168.0.255* then you could have the entries:

```
/usr/local
192.168.0.0/255.255.255.0(ro)
/home
192.168.0.0/255.255.255.0(rw)
```

  - Wildcards are also available for use:

```
/home       *.transy.edu(rw)
```

  - You can also use partial IP's. This represents all IP's that look like 192.168.*.* :

```
/home       192.168.(rw)
```

Linux System Administration

NFS - Network File System

Setting up an NFS server : `/etc/hosts.allow` and `/etc/hosts.deny`

- These two files specify which computers on the network can use services on your machine.
- Each line of the file contains a single entry listing a service and a set of machines.
- When the server gets a request from a machine, it does the following:
  - It first checks `hosts.allow` to see if the machine matches a description listed in there. If it does, then the machine is allowed access.
  - If the machine does not match an entry in `hosts.allow`, the server then checks `hosts.deny` to see if the client matches a listing in there. If it does then the machine is denied access.
  - If the client matches no listings in either file, then it is allowed access.

Linux System Administration

NFS - Network File System

Setting up an NFS server : `/etc/hosts.allow` and `/etc/hosts.deny`

- In addition to controlling access to services handled by **inetd** (such as telnet and FTP), this file can also control access to NFS by restricting connections to the daemons that provide NFS services.
- Restrictions are done on a per-service basis.
- The first daemon to restrict access to is the `portmapper`.
- This daemon essentially just tells requesting clients how to find all the NFS services on the system.
- Restricting access to the portmapper is the best defense against someone breaking into your system through NFS because completely unauthorized clients won't know where to find the NFS daemons.

Linux System Administration

NFS - Network File System

Setting up an NFS server : `/etc/hosts.allow` and `/etc/hosts.deny`

- In general it is a good idea with NFS (as with most internet services) to explicitly deny access to IP addresses that you don't need to allow access to.
- The first step in doing this is to add the followng entry to `/etc/hosts.deny`

  `portmap:ALL`

- Some sys admins choose to put the entry **`ALL:ALL`** in the file `/etc/hosts.deny`
  - This causes any service that looks at these files to deny access to all hosts unless it is explicitly allowed.
  - While this is more secure behavior, it may also get you in trouble when you are installing new services.
  - You may forget you put it there, and you can't figure out for the life of you why they won't work.

Linux System Administration

NFS - Network File System

Setting up an NFS server : `/etc/hosts.allow` and `/etc/hosts.deny`

- Next, we need to add an entry to `hosts.allow` to give any hosts access that we want to have access.
- If we just leave the ALL:ALL entry in `/etc/hosts.deny` then nobody will have access to NFS.
- Entries in `/etc/hosts.allow` follow the format :

```
service: host [or network/netmask] , host [or network/netmask]
```

- For example :

```
portmap: 192.168.0.1 , 192.168.0.2
sshd: 192.168.0.0/255.255.255.0
nfsd: 192.168.0.17
```

Linux System Administration

NFS - Network File System

Starting the NFS server :

- Make sure you have the applications installed.
- Make sure networking is up and running.
- Verify the `nfs` is running:

```
[root@Hamming ~]# service nfs
status                          [  OK  ]
rpc.mountd (pid 4858) is running...
nfsd (pid 4852 4851 4850 4849 4848 4847 4846
4845) is running...
rpc.rquotad (pid 4836) is running...
```

- If NFS is not running, then you will see the following:

```
[root@Hamming ~]# service nfs status
rpc.mountd is stopped
nfsd is stopped
rpc.rquotad is stopped
```

- You can start the service as follows:

```
[root@Hamming ~]# service nfs start
Starting NFS
services:                                  [  OK  ]
Starting NFS
quotas:                                    [  OK  ]
Starting NFS
daemon:                                    [  OK  ]
Starting NFS
mountd:                                    [  OK  ]
```

Linux System Administration

NFS - Network File System

Starting the NFS server :

- You can use chkconfig to set up NFS on reboot:

```
[root@Hamming ~]# chkconfig --list nfs
nfs                0:off   1:off   2:off
3:off   4:off   5:off   6:off

[root@Hamming ~]# chkconfig nfs on
[root@Hamming ~]# chkconfig --list nfs
nfs                0:off   1:off   2:on
3:on    4:on    5:on    6:off
```

- Make sure portmap is also running:

```
[root@Hamming ~]# chkconfig --list portmap
portmap            0:off   1:off   2:off
3:on    4:on    5:on    6:off
```

- Notice that portmap is set to on for run levels 3,4, and 5
- You can check the status to see if it is currently running:

```
[root@Hamming ~]# service portmap status
portmap (pid 2367) is running...
```

- If portmap is not running, then turn it on:

```
[root@Hamming ~]# service portmap start
Starting
portmap:                                                   [  OK
]
```

Linux System Administration

NFS - Network File System

NFS : Daemons Needed

- NFS serving is taken care of by five daemons:
  - **rpc.nfsd**: this does most of the work.
  - **rpc.lockd** and **rpc.statd**: these handle file locking.
  - **rpc.mountd** : this handles the initial mount requests.
  - **rpc.rquotad**, which handles user file quotas on exported volumes.
- Most recent Linux distributions will have startup scripts for these daemons.
- The daemons are all part of the nfs-utils package
  - They may be either in the `/sbin` directory or the `/usr/sbin` directory.
- If your distribution does not include them in the startup scripts, then then you should add them, configured to start in the following order:
  - **rpc.portmap**
  - **rpc.mountd**, **rpc.nfsd**
  - **rpc.statd**, **rpc.lockd** (if necessary), and **rpc.rquotad**

Linux System Administration

NFS - Network File System

Verifying that NFS is running

- Query the portmapper with the command **rpcinfo -p**

```
[root@Hamming ~]# rpcinfo -p                                [
OK  ]
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100011    1   udp    947  rquotad
    100011    2   udp    947  rquotad
    100011    1   tcp    950  rquotad
    100011    2   tcp    950  rquotad
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100021    1   udp  38363  nlockmgr
    100021    3   udp  38363  nlockmgr
    100021    4   udp  38363  nlockmgr
    100021    1   tcp  35494  nlockmgr
    100021    3   tcp  35494  nlockmgr
    100021    4   tcp  35494  nlockmgr
    100005    1   udp    963  mountd
    100005    1   tcp    966  mountd
    100005    2   udp    963  mountd
    100005    2   tcp    966  mountd
    100005    3   udp    963  mountd
```

- If you do not at least see a line that says `portmapper`, a line that says `nfs`, and a line that says `mountd` then you will need to backtrack and try again to start up the daemons

Linux System Administration

NFS - Network File System

Changing `/etc/exports`

- If you come back and change your `/etc/exports` file, the changes you make may not take effect immediately.
- You should run the command **exportfs -ra** to force **nfsd** to re-read the `/etc/exports` file.
- If that still doesn't work, don't forget to check `/etc/hosts.allow` to make sure you haven't forgotten to list any new client machines there.

Linux System Administration

NFS - Network File System

Starting the NFS client

- Suppose our server above is called *master.foo.com*,and we want to mount the `/home` directory on *slave1.foo.com*.
- All we have to do, from the root prompt on *slave1.foo.com*, is type:

```
# mount master.foo.com:/home /mnt/home
```

- The directory `/home` on master will appear as the directory `/mnt/home` on *slave1*.
- Remember to make the mount point!
- You can get rid of the file system by typing

```
# umount /mnt/home
```

Linux System Administration

NFS - Network File System

Mounting NFS Filesystems at Boot Time

- NFS file systems can be added to your `/etc/fstab` file the same way local file systems can.
- The only difference is that the file system type will be set to **`nfs`** and the dump and fsck order (the last two entries) will have to be set to zero.
- The entry in `/etc/fstab` might look like:

```
# device          mountpoint      fs-type
options       dump fsckorder
...
master.foo.com:/home   /mnt     nfs
rw              0    0
```

Linux System Administration

NFS - Network File System

NFS - Mounting Options

- There are some options you should consider adding at once.
- They govern the way the NFS client handles a server crash or network outage.
- There are two distinct failure modes:

soft

If a file request fails, the NFS client will report an error to the process on the client machine requesting the file access. Some programs can handle this with composure, most won't. We do not recommend using this setting; it is a recipe for corrupted files and lost data. You should especially not use this for mail disks --- if you value your mail, that is.

hard

The program accessing a file on a NFS mounted file system will hang when the server crashes. The process cannot be interrupted or killed (except by a "sure kill") unless you also specify **`intr`**. When the NFS server is back online the program will continue undisturbed from where it was. We recommend using **`hard,intr`** on all NFS mounted file systems.

Linux System Administration

NFS - Network File System

NFS - Mounting Options

- Picking up the from previous example, the fstab entry would now look like:

```
   # device                mountpoint  fs-type
options    dump fsckord
   ...
   master.foo.com:/home  /mnt/home   nfs
rw,hard,intr  0     0
   ...
```

Linux System Administration

NFS - Network File System

Props

- Most of the information in this lesson came from the NFS HOWTO:

http://nfs.sourceforge.net/nfs-howto/

Linux System Administration

CUPS and CRON

CUPS

- You can configure a printer with a few tools:
  - system-config-printer
  - system-config-printer-gui
  - system-config-printer-tui
  - lpadmin (command line tool)
- The Common Unix Printing System
  - Access via web browser through port ~~613~~ --> 127.0.0.1:~~613~~
  - Configuration file : /etc/cups/cupsd.conf

PORT 631 NOT 613

Prev Page 1 Next

Linux System Administration

CUPS and CRON

CRON

- This is a daemon used to schedule recurring events.
  - For example: scripts, checking logs, etc.

- Events scheduled by cron are run by the crind daemon, so make sure this daemon is up and running.

<>

- Use `crontab` to edit , install, and view scheduled jobs.
  - Remember to set the EDITOR to emacs or nano if you do not want to use vi.

```
$ export EDITOR=emacs
```

- You can edit the `crontab` with the -e option.

```
$ crontab -e
```

Linux System Administration

CUPS and CRON

/etc/crontab

- Here is an example of what a crontab might look like:

```
[mlevan@localhost ~]$ more /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

The first part is almost self explanatory; it sets the variables for cron.

SHELL is the 'shell' cron runs under. If unspecified, it will default to the entry in the /etc/passwd file.

PATH contains the directories which will be in the search path for cron e.g if you've got a program 'foo' in the directory /usr/cog/bin, it might be worth adding /usr/cog/bin to the path, as it will stop you having to use the full path to 'foo' every time you want to call it.

MAILTO is who gets mailed the output of each command. If a command cron is running has output (e.g. status reports, or errors), cron will email the output to whoever is specified in this variable. If no one if specified, then the output will be mailed to the owner of the process that produced the output.

HOME is the home directory that is used for cron. If unspecified, it will default to the entry in the /etc/passwd file.

Linux System Administration

CUPS and CRON

/etc/crontab

- Here is an example of what a crontab might look like:

```
[mlevan@localhost ~]$ more /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/
# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Here
Now for the more complicated second part of a crontab file. An entry in cron is made up of a series of fields, much like the /etc/passwd file is, but in the crontab they are separated by a space. There are normally seven fields in one entry. The fields are:

minute hour dom month dow cmd

***minute*:** This controls what minute of the hour the command will run on, and is between '0' and '59'

hour: This controls what hour the command will run on, and is specified in the 24 hour clock, values must be between 0 and 23 (0 is midnight)

***dom*:** This is the Day of Month, that you want the command run on, e.g. to run a command on the 19th of each month, the dom would be 19.

month: This is the month a specified command will run on, it may be specified numerically (0-12), or as the name of the month (e.g. May)

***dow*:** This is the Day of Week that you want a command to be run on, it can also be numeric (0-7) or as the name of the day (e.g. sun).

***cmd*:** This is the command that you want run. This field may contain multiple words or spaces.

Linux System Administration

CUPS and CRON

/etc/crontab

If you don't wish to specify a value for a field, just place a * in the field.

minute hour dom month dow cmd

Examples:

```
01 * * * * echo "This command is run at one min past every hour"

17 8 * * * echo "This command is run daily at 8:17 am"

17 20 * * * echo "This command is run daily at 8:17 pm"

00 4 * * 0 echo "This command is run at 4 am every Sunday"

* 4 * * Sun echo "So is this"

42 4 1 * *  echo "This command is run 4:42 am every 1st of the month"

01 * 19 07 *  echo "This command is run hourly on the 19th of July"
```

Linux System Administration

CUPS and CRON

/etc/crontab

Notes:

Under dow 0 and 7 are both Sunday.

If both the dom and dow are specified, the command will be executed when either of the events happen.

For example:

*** 12 16 * Mon command**

Will run cmd at midday every Monday and every 16th, and will produce the same result as both of these entries put together would:

*** 12 16 * * command**

*** 12 * * Mon command**

Linux System Administration

CUPS and CRON

/etc/crontab

Cron also accepts lists in the fields. Lists can be in the form, 1,2,3 (meaning 1 and 2 and 3) or 1-3 (also meaning 1 and 2 and 3).

Example:

59 11 * * 1,2,3,4,5 backup.sh

Will run backup.sh at 11:59 Monday, Tuesday, Wednesday, Thursday and Friday, as will:

59 11 * * 1-5 backup.sh

Cron also supports 'step' values.
A value of **/2* in the dom field would mean the command runs every two days and likewise, **/5* in the hours field would mean the command runs every 5 hours.

Example:

*** 12 10-16/2 * * backup.sh**

is the same as:

*** 12 10,12,14,16 * * backup.sh**

***/15 9-17 * * * connection.test**

Will run connection.test every 15 mins between the hours or 9am and 5pm

Linux System Administration

CUPS and CRON

/etc/crontab

Lists can also be combined with each other, or with steps:

*** 12 1-15,17,20-25 * * command**

Will run cmd every midday between the 1st and the 15th as well as the 20th and 25th (inclusive) and also on the 17th of every month.

*** 12 10-16/2 * * backup.sh**

is the same as:

*** 12 10,12,14,16 * * backup.sh**

When using the names of weekdays or months, it isn't case sensitive, but only the first three letters should be used, e.g. Mon, Sun or Mar, Jul.

Comments are allowed in crontabs, but they must be preceded with a '#', and must be on a line by them self.

Linux System Administration

CUPS and CRON

Restricting Use to `cron`

- Cron has a built in feature of allowing you to specify who may, and who may not use it.
- It does this by the use of */etc/cron.allow* and */etc/cron.deny* files.
- These files work the same way as the allow/deny files for other daemons do.
- To stop a user using cron, just put their name in cron.deny, to allow a user put their name in the cron.allow.
- If you wanted to prevent all users from using cron, you could add the line *ALL* to the cron.deny file.

```
root@localhost # echo ALL >>/etc/cron.deny
```

- If you want user **mlala** to be able to use cron, you would add the line **mlala** to the cron.allow file:

```
root@localhost # echo mlala >>/etc/cron.allow
```

Linux System Administration

CUPS and CRON

Restricting Use to `cron`

- If there is neither a cron.allow nor a cron.deny file, then the use of cron is unrestricted (i.e. every user can use it).
- If you were to put the name of some users into the cron.allow file, without creating a cron.deny file, it would have the same effect as creating a cron.deny file with ALL in it.
- This means that any subsequent users that require cron access should be put in to the cron.allow file.

Linux System Administration

CUPS and CRON

CRON

- Two more commands:
  - `crontab -l` : This command will list your crontab.
  - `crontab -r` : This command will remove your crontab.

Outside Sources : Running Linux, http://www.tech-geeks.org/contrib/mdrone/cron-howto.html

Introduction to Linux Administration

Lab 1 - Installation

Due date: Thursday, 12 May 2005, 06:00 AM (20 days)
Maximum grade: 5

ntroduction to Linux Administration
Lab 1 - Installation
In this exercise, you will install Linux on your workstation. Note that Windows XP is already installed on this system. Our workstations have one hard drive and 512MB of RAM. The hard drive currently has two partitions.
DO NOT WRITE OVER THE FIRST PARTITION!!!!
If you do write over the first partition, then you shall receive a zero for the assignment. Consider this standard throughout the term.
1) Use the Fedora Core 3 boot disk and install Linux on the second partition of the hard drive. Use the following configuration:

/boot	100MB
/usr	2000MB
/home	1000MB
/var	400MB
/	1000MB
swap	512MB

When choosing the packages to install, choose the minimal option. I will provide the root password in class.
You will need to use the NFS install option. Here is some pertinent information:
NFS Server IP :
NFS directory : /var/ftp/pub
Note that this will give you a very basic stripped-down Linux installation. You will not have any X-windows applications at this time. Don't worry. We will be doing multiple installations of Linux and will get to it soon.

Introduction to Linux Administration

Lab 2 -chkconfig

Due date: Thursday, 12 May 2005, 07:00 AM (19 days 23 hours)
Maximum grade: 25

```
Introduction to Linux Administration
Lab 2 - chkconfig
Due Date: Wednesday, May 4th
Points : 25
1.Using chkconfig, determine which processes are currently active on your system.
2.For each service, write a small paragraph describing the purpose of the process.
3.Determine which processes you feel are necessary if you are going to use Linux as
an ordinary desktop machine. Write a paragraph detailing why you do (or do not)
decide that you need a certain process. In other words, give a good justification to
your answer.
4.For the purpose of this assignment, do not concern yourself with the xinet.d
srevices.
5.Your report must be written using Open Office. You will need to e-mail your report
to me by noon on Wednesday.
6.You are to work with your lab partner. (I.e., the other person in your row.)
```

Lab 3 - Adding Partitions

Due date: Thursday, 12 May 2005, 07:00 AM (19 days 23 hours)
Maximum grade: 15

```
Introduction to Linux Administration
Lab 3 – Adding New Partitions
Points : 15
1.Create a new partition that is 250MB in size, and is of type ext2.
2.Refresh the partition table. Make sure you use the correct filesystem type (ext2).
3.Create a new filesystem on the partition.
4.Create the mount point for the new partition. Call the directory NEW and put it in
the HOME directory.
5.Mount the new directory.
6.Download some random images and place them in the new directory.
7.Unmount the directory.
8.Make an appropriate entry in /etc/fstab.
9.Reboot to see if the NEW directory is mounted. (Hint : It should be!)
10. Crash your system by using the power button. On reboot, watch to see which
partitions  are checked. Try to time how long it takes this new partition to pass its
integrity test.
11. Conver t the new partition from ext2 to ext3.
12. Update the change in /etc/fstab.
13. Crash the system again, and determine which filesystems are checked during re-
boot. Is the time any faster than before?
```

Lab 4 - Raid & LVM

Due date: Thursday, 12 May 2005, 07:05 AM (19 days 23 hours)
Maximum grade: 20

```
Introduction to Linux Administration
Lab 4 – RAID & LVM
Points : 20
1.Create 3 partitions of sizes 50MB, 75MB, 90MB to combine into a RAID-0 array.
2.Create the RAID and make sure it is running when the machine is re-booted.
Remember the order of the commands:
fdisk
partprobe
mdadm –create
/etc/mdadm.conf
mdadm -As
mdadm -S
mke2fs
mkdir
mount
/etc/fstab
3.Create 2 partitions of size 100MB and 150MB to create a logical volume.
4.Create the logical volume and make sure it is active when the machine is re-booted.
Make the logical volume 200MB.
Remember the order of the commands:
fdisk
partprobe
pvcreate
vgscan
vgcreate
vgdisplay
lvcreate
mke2fs
mkdir
/etc/fstab
mount
5.Increase the size of the drive to 250MB.
6. Decrease the size of the drive by 20MB.
```

Introduction to Linux Administration

Lab 5 - User Administration

Due date: Thursday, 12 May 2005, 07:05 AM (19 days 23 hours)
Maximum grade: 15

```
Introduction to Linux Administration
Lab 5 – User Administration
Points : 15
Use the command line functions to carry out this assignment.
1. Create an account for the person in your row, as well as for me. Use their Transy
ID names. For example, mine should be mlevan. Set my user ID to 1000. Set the
password to be Linux2005. Let your partner pick their password.
2. Set up a limit to how much space these two new users can have in their home
directory. Make the quota 100MB for both new users. Note that you will have to do a
little research on your own to get this done.
3. Make a new group called TRANSY. Place the three regular users on your workstation
into this group. Set the group ID to 666.
4. Create a new directory in /home called Pioneer, and make the directory owned by
the TRANSY group.
5. Have mlevan create a file in the Pioneer directory that has permissions  -rwxrwx---
. Make sure the file is owned by the group TRANSY.
6. Disable the mlevan account.
```

Lab 6 - Software Management with RPM

Due date: Monday, 16 May 2005, 03:00 PM (15 days 15 hours)
Maximum grade: 15

```
Introduction to Linux Administration
Lab 6 – Software Management with RPM
```

Points : 15

1. Download the RPM package for XMMS.
2. Use the RPM command to determine the dependent packages.
3. Download all the dependent packages, and install them.
4. Write out all the packages that were needed for step 1.
5. Using the RPM commands, remove all the packages you installed in steps 1 and 3. (I know, I know, it's a pain, but you have to learn)
6. Download and install apt.
7. Using apt, install XMMS.
8. Using apt, unistall XMMS.
9. Using apt, install synaptic.
10. Using synaptic, install XMMS.

Quiz 1

Due date: Thursday, 12 May 2005, 09:00 AM (19 days 21 hours)
Maximum grade: 15

Here was a quiz I gave in class. See how you would do:

Introduction To Linux Administration Quiz

Points: 15

Instructions: Write out the following commands. You are not allowed to use any materials or your workstation. Good luck.

1. Briefly describe the steps of the boot process, from the power on to init.
2. Which file must be edited in order to change the default run level ?
3. What command will remove a non-empty directory?
4. What is the difference in the directories /bin and /sbin?
5. Write the steps needed if you are going to add a second hard drive to your system and want to mount it on the directory /home/Bob.

Lab 7 - SSH

Due date: Tuesday, 17 May 2005, 11:50 AM (14 days 19 hours)
Maximum grade: 10

Introduction to Linux Administration

Lab 7 – SSH

Points : 10

1. Check to see if the packages openssh, openssh-clients, and openssh-server are installed on your system.

2. Make sure the daemon is set to on when the system is booted. If sshd is currently stopped, then turn on the service.

3. See if you can SSH into your account on your partners machine.

4. See if you can transfer a file via scp from your machine to your account on your partner's machine.

5. See if you can use sftp to transfer a file from your machine to your account on your partner's machine.

6. Try to set up your machine with ssh-agent to see if you can log on to your account on your partner's machine without using your password. (If things are not properly configured, then ssh will ask you for the password on your partner's machine.)

7. Try to get an X program running through ssh.

Introduction to Linux Administration

Quiz 2

Due date: Tuesday, 17 May 2005, 11:50 AM (14 days 19 hours)
Maximum grade: 15

Here is another quiz to see if you can remember what is going on!

Heart,
Mike

Points: 15

Instructions: Write out the following commands. You are not allowed to use any materials or your workstation. Good luck.

1. Oh, no! You just upgraded the Windows side of your dual-boot machine. Unfortunately, Windows wrote over the Master Boot Record. This means the boot loader that loaded both OS's is gone, and only Windows will boot. List the steps needed to re-install the bootloader.

2. Which file can be checked in order to see if someone has been trying to hack into your system through ssh?

3. You are looking for the file blah.txt on your system. It could be in any directory. What command will look for the file on you entire system?
4. What command will show you who is logged on to the system and what they are doing?
5. What command will show you all the commands you have done in your shell? What could you type if you wanted to run the 467th command again?

Lab 8 - Desktop Sharing

Due date: Wednesday, 18 May 2005, 11:00 AM (13 days 19 hours)
Maximum grade: 10

Hi all,

We will conclude our remote access by talking about remote desktop access.

Mike

Introduction to Linux Administration
Lab 8 – Desktop Sharing

```
Points : 10

1. Make sure you have the packages you need installed on your system.
If not, download and install the needed packages.

2. Configure the server software.

3. Turn on the server if it is not running.

4. Configure your system so the vncserver is activated at boot time.

5. Make a connection with your partner.
```

Introduction to Linux Administration

Lab 9 - NFS

Due date: Thursday, 19 May 2005, 11:00 AM (12 days 19 hours)
Maximum grade: 15

Introduction to Linux Administration
Lab 9 - NFS
Points : 15

1. Make sure you have the packages you need installed on your system. If not, download and install the needed packages.

2. Make sure the appropriate services are running and that they will be turned on the next time you boot the system.

3. Create a directory /NFS-SHARE1. This is the directory that you will allow others to mount.

4. Create a directory /NFS-SHARE2. This is the directory that you will mount your partner's NFS filesystem.

5. Edit the /etc/exports file to show the NFS filesystem you are going to allow others to mount. Add any appropriate options. Allow anyone from our class to access the NFS share. Remember to export the new NFS filesystem.

6. Edit /etc/hosts.deny to disallow anyone to access your system. Note this will affect all TCP programs such as SSH.

7. Edit /etc/hosts.allow to let your partner access your NFS filesystem. Also let your partner use SSH to get into your machine.

8. Use rpcinfo to make sure everything you need to be running is actually running.

9. Mount your partner's NFS filesystem.

10. Edit /etc/fstab to mount the NFS filesystem at boot time. Be careful of the soft vs. hard issue. You might want to do this one at a time.

11. Reboot to make sure it works.

12. Comment out the NFS line in /etc/fstab.

Lab 10 - CUPS and CRON

Due date: Friday, 20 May 2005, 11:30 AM (11 days 19 hours)
Maximum grade: 10

Introduction to Linux Administration

Lab 10 – CUPS and CRON
Points : 10

1. Using the web interface for CUPS, install the network printer we have in the lab. Print out a test page to make sure it is connected properly.

2. Create a crontab for your regular user. Check your mail to ensure the process is being carried out.

3. Place another your partner's username is the cron.deny file and try to create a crontab for that user. Determine if the crontab is working or not.

4. Remove your partner's name from the deny file and verify that the cron jobs are running.

5. Remove all cron jobs.

Introduction to Linux Administration

Practicum - Day 1

Due date: Monday, 23 May 2005, 11:30 AM (8 days 19 hours)
Maximum grade: 11

Hi all,

Today we will have our first practicum.

The first part is a troubleshooting activity. The second part is a Linux installation.

http://www.cs.transy.edu/levan/Practicum1.sxw

Mike

Introduction to Linux Administration

Practicum - Day 2

Due date: Tuesday, 24 May 2005, 11:30 AM (7 days 19 hours)
Maximum grade: 10

Here is the second day's activity.

All system administration -- all the time!

http://www.cs.transy.edu/levan/Practicum2.sxw

Mike

Introduction to Linux Administration

Paper grade

Due date: Thursday, 26 May 2005, 10:00 AM (5 days 20 hours)
Maximum grade: 100

Introduction to Linux Administration

Final Grade

Due date: Thursday, 26 May 2005, 10:05 AM (5 days 20 hours)
Maximum grade: 100

Introduction to Linux Administration

Here is what I did to everyone's workstations for the troubleshooting aspect of Practicum 1.

1) The root password has been changed! Change the root password so the user can not log in as root. There are a few fixes for this. The easiest is to go into run level 1 and reset the password.

2) Networking is turned off. Turn it on.

chkconfig network on

This will start the network on any subsequent re-boots.

service network start

This command will start the network for this session.

3) I removed the configuration for for the X windows server. This is the file /etc/xorg.conf in Fedora Core 3. In order to generate a new version, run the following:

system-config-display

4) I blanked out the path in the ~/.bash_profile for root. I placed the line

PATH=

in the .bash_profile.

You can look in /etc/skel/.bash_profile to see how it should be set.

5) I removed SSH. Simply re-install the RPM's from the CD's, or use apt to install the RPM's for you.

6) I changed /etc/fstab. I made the mount point for /usr to be /usr-XYZ. Unfortunately, this directory does not exist, so the user kept on getting an error message. Edit /etc/fstab to put the mount point ack to what it should be.

Last modified: Sunday, 29 May 2005, 05:24 PM
