

Week 02W: Review & Shell Scripting

Thuy Vu

- Office Hours

- Thuy Vu — 09–10a Tuesdays & Thursdays, BH2432

- Assignment 2

- Shell Scripting
 - web.cs.ucla.edu/classes/winter17/cs35L/assign/assign2.html
 - Time due 23:55 next Saturday, January 21

- Spell-checking Hawaiian
 - <http://mauimapp.com/moolelo/hwnwdseng.htm>
 - sed, cat, tr, grep, sort, comm
- `echo -e "abcdef\nxyz\nmno" | sed "/a/,/y/d"`
- `comm` - compare two **sorted** files line by line

- Find duplicate files `same1n D` \longrightarrow all duplicated regular files to hard links.
 - should keep file name starting with `.` or first-lexicographically-ordered (`.Y`, `.X`, `X`, `A`, `B`)
 - regular files, ignore symbolic links and directories
 - names contain space, asterisk, leading `"-`", return character, ...

- ❶ `^(Caret)` = match expression at the start of a line, as in `^A`.
- ❷ `$(Question)` = match expression at the end of a line, as in `A$`.
- ❸ `\ (Back Slash)=` turn off the special meaning of the next character, as in `\^`.
- ❹ `[] (Brackets)=` match any one of the enclosed characters, as in `[aeiou]`.
Use Hyphen `-` for a range, as in `[0-9]`.
- ❺ `[^]` = match any one character except those enclosed in `[]`, as in `[^0-9]`.
- ❻ `.(Period)` = match a single character of any value, except end of line.
- ❼ `*(Asterisk)` = match zero or more of the preceding character or expression.
- ❽ `\{x,y\}` = match x to y occurrences of the preceding.
- ❾ `\{x\}` = match exactly x occurrences of the preceding.
- ❿ `\{x,\}` = match x or more occurrences of the preceding.

- ❶ `grep smug files` — search files for lines with 'smug'
- ❷ `grep '^smug' files` — 'smug' at the start of a line
- ❸ `grep 'smug$' files` — 'smug' at the end of a line
- ❹ `grep '^smug$' files` — lines are 'smug'
- ❺ `grep '\^s' files` — lines containing '^s'
- ❻ `grep '[Ss]mug' files` — find 'Smug' or 'smug'
- ❼ `grep 'B[o0][bB]' files` — BOB, Bob, BOb or BoB
- ❽ `grep '^$' files` — search for blank lines
- ❾ `grep '[0-9][0-9]' files` — find pairs of digits
- ❿ `grep '^From: ' /usr/mail/$USER` — list your mail
- ❶❶ `grep '[a-zA-Z]'` — any line with at least one letter
- ❶❷ `grep '[^a-zA-Z0-9]` — any but letter or number
- ❶❸ `grep '[0-9]\{3\}-[0-9]\{4\}'` — 999-9999
- ❶❹ `grep '^.$'` — lines with exactly one character
- ❶❺ `grep '"smug"'` — 'smug' within double quotes
- ❶❻ `grep '"*smug"'` — 'smug', with or without quotes
- ❶❼ `grep '^\..'` — any line that starts with a "."
- ❶❿ `grep '^\. [a-z][a-z]'` — line of "." and 2 letters

1 `-e '1d' /etc/services`

2 `-e '1,10d' /etc/services`

3 `-e '/^#/d' /etc/services`

4 `-e '/regexp/d' /path/to/my/test/file`

5 `-n -e '/regexp/p' /path/to/my/test/file`

6 `-n -e '/BEGIN/,/END/p' /my/test/file`

7 `-n -e '/main[[:space:]]*{/,\/^}/p' sourcefile.c`

8 `-e 's/foo/bar/' myfile.txt`

9 `-e 's/foo/bar/g' myfile.txt`

10 `-e 's:/usr/local:/usr:g' mylist.txt`

11 `-e 's/<.*>//g' myfile.html`

12 `-e 's/<[^>]*>//g' myfile.html`

13 `-e 's/.*ralph said: &/' origmsg.txt`

14 `-n -e 's/p/' myfile.txt`

15 `-n -e 's/' -e 'p' myfile.txt`

16 `-n -f mycommands.sed myfile.txt`

17 `-e 's/$/\r/' myunix.txt > mydos.txt`

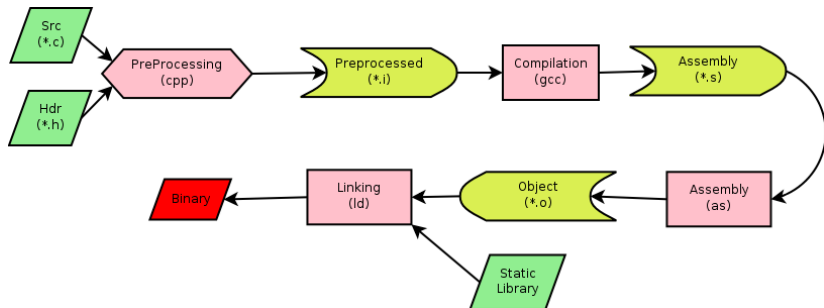
18 `-e 's/.$//' mydos.txt > myunix.txt`

19 `-e '1!G;h;$!d' forward.txt > backward.txt`

[:alnum:], [:alpha:], [:blank:], [:cntrl:], [:digit:], [:graph:], [:lower:], [:print:],
[:punct:], [:space:], [:upper:], [:xdigit:]

- http://www.funtoo.org/Sed_by_Example,_Part_1
- http://www.funtoo.org/Sed_by_Example,_Part_2
- http://www.funtoo.org/Sed_by_Example,_Part_3

Compilation Process



<https://bharathisubramanian.wordpress.com/2010/03/28/gcc-compilation-process/>

what if we make change on source files?

- on *.cpp file — just recompiles the changed one
- on *.h file — recompiles all those that include it

`make` process

- `configure` — check dependencies
- `make` — compile and make executables (require Makefile)
- `make install` — deploy (copy) executables to folders

“**ls** in Coreutils 7.6 is buggy, let's fix it!”

- 1 `wget ftp://ftp.gnu.org/gnu/coreutils/coreutils-7.6.tar.gz`
- 2 `tar -zxvf coreutils-7.6.tar.gz`
- 3 `mkdir CoreutilsInstall`
- 4 config using `configure`
- 5 compile
- 6 install
- 7 create `newpatch` file (from the link)
- 8 `patch` the `newpatch`
- 9 re-`make` it (don't `make install`, why?)

- Python, Anaconda
- Find duplicate files