Department of Computer Science and Engineering

Linux for Devices

Module-3, Lecture-1

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Using the Shell

- The program used to interpret and manage commands is referred to as the shell
- Shell are independent of Linux distribution
- It provides a way to create executable script files, run programs, work with file systems, compile computer code, and manage the computer
- The Linux distribution you are using has more than one shell installed by default and available for your use.
- We will focuses primarily on the bash shell because the Linux distributions Fedora and Red Hat Enterprise Linux, both use the bash shell by default

- The shell is a command language interpreter.
- There are several ways to get to a shell interface in Linux.
- Three of the most common are the
  - ✓ Shell prompt
  - ✓ Terminal window
  - ✓ Virtual console

- Choosing Your Shell
- Running Commands
- Recalling Commands Using Command History
- Command-line completion
- Command-line recall
- Connecting and Expanding Commands
- Piping between commands

- Sequential commands
- Background commands
- Expanding commands
- Expanding arithmetic expressions
- Expanding variables

Linux for Devices

Module-3, Lecture-2

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Moving Around the Filesystem

- Linux Filesystems versus Windows-Based Filesystems
- Using Basic Filesystem Commands
- Using Metacharacters and Operators
- Using fi le-redirection metacharacters
- Using brace expansion characters
- Listing Files and Directories

- Identifying Directories
- Understanding File Permissions and Ownership
- Changing permissions with chmod (numbers)
- Changing permissions with chmod (letters)
- Setting default fi le permission with umask
- Changing file ownership
- Moving, Copying, and Removing Files

Linux for Devices

Module-3, Lecture-3

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Working with Text Files

- Exploring the Text Editors
- Editing Files with vim and vi
  - ✓ Adding text
  - ✓ Moving around in the text
  - ✓ Deleting, copying, and changing text
  - ✓ Pasting (putting) text
  - ✓ Repeating commands
  - ✓ Exiting vi

- Skipping around in the file
- Searching for text
- Using ex mode
- Learning more about vi and vim
- Finding Files
- Using locate to find files by name
- Searching for files with find
- Finding files by name

- Finding files by name
- Finding files by user
- Finding files by permission
- Finding files by permission
- Using not and or when finding files
- Finding files and executing commands
- Searching in files with grep

Linux for Devices

Module-3, Lecture-4

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Managing Running Processes

- Understanding Processes
- Listing Processes
- Listing processes with ps
- Listing and changing processes with top
- Listing processes with System Monitor
- Managing Background and Foreground Processes
- Starting background processes

- Using foreground and background commands
- Killing and Renicing Processes
- Using kill to signal processes by PID
- Using killall to signal processes by name
- Setting processor priority with nice and renice

Listing and changing processes with top

Department of Computer Science and Engineering

System uptime

Average system load

Running tasks

RAM

Swap space

```
top - 07:53:13 up 11:37,  2 users,  load average: 1.21, 1.17, 1.12
Tasks: 277 total,   1 running, 276 sleeping,   0 stopped,   0 zombie
Cpu(s): 11.0%us,  2.1%sy,  0.0%ni, 86.5%id,  0.4%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   3716196k total,  2159508k used,  1556688k free,   227916k buffers
Swap:  4194296k total,        0k used,  4194296k free,   784696k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 3472 cnegus    20   0 1053m 418m  29m D 27.5 11.5  20:40.89 firefox
 3665 cnegus    20   0  172m  62m  13m S 15.9  1.7 104:04.93 npviewer.bin
 2577 cnegus    20   0  174m  26m  13m S  4.3  0.7   3:42.17 Xorg
```

Allocated memory

Memory actually in use

Sort by CPU use

Sort by memory use

Listing processes with System Monitor

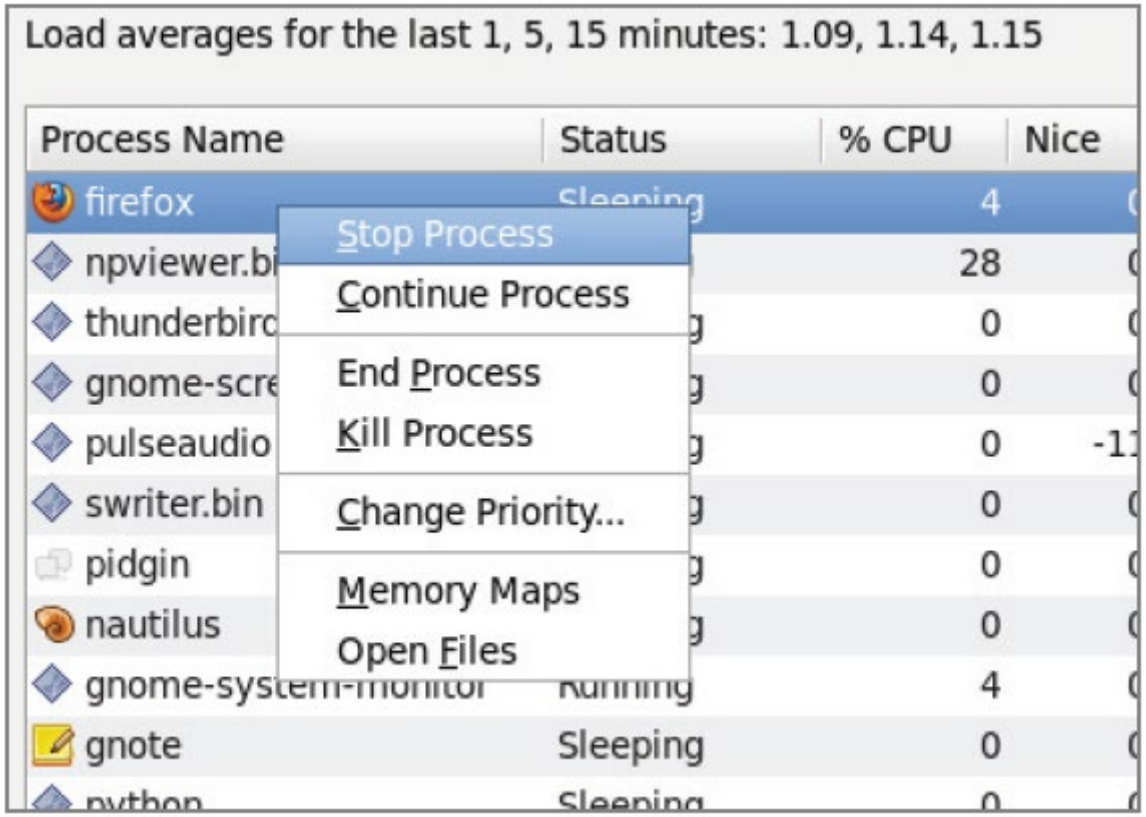
Monitor Edit View Help

System | Processes | Resources | File Systems

Load averages for the last 1, 5, 15 minutes: 1.33, 1.24, 1.21

Process Name	Status	% CPU	Nice	ID	Memory	Waiting Channel	Session
bash	Sleeping	0	0	10949	2.8 MiB	n_tty_read	
bash	Sleeping	0	0	5467	1.8 MiB	do_wait	
bluetooth-applet	Sleeping	0	0	2933	661.0 KiB	poll_schedule_timeout	
bonobo-activation-server	Sleeping	0	0	2919	276.0 KiB	poll_schedule_timeout	
clock-applet	Sleeping	0	0	3184	2.1 MiB	poll_schedule_timeout	
dbus-daemon	Sleeping	0	0	2756	900.0 KiB	poll_schedule_timeout	
dbus-launch	Sleeping	0	0	2755	128.0 KiB	poll_schedule_timeout	
diskwarn	Sleeping	0	0	2811	2.1 MiB	poll_schedule_timeout	
evolution-data-server-2.28	Sleeping	0	0	6373	1.1 MiB	poll_schedule_timeout	
firefox	Sleeping	12	0	12005	443.0 MiB	poll_schedule_timeout	
gconfd-2	Sleeping	0	0	2836	828.0 KiB	poll_schedule_timeout	
gconf-helper	Sleeping	0	0	2915	1.9 MiB	poll_schedule_timeout	
gconf-im-settings-daemon	Sleeping	0	0	3127	80.0 KiB	poll_schedule_timeout	

End Process


Load averages for the last 1, 5, 15 minutes: 1.09, 1.14, 1.15
Process Name
Status
% CPU
Nice
firefox
4
npviewer.bi
28
thunderbird
0
gnome-scre
0
pulseaudio
0
swriter.bin
0
pidgin
0
nautilus
0
gnome-system-monitor
4
gnote
Sleeping
0
Stop Process
Continue Process
End Process
Kill Process
Change Priority...
Memory Maps
Open Files

Signal	Number	Description
SIGHUP	1	Hangup detected on controlling terminal or death of controlling process.
SIGINT	2	Interrupt from keyboard.
SIGQUIT	3	Quit from keyboard.
SIGABRT	6	Abort signal from abort(3).
SIGKILL	9	Kill signal.
SIGTERM	15	Termination signal.
SIGCONT	19,18,25	Continue if stopped.
SIGSTOP	17,19,23	Stop process.

Linux for Devices

Module-3, Lecture-5

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

- Understanding Shell Scripts
  - ✓ Executing and debugging shell scripts
  - ✓ Understanding shell variables
  - ✓ Special shell positional parameters
  - ✓ Reading in parameters
  - ✓ Parameter expansion in bash
  - ✓ Performing arithmetic in shell scripts
  - ✓ Using programming constructs in shell scripts

Department of Computer Science and Engineering

Linux for Devices

Module-3, Lecture-6

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Linux Files

- The File Structure
- Home Directories
- Pathnames: Absolute and Relative
- System Directories
- Listing, Displaying, and Printing Files
  - Listing Files: ls
  - Displaying Files: cat, less, and more
  - Printing Files: lpr, lpq, and lprm

Department of Computer Science and Engineering

Linux for Devices

Module-3, Lecture-7

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Linux Directories

- Managing Directories:
  - ✓ Creating and Deleting Directories
  - ✓ Displaying Directory Contents
  - ✓ Moving Through Directories
  - ✓ Referencing the Parent Directory
- File and Directory Operations:
  - ✓ Searching Directories
  - ✓ Searching the Working Directory

- ✓ Locating Directories
- ✓ Copying Files
- ✓ Moving Files
- ✓ Copying and Moving Directories
- ✓ Erasing Files and Directories
- ✓ Links

- The mtools Utilities

Command	Execution
mkdir *directory*	Creates a directory.
rmdir *directory*	Erases a directory.
ls -F	Lists directory name with a preceding slash.
ls -R	Lists working directory as well as all subdirectories.
cd *directory name*	Changes to the specified directory, making it the working directory. **cd** without a directory name changes back to the home directory: **$ cd reports**
pwd	Displays the pathname of the working directory.
directory name/filename	A slash is used in pathnames to separate each directory name. In the case of pathnames for files, a slash separates the preceding directory names from the filename.

`..`	References the parent directory. You can use it as an argument or as part of a pathname:
	`$ cd ..`
	`$ mv ../larisa oldletters`

`.`	References the working directory. You can use it as an argument or as part of a pathname: `$ ls .`
~/*pathname*	The tilde is a special character that represents the pathname for the home directory. It is useful when you need to use an absolute pathname for a file or directory: `$ cp monday ~/today`

Linux for Devices

Module-3, Lecture-8

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Linux Archives

- Archiving and Compressing Files:
  - ✓ Archiving and Compressing Files with File Roller
  - ✓ Archive Files and Devices: tar
  - ✓ Displaying Archive Contents
  - ✓ Creating Archives
  - ✓ Extracting Archives
  - ✓ Updating Archives
  - ✓ Archiving to Floppies
  - ✓ Compressing Archives
  - ✓ Archiving to Tape

- File Compression:
  - ✓ Compression with gzip
  - ✓ Compressing with bzip2
  - ✓ Using Zip

Commands	Execution
tar *options files*	Backs up files to tape, device, or archive file.
tar *optionsf archive_name filelist*	Backs up files to a specific file or device specified as *archive_name. filelist*; can be filenames or directories.
Options	
c	Creates a new archive.
t	Lists the names of files in an archive.
r	Appends files to an archive.
U	Updates an archive with new and changed files; adds only those files modified since they were archived or files not already present in the archive.
--delete	Removes a file from the archive.
w	Waits for a confirmation from the user before archiving each file; enables you to update an archive selectively.
x	Extracts files from an archive.

m	When extracting a file from an archive, no new timestamp is assigned.
M	Creates a multiple-volume archive that may be stored on several floppy disks.
f *archive-name*	Saves the tape archive to the file archive name, instead of to the default tape device. When given an archive name, the **f** option saves the tar archive in a file of that name.
f *device-name*	Saves a tar archive to a device such as a floppy disk or tape. **/dev/fd0** is the device name for your floppy disk; the default device is held in **/etc/default/tar-file**.
v	Displays each filename as it is archived.
z	Compresses or decompresses archived files using gzip.
j	Compresses or decompresses archived files using bzip2.

Option	Execution
`-c`	Sends compressed version of file to standard output; each file listed is separately compressed: `gzip -c mydata preface > myfiles.gz`

`-d`	Decompresses a compressed file; or you can use gunzip:
	`gzip -d myfiles.gz`
	`gunzip myfiles.gz`

`-h`	Displays help listing.
`-l` *file-list*	Displays compressed and uncompressed size of each file listed: `gzip -l myfiles.gz`

-r *directory-name*	Recursively searches for specified directories and compresses all the files in them; the search begins from the current working directory. When used with **gunzip**, compressed files of a specified directory are uncompressed.
-v *file-list*	For each compressed or decompressed file, displays its name and the percentage of its reduction in size.
-*num*	Determines the speed and size of the compression; the range is from –1 to –9. A lower number gives greater speed but less compression, resulting in a larger file that compresses and decompresses quickly. Thus –1 gives the quickest compression but with the largest size; –9 results in a very small file that takes longer to compress and decompress. The default is –6.