BILL LABOON

FIAT VOLUNTAS TUA: THE POWER OF THE COMMAND LINE

WHY TO LEARN COMMAND LINE

- GUIs are fine and all
- But not very fine-grained or flexible
- Picture books are great for kids, but can you imagine Shakespeare or Tolstoy with only pictures?

THE UNIX PHILOSOPHY

- “Do one thing, and do it well”
- Everything communicates via plaintext (input and output)
- Small programs can be combined to do incredibly powerful things!

REAL-LIFE EXAMPLE

- I wanted to compile all of the Java files in a directory *except* any of the test files OR files that start with Binary
- Probably not a simple way to do that in an IDE!

BUT VERY, VERY SIMPLE WITH COMMAND LINE!

- ls -1 *.java | grep -v Test | grep -v ^Binary | xargs javac
  - List all java files, one per line
  - Filter out any lines with Test
  - Filter out any lines that start with Binary (^Binary)
  - For each line, call javac on it

THE PIPE

- A pipe (|) takes output from one program, and uses that output as input for the next program
- Very useful for chaining commands together

> AND >> : FILE REDIRECTS

- > puts the output into a file instead of a different program
- ls > files.txt - Will put a list of all files into files.txt
- >> - same as >, except will append instead of overwrite the file
  - If I did “ls > files.txt” again, it would delete whatever was in files.txt before
  - If I do “ls >> files.txt”, then I will have ANOTHER listing of files at the end

COMMON UNIX COMMANDS

- ls - List all files in a directory
  - ls -a - Show all files, incl. hidden files (those that begin with a ".")
  - ls -1 - One file printed out per line
  - ls -l - Print detailed information
  - ls -t - Sort files by time last modified
  - Can be combined
    - ls -lhat - List all files (even hidden ones) with detailed info, sorted by time, using unit suffixes

MEOW MEOW MEOW

- cat
  - Concatenates multiple files into one output stream
    - cat BinaryTree.java BinaryNode.java prints out both of the files to the output stream
  - Often just used to display a file - if used by itself, “concatenates” a file with nothing, thus displaying it
    - cat BinaryTree.java will put BinaryTree.java on the console

GREP

- Global Regular Expression Parser
  - Allows you to search for lines that contain certain patterns (regular expressions)
    - This regex can be a simple string
    - grep “meow” foo.txt - Shows all lines in foo.txt which contain the string “meow”
    - grep -v “meow” foo.txt - Shows all lines in foo.txt which DO NOT contain the string “meow”

FIND

- Find a file which meets certain parameters
  - find . -name Bill.txt - Find a file, starting from the current directory and looking down into subdirectories, which has the name Bill.txt
  - find . -type d -name Noogie - Find a file which is a directory (remember in Unix many things are “files”, even directories, links, devices, etc.) and has the name Noogie

P S

- Show all processes the user has
- ps -ef - Show all processes on the system
- pgrep - Show only the PIDs of processes whose names match a regex (very useful with xargs, next)
    - pgrep java - Shows PIDs of all processes whose names include the string “java”

XARGS

- Given a list of lines, perform a command on each one of them
- ls -1 *.java | xargs javac - Compile all Java programs (kind of alternative way of saying “javac *.java”)

ON KILLING

- kill forces a process to quit (actually sends a signal to the process, which by default is SIGTERM signal)
    - kill 9001 - sends a SIGTERM signal to process with PID 9001 asking it to shutdown
- kill -9 - SIGKILL signal (“shut down with extreme prejudice” - not ignorable or catchable by process)
    - kill -9 9001 - sends a SIGKILL signal to process with PID 9001 FORCING it to shutdown NOW

COMBINING

- Using pipes and file redirectors, we can now combine these to do very powerful and specific things!

- Example: Look for any phone numbers (in format xxx-xxx-xxxx) in HTML files under the current directory, and put into a file phone_nums.txt

  - find . -name “*.html” | xargs grep “[0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]” > phone_nums.txt

  - Another regex: find . -name “*.html” | xargs grep “[0-9]\{3\}-[0-9]\{3\}-[0-9]\{4\}" > phone_nums.txt

ANOTHER EXAMPLE

- Display the data from all the text files, except those that start with the word “Meow”, in one directory

- ls -1 *.txt | grep -v ^Noogie | xargs cat

ONE FOR THE ROAD

- Given a file processes.txt, which contains a list of process names, if any of them are running, kill them
- cat processes.txt | xargs pgrep | xargs kill

YOU TRY IT!

- Clone the repo to your local machine https://github.com/laboon/CommandLineFun
- There are several exercises in the README.md file. Please do them with a partner.
- If you do not have a Unix system (OS X or Linux), these should work with Cygwin or bash shell on Windows, or ssh into unixs