COSC 3750

Filesystem and Utilities

Kim Buckner

University of Wyoming

Jan 25, 2021

Filesystem

- Every operating system has its preferred filesystem
- Under Linux and UNIX there are a large number BUT they are all viewed the same way.
- The filesystem consists of files and directories.
- This is not Windows and there are no "folders".

(more . . .)

- Directories are hierarchical and can be viewed as a tree structure.
- The individual file systems begin at the root directory which is named “/”.
- There are two ways to reference files or directories
  1. Relative reference.
  2. Fully-qualified (full path) reference.

(more . . .)

- Every fully-qualified filename is based off the root. EX: `/home/faculty/kbuckner/test.txt`
- Relative references are from the directory that the shell is currently displaying, the “current working directory” ($cwd). EX: `fall2017/slides/aug25.pdf`

(more . . .)

- Some (most?) shells support the use of the tilde as a special character meaning "the current user's home directory". EX:
  `/home/faculty/kbuckner/test.txt`
  is equivalent to
  `~/test.txt`

(more . . .)

- Leaving out the slash after the tilde causes the shell to think you mean the home directory of some **other** user. EX:

```
$> ls ~Courses
Unknown user: Courses.
$>
```

Security

- Windows is based on DOS which was a one-owner operating system.
- Initially there was NO protection provided for any DOS filesystem components other than hiding names.
- UNIX variants were designed as multi-user systems and as such have implemented filesystem security in multiple ways.
- The basic method is through ownership

Ownership

- All components of the filesystem have metadata attributes.
- Can you tell me what “metadata ” means?

Ownership

- All components of the filesystem have metadata attributes.
- Can you tell me what “metadata ” means?
- Data about data.
- In general, this consists of the owner of the component, the component’s permissions, and its type, such as directory, block device, regular file, etc.

(more . . .)

- The owner of a component can set *permissions* for the component. These can restrict access to the only the owner, make it executable, allow access to some users or all users, allow execution by some or all users, and more.
- This ownership information can be displayed using `/bin/ls` and modified with `/bin/chmod`.

Special names

- There are two special names used on Linux
- These are "." and ".."
- They allow some simplified navigation and reference.
- Dot (.) is another name for the current directory.
- DotDot (..)is a reference to the parent directory.

More Security.

- By default your accounts are set so that directories and files allow group access.
- Means all undergraduates can read ALL other undergraduate files/directories.
- I STRONGLY suggest that you change the permissions on any directories you do NOT want someone else to read to 0700.

```
$> chmod 0700 dname
```

(more . . .)

- That means no one but you can list the contents of or change directory into any such directory.
- To keep from having to do this on a regular basis you can change your **umask**.
- If using tcsh/csh add the line “umask 077” to your .cshrc file.
- If using bash/sh add the line to the .bashrc file.

(more . . .)

- The next time you open a terminal this should take effect IN THAT TERMINAL.
- If a program (say firefox) creates a file, that change will not take effect until you log out and back in.

(more . . .)

- Just like entering a stranger's house without permisssion, reading someone else's files without permission is a **bad thing**.
- Do NOT do it.
- Remember, in civilized society, "can" does not mean "should."

Utilities

Utilities

I categorize these into four basic groups

1. Process control - some shell functions, some actual executables
2. File control and management - Same
3. Text processing
4. Miscellaneous

Man and info

- First we will discuss two miscellaneous utilities.
- *man* displays “manual pages.”
- Was initially an electronic copy of the printed manuals from a UNIX distribution.
- Now, only the electronic version exits.
- Use the system version (distribution version) not some web page version.

(more . . .)

- *info* is a relatively (20+ years) new version of electronic documentation.
- Sometimes there is more information in the info document than in the man document. And sometimes not.
- This seems to the preferred method for GNU documents.
- That does not mean that everything is in *man* or *info*.

Process control

Process control and monitoring

- There are a number of utilities that are used for this.
- I am going to cover the one's I use regularly.
- Note that as with all Linux, every time you **read** about something, you discover some new method or utility.
- That means read the *man* and/or *info* docs for these.

kill

- Normally just used to terminate a process but really sends the process a **signal**.
- Can only signal processes you control.
- Control comes in several flavors, for now we will consider that it means a process you "own".
- If no signal is specified, the SIGTERM (15) signal is sent.

(more . . .)

- This signal can be ignored by the process.
- May require that you use the SIGKILL (9) signal as in

  kill -9 22345.
- Signals are listed in some signal.h, exactly which one is subject to installation.
- *man signal.h* and *man 7 signal* may be useful.

(more . . .)

- Try doing *more /usr/include/asm/signal.h*
- *kill* may be (probably is) implemented by your shell.
- To get the system version use `/bin/kill`.

ps

- List processes and provide some status about them.
- This is not limited to processes you own.
- Some types of the information available:
  - running time
  - percent of CPU usage
  - current memory size
  - ownership information

(more . . .)

- I have a couple of version of this I use
    - ps -U 8036 -o pid,args
    - ps -Ao pid,cmd,pcpu,pmem,psr,user,time
    - ps -U 8036 -o user,pid,vsz,pmem,pcpu,s,time,comm
- Can find your user id (uid) with the *id* utility.

pgrep – process grep

- This lists processes that match some regular expression
- It can list processes owned by anyone but defaults to the current user.
- Only provides PID and optionally the name
- *pkill* is like pgrep only it sends the specified signal to the selected process(es).

top

- See the *top n* (usually 20) processes.
- Very handy when your machine does not seem to be performing well.
- Especially if you are testing threaded processes or multi-process operations.

(more . . .)

- Can also try *uptime*
- And *w* will tell you who is logged in.

Text Utilities

wc – word count

- This can be handy when you are creating test files or when doing homework.
- Basic operation prints the number of newlines, number of words and number of characters on the input.
- If there are multiple files, then it will also print a total over all files.
- Can have it print just newlines, just words, or just characters.

head

- Prints the **first** lines of its input, default is usually 10
- Number of lines can be changed.
- *head -20* `filename`

tail

- Like head but prints the **last** 10 lines of the file.
- Number of lines can be changed.
- *tail -20* `filename`
- And the 43rd is

```
head -43 x | tail -1
```

split

- Breaks the input into equal size files (last can be smaller)
- You provide a prefix and it adds “aa” to the first, “ab” to the second and so on.
- Default prefix is “x”.
- Can specify max size of files, defaults to 1000 lines.

sort

- Used to sort text files, line by line.
- Default is operate on entire line.
- Can specify fields, columns, field separator, sort order, etc.
- Can be very handy when processing lists of names.

uniq

- Use on input to delete duplicate **adjacent** lines.
- Input does not have to be sorted, but only works on adjacent lines.
- Can use *sort -u* instead.

cut

- Operates on fields
- Returns some subset of the fields on each line of input.
- Can specify fields, field delimiters, more.
- Really useful only if the input is in some fixed format.

tr – translate characters

- This is very useful to change all uppercase letters to lower or vice versa.
- To modify/delete special characters like tabs.
- Give it two sets, characters to match and replacements.
- Always reads from standard input and writes to standard output.