

CS311 - FA13: Homework 1

Trevor Bramwell

October 14, 2013

1. Describe at least 2 ways of transferring files from a remote server to a local machine.

(a) `$ scp user@remotehost:filepath localpath`

(b) `$ {wget,curl -L} URL`

2. What are revision control systems? Why are they useful? Explain how to create a subversion or git repository on os-class (and create it, while you're at it).

A 'revision control system' is a program from keeping track of revisions to files. They are useful in software development for keeping track of the history of a project and keeping code between developers in sync.

(a) `$ svnadmin create`

(b) `$ git init`

3. What is the difference between redirecting and piping? Describe each.

Redirecting allows you to redirect a command's input or output to a file. Since stdin, stdout, and stderr are also files, they are include in what you can redirect to.

For example: `$ echo "hello, world" > newfile` would redirect echo's stdout to newfile. newfile would now contain "hello, world".

Piping connects the stdout of a command to the stdin of a second command by way of a pipe, before any redirection happens. Piping sets up a link between two commands so the output of the first can easily be chained to the input of the second.

4. What is make, and how is it useful?

Make is a language and a program for handling build dependencies. It is useful for compiling only the parts of a program that have changed. For example: say you have a program that is dependend upon functions from multiple files of 'C' code. If one of those files changes, then you will have to recompile every each one of the files along with your program.

Make resolves this by only recompiling the single file that changed, and building any other dependencies that it relies on.

Make can also be used as a simple interface for a collection of commands; similar to bash aliases.

5. Describe, in detail, the syntax of a make file.

A makefile is made up of 2 major elements: Rules and Macros. They can also contain comments and empty lines.

A makefile generally has a list of macros at the top, followed by any number of rules.

Macros are used to easily reference things like CFLAGS and LDFLAGS, and the list of objects to be compiled.

Each rule consists of a target, some optional dependencies, and a list of 1 or more commands to execute in order. If make sees a dependency is out of date when running a target, it will run the command to rebuild that dependency.

6. Give a find command that will run the file command on every regular file (not directories!) within the current filesystem subtree.

```
$ find . -type f -exec file '{}' \;
```