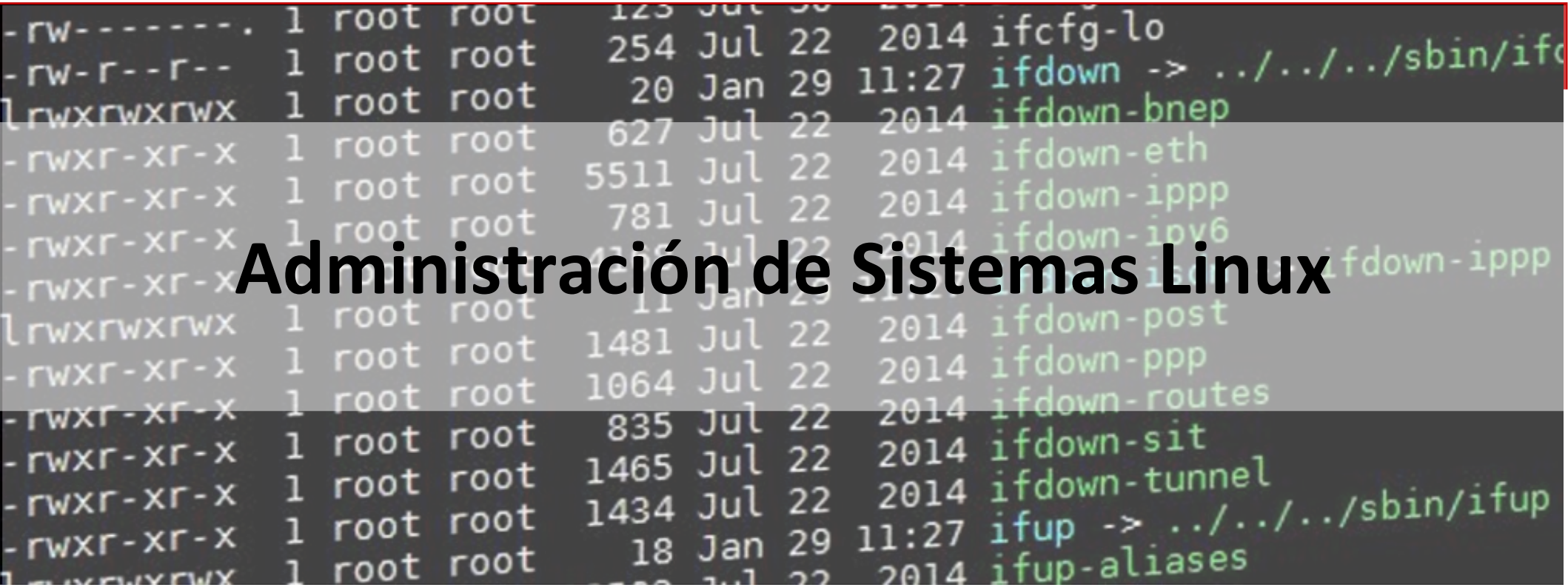
core networks



Administración de Sistemas Linux

LESSON 1

Basic Commands

Basic Linux Commands

The command line is your friend.

Generally the desktop versions as Ubuntu example have a graphical graphic that allows us to manage everything with the mouse.
But there are things that can only be done with the terminal, such as running a script or editing a file on the system.

Basic Linux Commands

Basic Linux Commands

Help and Documentation: man

Man is one of the most useful commands that you can find Linux, that's why we put it first

It is a help command that shows information about the command and the different attributes that can be used.

To prove it we only have to write in the terminal:

 # man command

Basic Linux Commands

List Files and Folders: ls

The next command you should know is ls.
It serves to list the files and folders that are inside the directory in which you are.
If by default you are in / home / they will show you everything inside.
To execute it simply write:
ls / route / del / directory /
or if you are already in said directory:
ls

Basic Linux Commands

Change of Directory: cd

The cd command is used to change directory, for example if you are in / home / directory / and want to go to / home / directory2 /, you should write:
cd / home / directory2 /

If you would like to go to the top directory, read / home /, you can type:
cd ..

Basic Linux Commands

Create a New Directory: mkdir

The mkdir command is used to create a new directory. Only that.
You have to take into account that you create it by default in the address you are in (it always indicates it in the terminal).
If you would like to create it in another directory you should include the route, for example:
 # mkdir / newyourdirectory /
O well
 # mkdir / route / from / newyourdirectory /

Basic Linux Commands

Create a New File: touch

This command is used to create a new empty file if it does not exist.
For example if we want to create a new text file to leave a note, we can type:
 # touch file.txt
if we want to create it in another route:
 # touch /path/file.txt

Basic Linux Commands

Delete a File / Directory: rm

If we want to delete some file or directory, we can use the rm command.
For this we will use:
rm file.txt
O well:
rm /path/file.txt
If we want to delete a directory that contains more files, we can use the -r attribute, this is:
rm -r / directory / or # rm -r /route/del/directory/

Basic Linux Commands

Copy a File / Directory: cp

When copying files we will need the cp command.
We have to indicate the origin route and the destination route, in this order:
cp / sourcepath /.txt file / destinationpath/file.txt

Basic Linux Commands

Move a File / Directory: mv

To move a directory or file we will use this command. This will only move the files without copying them from one directory to another.

It works in the same way as cp, indicating the source route and the destination route:

mv / sourcepath /.txt file / destinationpath/file.txt

Basic Linux Commands

See the Contents of a File: cat

Cat is used to view the contents of a file without editing it. It simply shows us its content without the possibility of changing it.

```
# cat file.txt
```

Basic Linux Commands

Edit a File: vi / nano

Vim and nano are two text editors.

Vim usually comes by default in all systems, although nano may have to install it. this varies from one distribution to another, although as a curiosity in Ubuntu we can do it with # apt-get install nano.

Basic Linux Commands

Edit a File: vi / nano

To execute these editors we will have to use the commands
vi /path/file.txt
O well
nano /path/file.txt
Although nano shows the commands at the bottom of the screen (that's why it's so popular), vim does not show them, so before editing I advise you to type a #man
Better be cautious if we edit something important.

Basic Linux Commands

Switch to Superuser Mode: su

Switch to Superuser Mode: su
The su command changes to superuser or "root" mode. This mode is what you will need if you want to change something important or you need permission to access certain files. It is the system administrator.
When you change to this mode you will ask us for a password, when you type it you will not see anything, so make sure you write it correctly.
su

Basic Linux Commands

Run in Superuser Mode: sudo

It serves for the option to sort.

If you want to trigger an order with administrator privileges you must type this command
sudo command

Basic Linux Commands

Change User Password: passwd

Change the password of the current user. Once again when changing it, make sure you are writing it correctly (are capital letters activated?).

```
# passwd
```

Basic Linux Commands

Change Root Password: sudo passwd

Change the password of the root
user.

sudo passwd

Basic Linux Commands

Compress / Unzip Zip Files:

zip / unzip
Compress or unzip a directory or file in .zip format
zip -r file.zip filesacompress
or
unzip file.zip

Basic Linux Commands

Compress / Unpack Tar:

tar / untar files
It works in a similar way, although with other attributes:
Compress # tar cvf file.rar archivestocompress
Unzip # tar xvf file.rar

Basic Linux Commands

Restart the System: reboot

Restart the operating system.
reboot

Shut down the system: halt

Shut down the operating system completely
halt

Basic Linux Commands

Clean the Terminal: clear

Clean the text of the terminal. It is always good to see what we are doing :)

```
# clear
```

Exit the Terminal: exit

Close the session in the terminal.

```
# exit
```

Basic Linux Commands

This is the list.

There are many more, but with these 20 commands you have enough to move around the terminal a bit.

LESSON 2

System Logging

System Logging

Log files contain messages about the *system*, the *kernel*, *services*, and *applications*.

Some *log files* are controlled by the rsyslogd daemon.

The main configuration file for system logging is /etc/rsyslog.conf, which contains global directives, modules, and rules.

Some log files are controlled by the rsyslogd daemon.

Configuration

Global Directives

- specify configuration options that apply to the rsyslogd daemon.

Modules

- rsyslog has a modular design. This enables functionality to be dynamically loaded from modules.

```
$ModLoad <MODULE>
```

Rules

- A rule is specified by a *filter* part, which selects a subset of rsyslog messages, and an *action* part, which specifies what to do with the selected messages.

Configuration

Filters

- rsyslog offers various ways to filter rsyslog messages according to various properties. A defined filter is called a selector.

Actions

- Actions specify what is to be done with the filtered messages.

Templates

- Any output that is generated by rsyslog can be modified and formatted by using templates.

Facility/Priority

- Messages are filtered based on two conditions: ***Facility and priority*.**

Facility	Priority
auth	
cron	debug
daemon	info
kern	notice
lpr	warning
mail	err
news	crit
syslog	alert
user	emerg

Facility/Priority

- Examples:

  kern.*
  mail.crit
  cron.!info,!debug

- /etc/rsyslog:

```
# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

Actions

- Save messages to log files
- Send messages over the network
- Send messages to specific users
- Execute a program
- Input messages to a database
- Discard messages

Examples:

```
cron.*      /var/log/cron.log
*.*         @host.com:18
```

Templates

Templates modify and format output generated by rsyslog.

- **Syntax:**

  $template <TEMPLATE_NAME>,"text %<PROPERTY>%more text", [<OPTION>]

- **Templates can be used to generate dynamic file names:**

  $template DynamicFile, "/var/log/test_logs/%timegenerated%-test.log“

Templates

- **Example:**
  - $template verbose,"%syslogseverity%,%syslogfacility%,%timegenerated%,%hostname%,%syslogtag%,%msg%\n“
  - *.* /var/log/logfile; verbose
- **/etc/rsyslog.conf:**

```
# A template to for higher precision timestamps + severity logging
$template SpiceTmpl,"%TIMESTAMP%.%TIMESTAMP:::date-subseconds% %syslogtag% %syslogseverity-text%:%msg:::sp-if-no-1st-sp%%msg:::drop-last-lf%\n"

:programname, startswith, "spice-vdagent"       /var/log/spice-vdagent.log;SpiceTmpl
```

Configuring Log Rotation

- Log rotate is a utility to automatically manage log files. - /etc/ logrotate.conf is the global configuration file. - /etc/logrotate.d is a directory with the special configuration files.

- Options:

  - How often to rotate files
  - The number of rotated log files to keep
  - Scripts to run before or after rotating
  - Specify log files to be mailed
  - Enable compression of log files

LESSON 3

User & Group Administration

Introduction to Users and Groups

- User account information is stored in /etc/passwd.
- Group information:
  - Group information is stored in /etc/group.
  - Each user has a private group (UPG).
  - Users can belong to more than one group.
- Linux uses shadow passwords.
  - /etc/shadow: Hashed user passwords
  - /etc/gshadow: Hashed group passwords
  - /etc/login.defs: Security policies

User and Group Configuration Files

- Contents of /etc/passwd:
  - username:x:UID:GID:GECOS:home dir: shell
- Contents of /etc/shadow:
  - username: hashed password: password aging inform.
- Contents of /etc/group:
  - groupname: x: GID: comma-separated members
- Contents of /etc/gshadow:
  - groupname: hashed password: GID: comma-separated administrators: comma-separated members
  - Group passwords are rarely used.

User and Group Configuration Files

```
[root@localhost modules]# tail -4 /etc/passwd
centos:x:500:500:Usuario Centos:/home/centos:/bin/bash
student:x:501:501::/home/student:/bin/bash
usu1:x:502:502::/home/usu1:/bin/bash
usu2:x:503:503::/home/usu2:/bin/bash
[root@localhost modules]# tail -4 /etc/shadow
centos:$6$AztWeYsPCxw9H8pa$hxwRhYtaIz2j1tCEEIWs2k9tdwH6fF2GSP4hXxFiLomnh7yI4O5JxTBKHAYPA
lGEDqJD3s6hiG0dvASK.0FtI.:17052:0:99999:7:::
student:$6$WrLR9nMP$vWNsF2ftOybPWLWtYuWT1SWTVEdXED50cSLtW6RCqzBVtM7PJWsR.Fn/LiDz5TjxOouX
ZnZKdv90bjMlrhD3B.:17054:0:99999:7:::
usu1:!!:17066:0:99999:7:::
usu2:!!:17066:0:99999:7:::
[root@localhost modules]# tail -4 /etc/group
centos:x:500:
student:x:501:
usu1:x:502:
usu2:x:503:
[root@localhost modules]# tail -4 /etc/gshadow
centos:!!::
student:!::
usu1:!::
usu2:!::
[root@localhost modules]#
```

Adding a User Account

- To add a user:
  - useradd [options] *user_name*
- To create a password:
  - passwd [options] *user_name*
- User default settings are stored in:
  - /etc/default/useradd
- Use the –D option to display or modify defaults:
  - useradd –D [options]
- A new user's home directory is populated with files from:
  - /etc/skel directory
- To create a nologin user:
  - useradd –s /sbin/nologin *user_name*

Modifying or Deleting User Accounts

- To modify a user:

  usermod [options] *user_name*

  – Example:

  usermod –aG 517 *user_name*

- To delete a user:

  userdel [options] *user_name*

- Options to userdel include:

  -f: Force removal even if user is logged in -r:

  Remove the user's home directory

Group Account Administration

- To add a group account:

  groupadd [options] *group_name*

- To modify a group account:

  groupmod [options] *group_name*

- To delete a group account:

  groupdel *group_name*

- To administer group accounts:

  gpasswd [options] *group_name*

  – Example: To add a user (jim) to a group (students):

  gpasswd –a jim students

- The groups command prints the groups to which a user belongs.
- The newgrp command changes the real group identification.

User Private Groups

Each user belongs to a unique group.

- Eliminates the need for `umask=0022`

Allows `umask=0022`

Additional steps to implement:

Create a directory to share.

Create a new group.

Add users to this new group.

Change the group ownership for the directory.

Set the setgid bit on the directory.

Password Configuration

- Password aging requires users to change their password.
- Use the **chage** command to configure password aging:
  - **`chage [options] user_name`**
- Current values are displayed and changed interactively:
  - `Minimum Password Age [0]:`
  - `Maximum Password Age [99999]:`
  - `Last Password Change [2015-11-06]:`
  - `Password Expiration Warning [7]:`
  - `Password Inactive [-1]:`
  - `Account Expiration Date [1969-12-31]:`

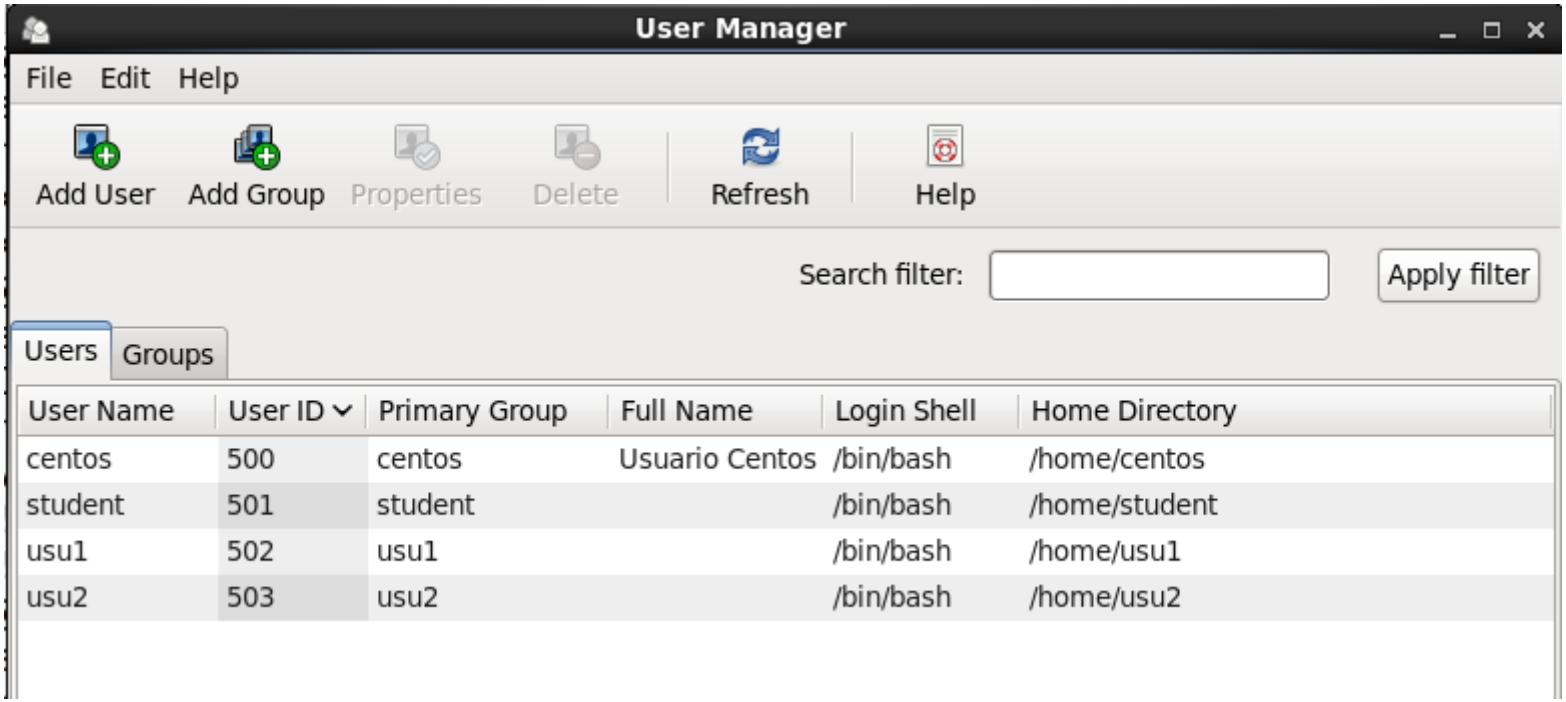
Use the **authconfig** command to configure the password hashing algorithm:

```
authconfig --passalgo=<algorithm> --update
```

The /etc/login.defs File

- The `/etc/login.defs` file provides default user account settings.
- Default values include:
  - Location of user mailboxes
  - Password aging controls
  - Values for automatic UID selection
  - Values for automatic GID selection
  - User home directory creation options
  - `umask` value
  - Encryption method used to encrypt passwords

User Manager Tool



The system-config-users command starts User Manager.

LESSON 4

File Systems

Partition Table Utilities

Linux has three utilities to partition disks.

```
fdisk
cfdisk
parted
```

It should not partition a device that is in use.

Disk Partitions

Partitioning divides a disk drive into logical disks.

The system disk must have at list 3 partitions:

`/` (root) `/boot` `swap`

The original partitioning scheme for PC hard disks allowed only **four** partitions, called *primary* partitions.

To create more than **four** partitions, one of these four partitions can be divided into many smaller partitions, called **logical** partitions.

fdisk -l

```
[root@localhost ~]# fdisk -l /dev/sda

Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00026bdf

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          64      512000   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              64        1339    10240000   83  Linux
/dev/sda3            1339        1977     5120000   83  Linux
/dev/sda4            1977        2611     5098496    5  Extended
/dev/sda5            1977        2611     5097472   82  Linux swap / Solaris
[root@localhost ~]#
```

Using fdisk

The `fdisk` utility is interactive.

`fdisk` commands include:

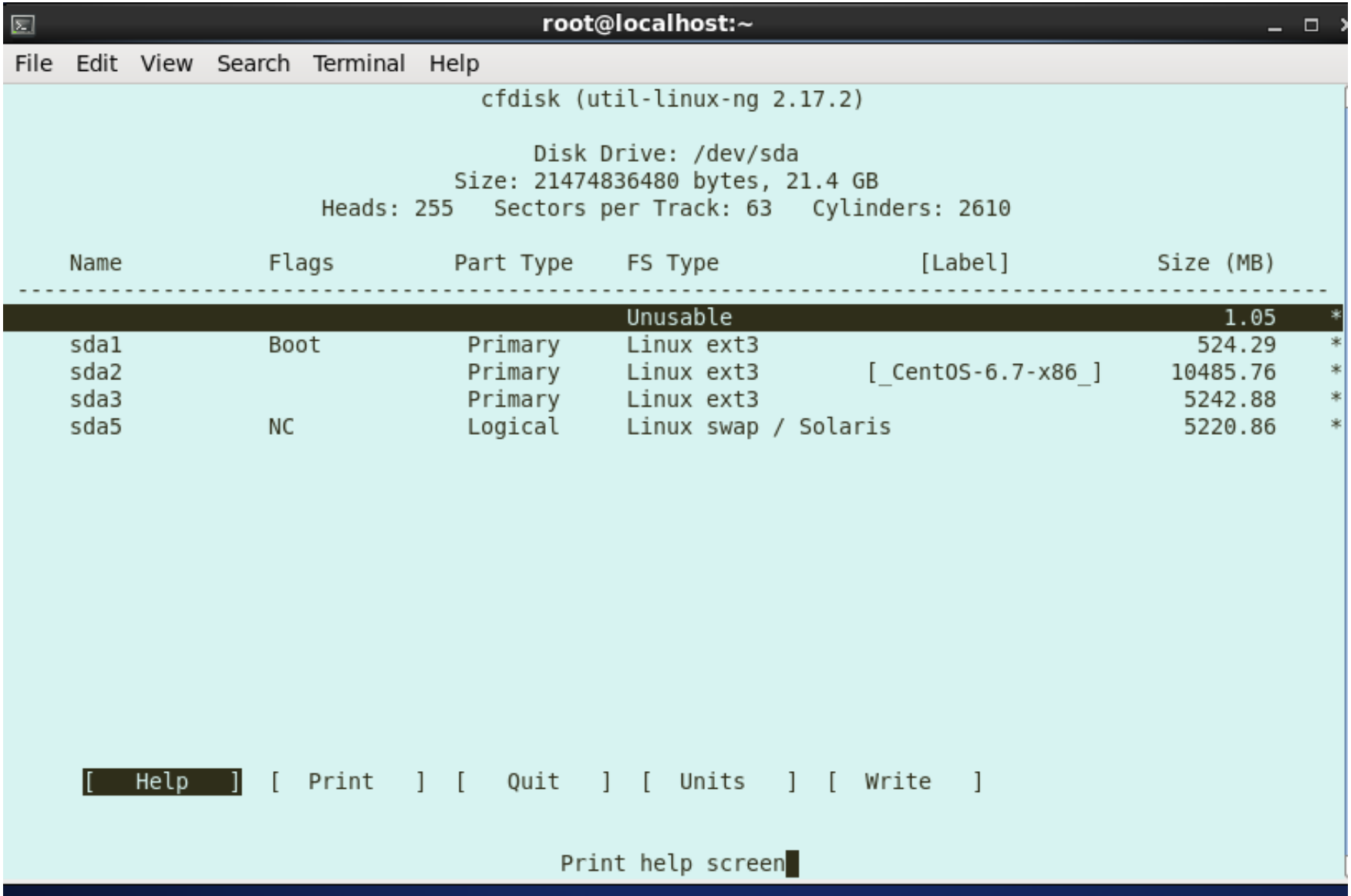
- d: **Delete a partition.**
- l: **List the known partition types.**
- m: **Print the available commands.**
- n: **Add a new partition.**
- p: **Print the partition table.**
- q: **Quit without saving changes.**
- w: **Write the table to disk and exit fdisk.**

Use `partprobe` *`device`* to have the kernel re-read the partition table

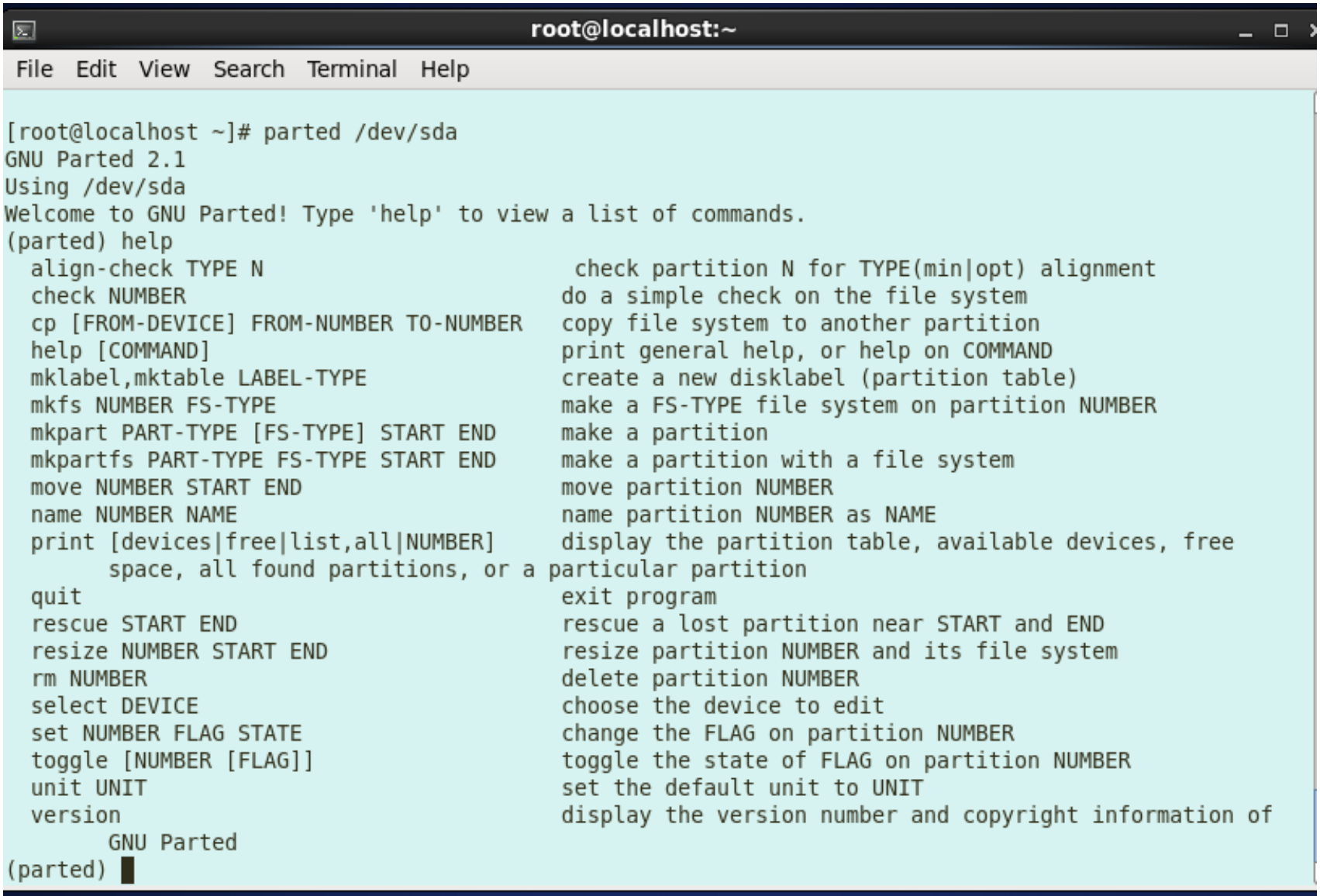
Using fdisk

```
[root@localhost ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

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

Command (m for help):
```

Using cfdisk



Using parted



File System Types

ext2

High performance for fixed disk and removable media

ext3

Journaling version of ext2

ext4

Supports larger files and file system sizes

vfat

MS-DOS file system useful when sharing files between Windows and Linux

Btrfs

Addresses scalability requirements of large storage systems

Making File System

The `mkfs` command make a Linux file system.

You can use:

`mkfs.ext2, mkfs.ext3, mkfs.ext4`

`/etc/mk2fs.conf` has the default parameters.

To display the attributes of a block device: `blkid`

To display and modify the file system label: `e2label`

mkfs

```
[root@localhost ~]# mkfs.ext4 /dev/sdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
131648 inodes, 526120 blocks
26306 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=541065216
17 block groups
32768 blocks per group, 32768 fragments per group
7744 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 23 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@localhost ~]#
```

blkid

Example:

```
[root@localhost ~]# blkid /dev/sdb1
/dev/sdb1: UUID="03cb929c-ac80-40da-9f3e-40a0c61ffae5" TYPE="ext4"
[root@localhost ~]#
```

```
[root@localhost ~]# mkfs.ext3 -L PROGRAM /dev/sdb2
```

```
[root@localhost ~]# blkid /dev/sdb2
/dev/sdb2: LABEL="PROGRAM" UUID="ffa98b7e-0877-4418-ad31-40a44a3e7463" SEC_TYPE="ext2" TYPE="ext3"
[root@localhost ~]#
```

Mounting File Systems

The mount command attach a device to a directory

```
mount [option] device mount_point
```

Examples:

```
mount /dev/sdd1 /dir1
mount UUID="uuid_number" /dir1
mount LABEL="label_name" /dir1

mount -o nouser,ro /dev/xvdd1 /test
```

To unmount a file system:

```
umount /dev/sdd1
```

Mounting File Systems

Example:

```
[root@localhost ~]# mkdir /disco_b1
[root@localhost ~]# ls /disco_b1
[root@localhost ~]# mount /dev/sdb1 /disco_b1
[root@localhost ~]# ls /disco_b1
lost+found
[root@localhost ~]# touch /disco_b1/file_1
[root@localhost ~]# ls /disco_b1
file_1  lost+found
[root@localhost ~]# umount /disco_b1
[root@localhost ~]# ls /disco_b1
[root@localhost ~]#
```

Swap space

Swap space is used when there is insufficient RAM.

Swap space is a partition, a file, or both.

To create a swap partition use: `fdisk`, `cfdisk`, **or** `parted`

To create a swap file:

```
# dd if=/dev/zero of=/swapfile bs=1024 count=1000000
```

To initialize a swap partition or file: `mkswap {device|file}`

To enable and disable devices for swapping:

```
swapon {device|file} swapoff
{device|file}
```

Swap space

Example:

```
[root@localhost ~]# dd if=/dev/zero of=/swapfile bs=1024 count=1000000
1000000+0 records in
1000000+0 records out
1024000000 bytes (1.0 GB) copied, 7.24588 s, 141 MB/s
[root@localhost ~]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/sda5                               partition       5097468 0       -1
[root@localhost ~]# mkswap -f /swapfile
Setting up swapspace version 1, size = 999996 KiB
no label, UUID=7b3d041c-eff4-4ae8-9b07-65fb59b3403b
[root@localhost ~]# swapon -a /swapfile
[root@localhost ~]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/sda5                               partition       5097468 0       -1
/swapfile                               file            999996  0       -2
```

LESSON 5

Access Security and Permissions

Authentication

- **Authentication** is the verification of the identity of a user.

- A user logs in by providing a *username* and a *password* and is **authenticated** by comparing this information to data stored on the system.

- If the login credentials match and the user account is active, then the user is authenticated and can successfully access the system.

Controlling Access to Systems

There are multiple ways in which you can control access to a system.

- Securing **logins** and **passwords**
- Changing the **password algorithm**

Login and Password

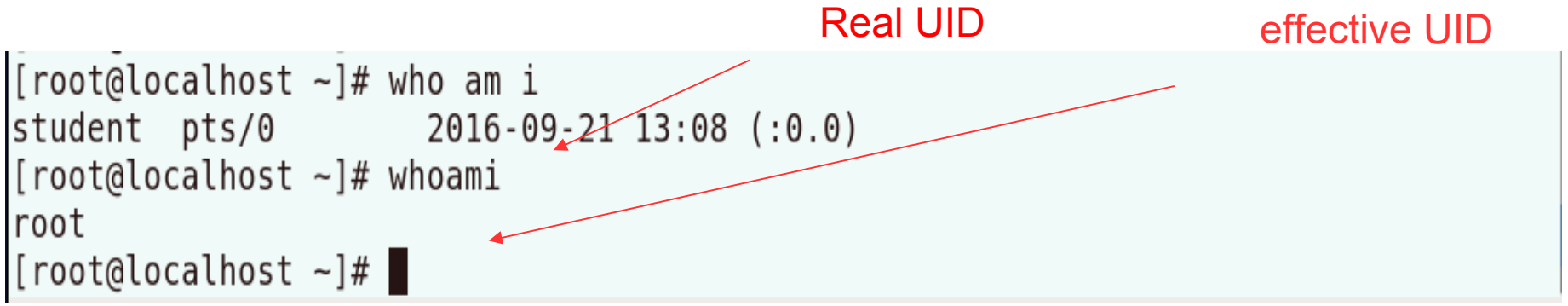
- The ***login*** command:
  - Verifies the username and password
  - Denies access to the system if the username and/or password are incorrect.
- Ensure that all the accounts on a system have a password.
- ***Passwords*** are kept secure through:
  - Encryption
  - Placement in a separate file from username and other information.

Monitoring System Activities

To control and monitor system activity you should perform the following:

- Setting limits on who can use what resources
- Logging resource use
- Monitoring who is using the resources

The system tracks real and effective user and group ID logins.



Restricting Use of the su Command

- You can limit access to the su command to only those users who are members of the wheel group.
- To limit access to the su command to the student user, add the student user to the wheel group as follows:
  - `usermod -aG wheel student`
- Add the following line to the /etc/pam.d/su file to only permit root access to members of the wheel group:
  - `auth required pam_wheel.so use_uid`

Allowing Use of the sudo Command

- `sudo` privileges are configured in the `/etc/sudoers` file.
- The following entry is present in the `/etc/sudoers` file:
  - `root ALL=(ALL)  ALL`
- The following entry in `/etc/sudoers` allows the `student` user to use `sudo` to run administrative commands:
  - `student ALL=(ALL)  ALL`
- The `student` user can now run administrative commands by preceding the command with `sudo`, for example:

```
$ sudo useradd new_user
[sudo] password for student:
```

- You are prompted for the `student` user password, not the `root` user password.

Controlling Access to Files

To secure files and directories in Linux, you can use:

- UNIX file permissions
- Access control lists (ACLs)

To see and modify the file permission Linux has:

- ls command
- chown command
- chgrp command
- chmod command

Files Types

Symbol	Description
`b`	Block special file
`c`	Character special file
`d`	Directory
`l`	Symbolic link
`s`	Socket
`D`	Door
`P`	Named pipe
`-` (minus sign)	Regular text file or a program

File Permissions

Symbol	Permission	Object	Description
`r`	Read	File	Designated users can open and read the contents of a file.
		Directory	Designated users can list the files in the directory.
`w`	Write	File	Designated users can modify the contents of the file or delete the file.
		Directory	Designated users can add files or add links in the directory. They can also remove files or remove links in the directory.
`x`	Execute	File	Designated users can execute the file, if it is a program or shell script.
		Directory	Designated users can open files or execute files in the directory. Users can `cd` into the directory.
`-`	Denied	File and Directory	Designated users cannot read, write, or execute the file.

Special File Permissions

- The special permission types files and directories are:

  - **setuid:** Grants access to the files and directories that are normally available only to the owner.
  - **setgid:** Grants access based on the permissions that are granted to a particular group.
  - **sticky bit:** Protects the files within a directory

File Permissions Modes

- The special permission types for files and directories are:
  - **Symbolic Mode: r**ead, **w**rite, e**x**ecute

```
[student@localhost ~]$ ls -l file_1
-rw-r--r--. 1 student student 1818 Sep 29 18:42 file_1
[student@localhost ~]$ chmod g+w file_1
[student@localhost ~]$ ls -l file_1
-rw-rw-r--. 1 student student 1818 Sep 29 18:42 file_1
[student@localhost ~]$
```

  - **Absolute Mode:** Numbers

```
[student@localhost ~]$ chmod 755 file_1
[student@localhost ~]$ ls -l file_1
-rwxr-xr-x. 1 student student 1818 Sep 29 18:42 file_1
[student@localhost ~]$
```

File Permissions Modes

Octal Value	File Permissions Set	Permissions Description
0	`---`	No permissions
1	`--x`	Execute permission only
2	`-w-`	Write permission only
3	`-wx`	Write and execute permissions
4	`r--`	Read permission only
5	`r-x`	Read and execute permissions
6	`rw-`	Read and write permissions
7	`rwx`	Read, write, and execute permissions

Special File Permissions Modes

Octal Value	Special File Permissions
1	Sticky bit
2	setgid
4	setuid

```
[student@localhost ~]$ chmod 755 file_1
[student@localhost ~]$ ls -l file_1
-rwxr-xr-x. 1 student student 1818 Sep 29 18:42 file_1
[student@localhost ~]$ chmod 4755 file_1
[student@localhost ~]$ ls -l file_1
-rwsr-xr-x. 1 student student 1818 Sep 29 18:42 file_1
[student@localhost ~]$ chmod 2755 file_1
[student@localhost ~]$ ls -l file_1
-rwxr-sr-x. 1 student student 1818 Sep 29 18:42 file_1
[student@localhost ~]$
```

LESSON 6

Network Configuration

Network Interfaces

- Each physical network device has an associated network interface configuration file.
- Network interface configuration files are located in the `/etc/sysconfig/network-scripts` directory.
- Configuration file names are `ifcfg-`*`interface`* where interface is eth0, eth1, ppp0, irlan0, plip0,.....

Network Interfaces

Configuration parameters include:

```
DEVICE=eth0
BOOTPROTO=none
TYPE=Ethernet
HWADDR=00:16:3E:00:01:02
IPADDR=192.0.2.102
NETMASK=255.255.255.0
```

Additional Network Configuration Files

- `/etc/hosts` associates host names with IP addresses.
  - Larger networks would use DNS to perform this resolution.
  - Specify the IP address of the loopback device.
- `/etc/resolv.conf:`
  - Provides access to DNS
  - Identifies DNS name server(s) and search domain
- `/etc/sysconfig/network` specifies routing and host information for all network interfaces.
- `/etc/nsswitch.conf` lists the order of host name searches.

Command-Line Network Interface Utilities

- `ifconfig` is used:
  - At boot time to configure kernel-resident network interface
  - To display the status of an interface
  - To configure (non-persistent) properties

```
[root@localhost modules]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:F4:22:DD
          inet addr:192.168.28.132  Bcast:192.168.28.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef4:22dd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:92903 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29418 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:67648257 (64.5 MiB)  TX bytes:1671953 (1.5 MiB)
          Interrupt:19 Base address:0x2024

eth1      Link encap:Ethernet  HWaddr 00:0C:29:F4:22:E7
          inet addr:192.168.28.137  Bcast:192.168.28.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef4:22e7/64 Scope:Link
```

Command-Line Network Interface Utilities

`ifup` and `ifdown` are:

Interface control scripts

Used to activate and deactivate network interfaces

`ethtool`

`ethtool` is used to query and set low-level network interface properties.

Changes made by `ethtool` do not persist after a reboot.

Address Resolution Protocol (ARP)

- ARP resolves an IP address to the MAC address.
- IP addresses and associated MAC addresses are cached in an ARP table.
- By default, entries are cached for 60 seconds.
- Use the `arp` command to display, add, or delete entries in the ARP cache.
  - For example, to display all entries:

```
# arp -n
```

- Alternatively, use the `ip neigh` command to modify the ARP cache.
  - For example, to delete all entries:

```
# ip neigh flush all
```

Network Interface Bonding

- Network interface *bonding*:
  - Combines multiple network connections into a single logical interface
  - Is used to increase throughput and provide redundancy
- Example of creating a bonding interface file:

```
/etc/sysconfig/network-scripts/ifcfg-bond0
DEVICE=bond0
```

- Physical interface files need MASTER and SLAVE directives:

```
MASTER=bond0
SLAVE=yes
```

- Load the bonding kernel module.
- You can also use the `ifenslave` command-line utility.

Virtual Local Area Networks

- A VLAN is a group of machines that can communicate as if they were attached to the same broadcast domain.
- With VLANs, network switches (not routers) create the broadcast domain.
- Switch ports are assigned to a VLAN ID, and all ports assigned to a single VLAN are in a single broadcast domain.

Virtual Local Area Networks

To create the `ifcfg-eth0.5` file for VLAN ID `5` on `eth0`:

```
DEVICE=eth0.5
VLAN=yes
```

Alternatively, use the `vconfig` command:

```
# vconfig add eth0 5
```

View the `/proc/net/vlan` directory to get detailed information about VLAN interfaces.

route **Utility**

- The `route` utility is used to display or manipulate the IP routing table.

- The default route, `GATEWAY`, is configured in the `/etc/sysconfig/network` file.

route **Utility**

To display the routing table:

route –n netstat –r

To add an entry to the routing table:

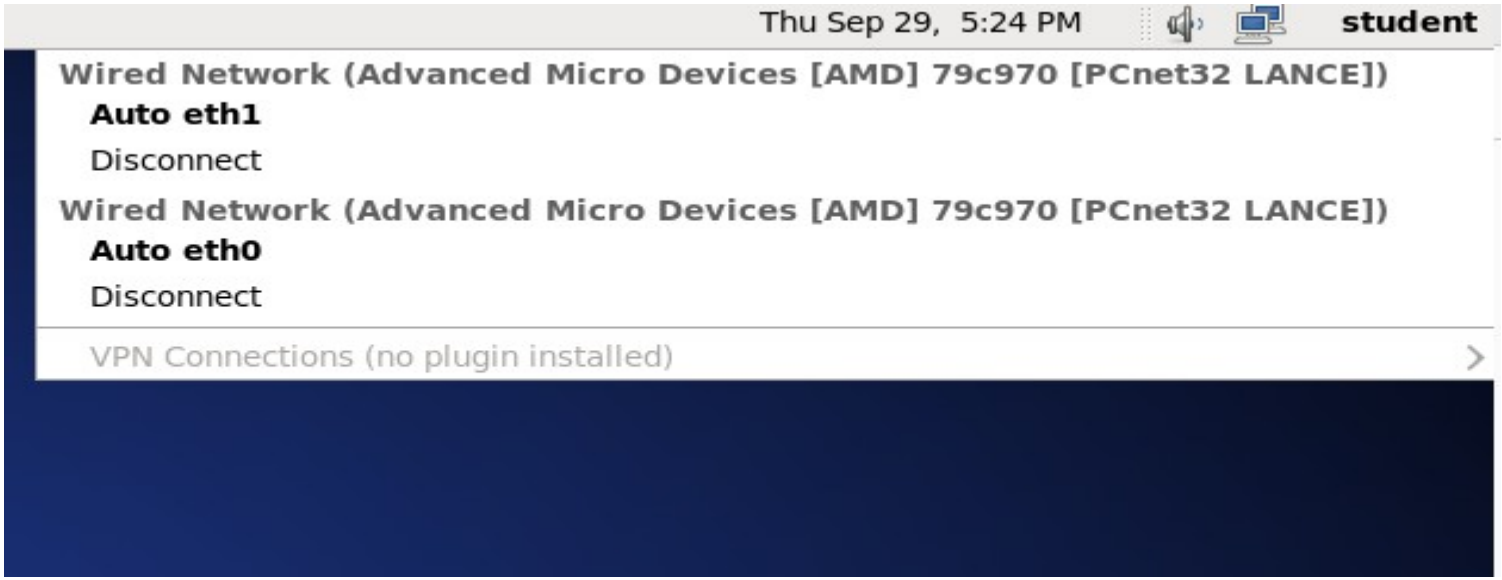
route add default gw 192.0.2.2

route add –net 192.18.21.0 netmask 255.255.255.0 eth0

Configure permanent static routes in the /etc/sysconfig/network-scripts/*route-interface* file.

Network Manager

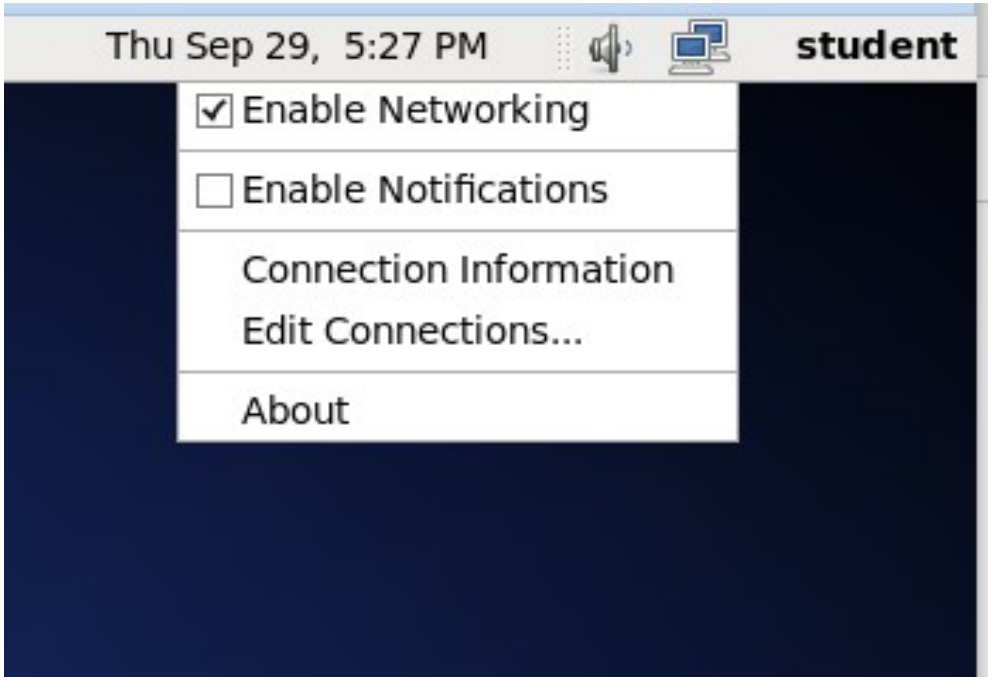
- NetworkManager:
  - Dynamically detects and configures network connections
  - Includes a GNOME Notification Area applet
- Click the icon to display the drop-down menu.



Network Manager

- Right-click the icon to display the drop-down menu.
- Select **Edit Connections** from the menu to display

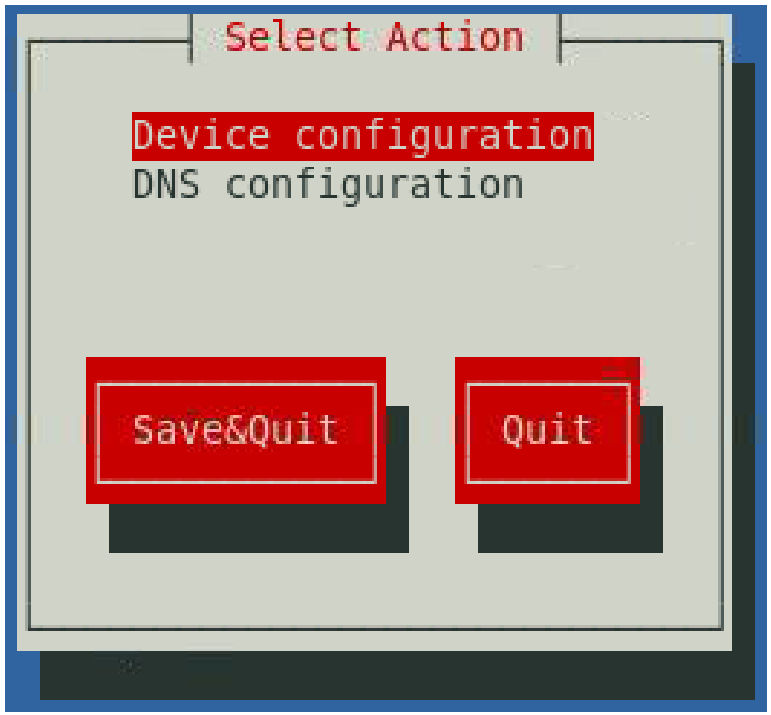
the Network Connections window.

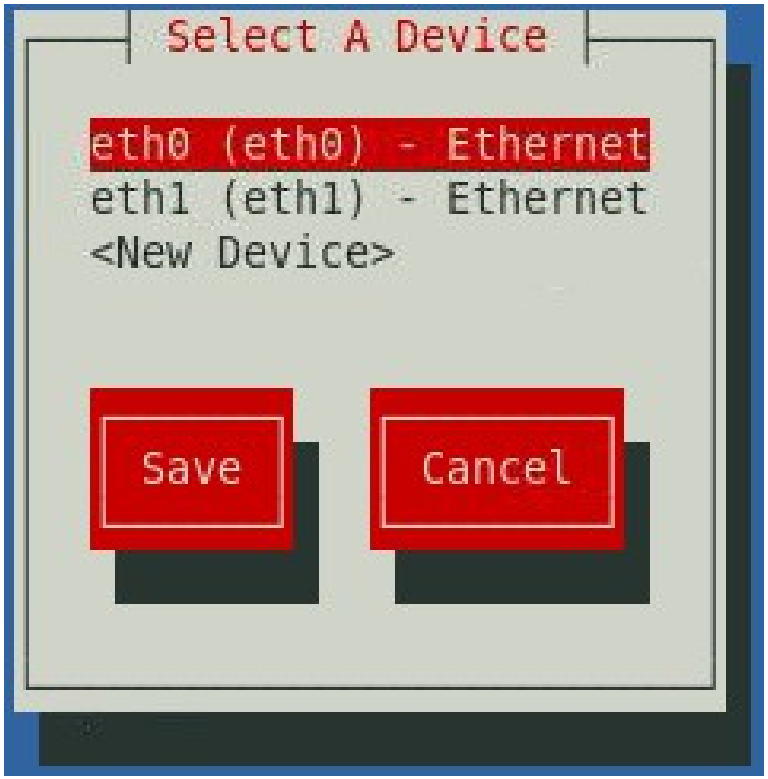


Network Connections Window

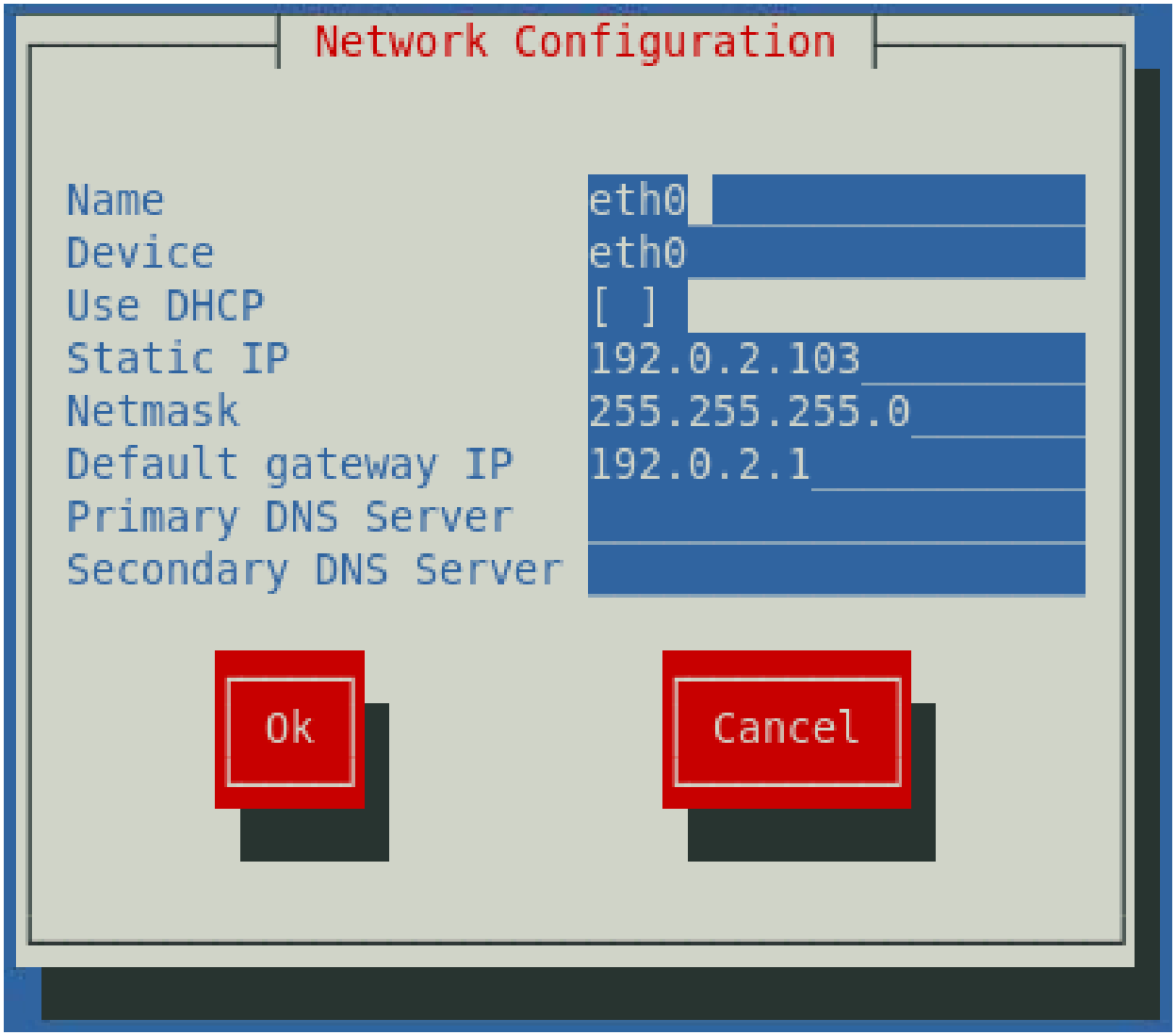
system-config-network **Utility**

system-config-network





Device Configuration



DNS Client Configuration