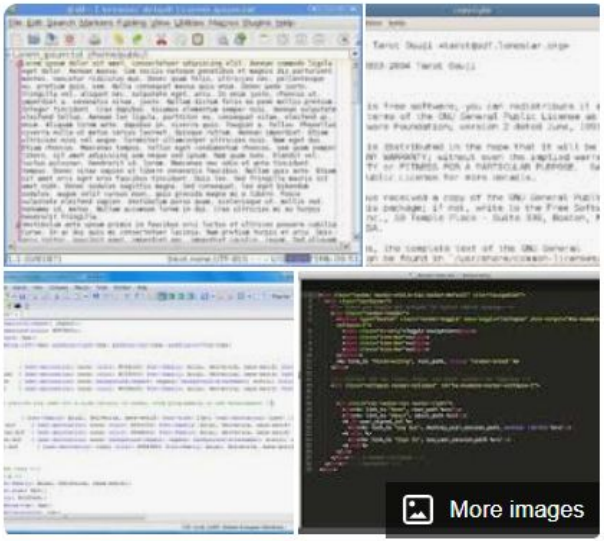
CSE 374 Lecture 5

Scripting Continued

Text Editors



A text editor is a type of computer program that edits plain text. Such programs are sometimes known as "notepad" software, following the naming of Microsoft Notepad. Wikipedia

Vi

- Move around, mark edit using letter keys
- Get to menu by typing “:”
- Save and quit “:wq”, no-save and quit: “:q!”

Emacs

- Endlessly extendable; can use arrows to move
- Menu commands use C (control) or M (meta/alt)
- Quit: C-x C-c
- Tutorial, C-h for help

Some useful utilities

Use `man -k`: find commands with subject search

Use `find`: location a file on a computer (`locate`: locate a file in the directory database)
`whereis`: finds files with a program's name, `which`: where the executable in your path is found

Use `!?phrase`: execute the last command containing phrase

Use `^typo^correct`: correct a typo in the last command

Use `diff f1 f2`: find lines that are different in f2 than in f1 (or `sdiff`)

Process Management

Figure out what's running:

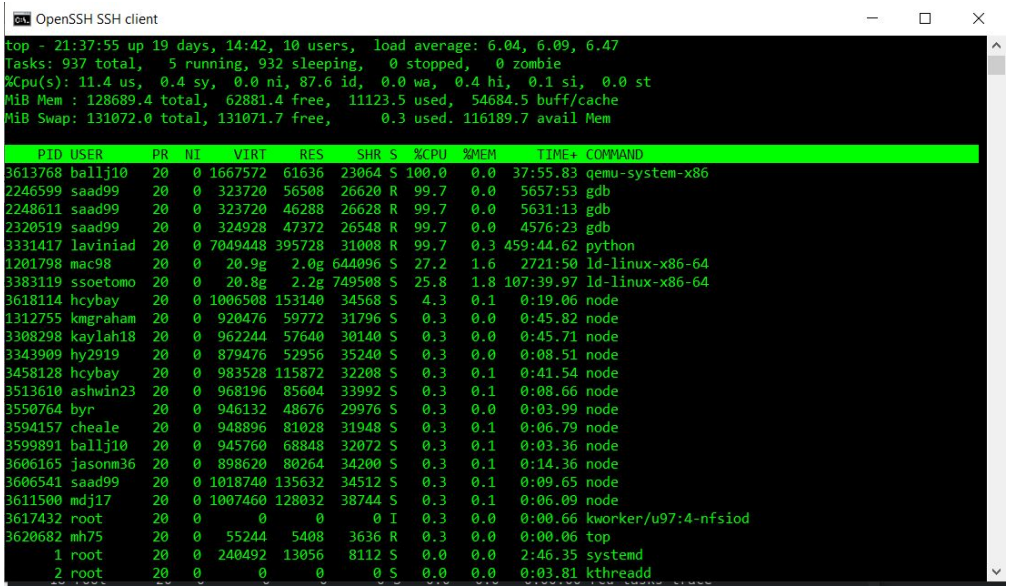
- Top
- ps (many options)

Stop processes:

- Ctrl-c (Send kill command)
- Kill (with options) PID

Manage processes:

- Ctrl-z (suspend process) / fg
- nice



Variables useful in a script

$# stores number of parameters (strings) entered

$0 first string entered - the command name

$N returns the Nth argument

$? Returns state of last exit

$* returns all the arguments

$@ returns a space separated string with each argument

(* returns one string with spaces, @ returns an array of words)

Arithmetic

Variables hold strings, so we need a way to tell the shell to evaluate them numerically:

K=$i+$j does not add the numbers

Use the shell function ((

k=$(($i+$j))

Or let k=”$i+$j”

The shell will automatically convert the strings to the numbers

Conditionals

Binary operators: -eq -ne -lt -le -gt -ge

Can use the [[shell command to use < , > , ==

Syntax is a little different, but commands works as expected

if *test*; then
    commands
fi

while *test*; do
    commands
done

for *variable* in *words*; do
    commands
done

Flow control

test *expression* or [*expression*]

```
if [ -f .bash_profile ]; then
    echo "You have a .bash_profile.
Things are fine."
else
    echo "Yikes! You have no
.bash_profile!"
fi
```

http://linuxcommand.org/lc3_man_pages/testh.html

Shell-scripting Notes

Bash Scripting	Java Programming
Interpreted	Compiled
Esoteric variable access	Highly structured, Strongly typed
Everything is a string	Strings have library processing
Easy access to files and program	Data structures and libraries
Good for quick & interactive programs	Good for large complex programs

Scripting Style Guide

Scripts should generally be <200 lines

Do one thing and do it well.

Always use spaces, not tabs (indent line with two spaces)

Comment code with ‘#’

https://google.github.io/styleguide/shell.xml

Practice ...

Up next: Regular expressions

Regular expressions: string of symbols and characters used for pattern matching