



Intro to Bash Shell Script and VIM

Iacovos G. Kolokasis

Department of Computer Science, University of Crete

What is a Bash Shell Scripting

- **Bash:** Is a command language interpreter
- **Shell:** Is a macro processor which allows for interacting or non-interacting command execution
- **Script:** Allows for an automatic command execution

- **#**: comments
- **~**: home directory
- **cd dir**: Change directory to dir
- **mkdir dir**: Create a new directory named dir
- **rm foo**: Delete a file named foo
- **ls**: Print the files and subdirectories of the current directory
- **touch foo**: Create a new empty file named foo if one does not already exist

I/O Redirection

- **ls > foo.txt:** Calls ls and write outputs to foo.txt. foo.txt is truncated (its length becomes 0) before writing.
- **cat < foo.txt:** Call cat and redirect the contents of foo.txt to its standard input. This means that any scanf calls in cat's source code will be reading from foo.txt instead of the keyboard.
- **ls >> foo.txt:** Call ls and append its output to foo.txt. If foo.txt does not exist it is created.
- **|:** pipe, similar to >

find: Search for files in a directory hierarchy

- **find . -name "hello.c"**: Look for a file named hello.c in the directory tree that starts from the current directory
- **find myexercise1 -name "hello.c"**: Look for a file named hello.c in the directory tree that starts from the directory named myexercise1
- **find . -maxdepth 2 -name "*.h"**: Look for all files ending with .h in the directory tree that starts from the current directory and up to directory depth 2

grep: Find lines matching a pattern

- **grep foo bar.c:** Search for matches to pattern foo in the file bar.c
- **grep -n foo bar.c:** Search for matches to pattern foo in the file bar.c and print the line numbers
- **grep -r foo :** Search for foo, recursively, in all the files in the current directory and its subdirectories

VIM: Modes

- Normal - **ESC**
- Insert - **i**
- Visual - **v**
- Command - **:**

- h, j, k, l or arrows : Move the cursor left, down, up or right
- `< number > gg - Move to the line < number >`
- `gg` - Move to the first line of the file
- `G` - Move to the last line of the file

Use in command mode:

- w - Save file
- w filename - Save to a file named filename
- q - Close current window
- wq - Save file and close current window

Use in command mode:

- x - Cut
- y - Copy (use the visual mode)
- p - Paste the text
- D - Kill line

Use in command mode:

- `/text to search` - highlights all occurrences of "text to search" in the file. Use `n` and `N` to move to the next or previous occurrence

Ctags Find struct & Function Definitions

Create a tag file using ctags:

- `ctags -R .` - Generate a tag file for all files in the current directory to later use with vim.

Navigate to definitions:

- `ctrl +] .` - Jump to definition of function/struct identifier under the cursor
- `ctrl + t` - Jump back to previous place in file, before using "`ctrl +]`"