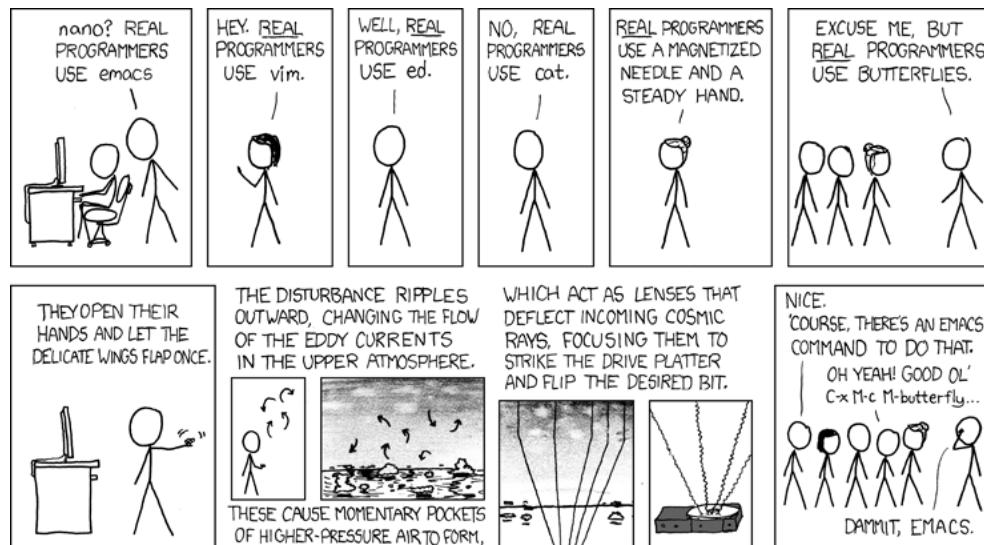


Chapter 7 - Linux Utilities

Monday, February 23, 2015 12:09 AM



Objectives

- At the end of this lecture you will be able to
 - Use basic utilities to list files & display text files
 - Copy, move, and remove files
 - Search, sort, print, categorize, and compare text files
 - Compress and decompress archive files
 - Count lines, letters, and words in a file
 - Locate utilities on the system
 - Change datetime on a file.
 - Edit text files
- Edit text files** → Money → Next week

- Since Linux/Unix was created in a time without a GUI
every thing is done in **text**, via the Command Line/shell.

Basic Utilities

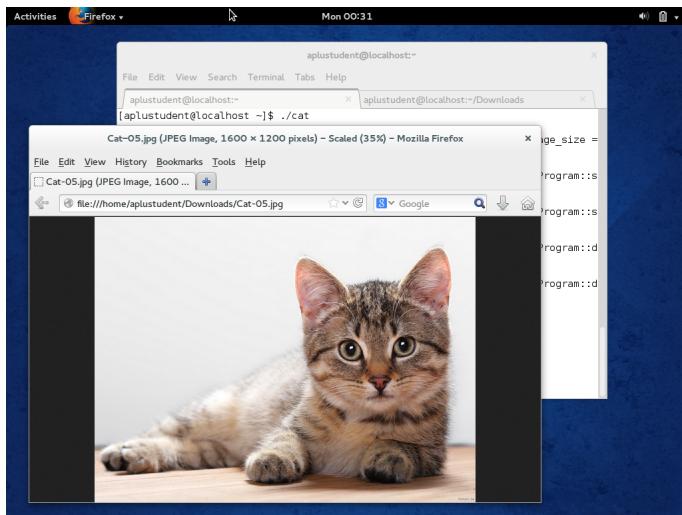
Cat → what does this command do?

... I only wish it did this...





but maybe I can make Linux do this....



Why can my system
do this and yours
cannot?

Cat displays content of file forward
Tac displays file contents backward
Where would Tac be useful?

Date command

↳ Prints the date, time, day, month, year, and time zone

- ↳ series of values you can use to adjust output
- ↳ try it date + "%A %B %d" ↳ spaces optional
↑ no space
- ↳ -d is datestring
- ↳ -u is utc
- ↳ **timedatectl** is a system control tool which can be used to set/adjust the system clock

~~↳ Also a command line cal tool to view calendar data~~

Echo

- ↳ Displays content back to the screen
- ↳ Good for displaying contents of variables back to the screen.
 - ↳ echo \$PATH
 - ↳ echo \$SHELL

Hostname

↳ **hostname**

↳ will print the "name" of "your host" or system name.

↳ -f prints FQDN

↳ there is now a **hostnamectl** tool (part of systemd)

↳ Try it!

Display a text file one screen at a time.

- ↳ Each terminal starts out 80 by 24 (code to original terminals)
- ↳ Cat dumps the entire file ^{out} to (std out)
 - ↳ What if the file is multiple 1000's of lines?
- ↳ First the **more** command was created

- ↳ Allows you to advance line by line (space bar or enter)
- ↳ Allows you to jump one (80 by 20 page) at a time
- ↳ Doesn't let you scroll back...
- ↳ Enter **less**
 - ↳ Because less is more...
 - ↳ Less can do everything that **more** can but it can scroll backwards (page back too!)
 - ↳ Also a **most** command
 - ↳ We can skip it for now
 - ↳ You can install it and run it if you want.

Listing files

- ↳ **ls** command
 - ↳ we are very familiar with this
 - ↳ remember you can use shell metacharacters with this

Remove Files (Delete)

- ↳ **rm** command
 - ↳ -r recursive
 - ↳ -f force
 - ↳ -i interactive (Yes/no prompt)
 - ↳ -v verbose (usually everywhere)
 - ↳ shred command overwrites a files memory space
(making sure it is deleted)

Copy Files
↳ **cp** command

- a attempt to preserve ownership, permissions, and timestamps
 - b backup makes a backup copy of a file that would be overwritten
 - p preserve file attributes in copy
 - i interactive
 - v verbose
-

Select characters or Fields from Input Lines

↳ cut

- ↳ selects characters and prints to standard out
- ↳ example how to see just file permissions?

↳ ls -l

- ↳ try this ls -l | cut -c 2-10
↳ what does it do? why? ↑ no space
- ↳ -d is a delimiter
 - ↳ good way to cut on a space or ":" ↑ no space
 - ↳ try this: cat /etc/passwd
 - ↳ try this: cat /etc/passwd | cut -d ":" -f 5

Try this:

```
ls -l | tr -s ' ' | cut -f 5,9 -d ' '
```

what does it output?

Display difference between two files

- ↳ diff → show what "kind" of file a file is
 - ↳ file
 - ↳ text / Ascii
 - ↳ Binary
 - ↳ ELF
 - ↳ Directory
 - ↳ zip
- en.wikipedia.org/wiki/Executable_and_Linkable_Format

point to a printer

- ↳ will print to a printer (not stdout)
 - ↳ lpr
 - ↳ use cat
 - ↳ lpr processes.txt
 - ↳ lpq is the print queue
- ↳ lprm remove a job from the queue

Find Files based on criteria

- ↳ find command
- Does not take options
- Takes "expressions"
- Try it `find . -atime +10`
 - ↑ ↑ ↑
 - Command location what to look for
 - range
 - time interval

(recursing down the tree)

↳ Try it: find /etc -name "*/*.xml"

↳ -links (file that has softlinks)
 -mtime (any file modified in x time)

↳ -size (look for files of x size)

↳ options can be strung together to make a specific expression range.

↳ -exec command allows you to "pipe" the output of another command as (stdin)

↳ -f type finds a file of type

-d directory

-f file

↳ -user finds all files owned by a user name

Search for a pattern in files

↳ grep

↳ grep utility allows you to search one or more files for a pattern

↳ from a string expression
 a regular expression

↳ grep can be used as a direct utility

↳ Or as part of a "pipeline"

↳ grep send lines that match your pattern to (stdout)

↳ -c adds a count function to return the number of matches

↳ --color highlights the match

↳ -i ignore-case (remember Linux case matters)

↳ -l (el) displays only the filename(s) that contains your pattern

search

-r recursive

-v invert search

Display beginning of a file

head

→ displays the first 10 lines of a file

tail → displays the last 10 lines of a file

↳ both a -n modify to many lines.

how

Sort and Merge Files

↳ Sort

↳ takes (stdin) and sorts it and prints to (stdout)

Given

↳ two inputs sort will also merge contents to

(stdout)

↳ -c check to see if things are already sorted

↳ sends merged sort to an output file

File Modification and Access Times

change

touch

↳ a file but also update a timestamp

↳ create

Compression and Archiving

↳ `dtrx` (needs to be installed) (safest)

↳ `gzip` → (stream based) oldest GNU project open source version of `zip`/`unzip`

↳ `bzip2` → (block based) smaller compression

Try it: ↳ `xz` Try it:

↳ `cd ~` -d means decompress

↳ `ls -l`

add a -l-q value

↳ `ls -fontent` extreme compression, slow time but so = q file

↳ `xz` fast compression, larger file small size

Compress tool (standard on macs)

↳ `bzip2` like

acts → using -d for decompress

↳ `gzip` / `gunzip` simply takes an argument

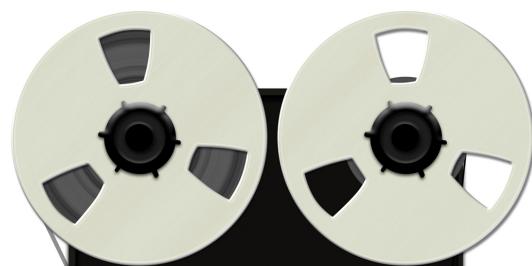
↳ all utilities need to be told to recursively compress.

File

and Directory Archives

↳ `tar`

↳ Tape Archive = tar



↳ Allows you to place a structure into a directory
single file (like an iso file)



↳ Archive retains file structure and can be easily compressed

↳ tar is used in combo traditionally with gzip

↳ called a tar ball

will see the file extension
or *.tar.gz
or *.tgz

↳ to extract it tar -xvf (this unzips as well as unarchives
z in gzip
(letter changed to match compression choice))

Summary

Utility	Function
<code>cat</code>	Joins and displays files (page 216)
<code>cp</code>	Copies files (page 224)
<code>cut</code>	Selects characters or fields from input lines (page 225)
<code>diff</code>	Displays the differences between two text files (page 227)
<code>dos2unix</code>	Converts Windows files to Linux format (page 260)
<code>file</code>	Discusses the classification of a file (page 229)

Utility	Function
<code>date</code>	Displays or sets the system time and date (page 218)
<code>free</code>	Displays memory usage information (page 253)
<code>hostname</code>	Displays the system name (page 219)
<code>uptime</code>	Displays system load and duration information (page 253)
<code>w</code>	Lists users on the system (page 254)
<code>who</code>	Lists users on the system (page 254)

Utility	Function
<code>bunzip2</code>	Decompresses a file compressed using <code>bzip2</code> (page 248)
<code>bzcat</code>	Displays a file compressed using <code>bzip2</code> (page 248)
<code>bzip2</code>	Compresses or decompresses files (page 245)
<code>compress</code>	Compresses a file (page 249)
<code>dtr</code>	Extracts and decompresses files intelligently (page 245)
<code>gunzip</code>	Decompresses a file compressed using <code>gzip</code> or <code>compress</code> (page 248)

Utility	Function
<code>echo</code>	Displays arguments (page 219)
<code>locate</code>	Searches for a file (page 256)
<code>nano</code>	Edits a text file (page 270)
<code>script</code>	Records a shell session (page 257)
<code>vi/vim</code>	Edits a text file (page 262)
<code>whereis</code>	Locates a utility (page 255)
<code>which</code>	Locates a utility (page 255)
<code>xargs</code>	Converts standard input to command lines (page 260)