

# The Command Line

Matthew Bender

CMSC Command Line Workshop

November 13, 2015

# Section 1

## Micellaneous Programs

# tar

`tar` (**t**ape **a**rchive) is a program that collects many files into one, preserving directory structure.

To create a tar file, use `$ tar -cf archive.tar [files]`  
The `-c` flag means we are creating a tar file, the `-f archive.tar` tells tar what to call the tar file, and `[files]` is a list of files and directories to include.

To extract the files in a tar file, use `$ tar -xf archive.tar` (this will bring the files into the current directory, but keep the tar file).

The `-x` flag means we are extracting from a tar file, and again the `-f archive.tar` flag tells tar which file we extract from.

The `-v` flag (**v**erbose) can be added to either of these to print all the files as they are being tar'ed/extracted.

To list all the files in a tar file without extracting, use the `-t` flag

## gzip and bzip2

gzip and bzip2 are compression programs - given a file, they make it smaller.

To compress a file, use `$ gzip file.txt` or `$ bzip2 file.txt`

This will remove the original file and create a file called `file.txt.gz` or `file.txt.bz2`

To uncompress the file, use `gunzip file.txt.gz` or `bunzip2 file.txt.bz2`

Note that gzip and bzip2 only work on single files, not multiple or directories.

To work with this, often you first `tar` your files/directories, then gzip or bzip2 them:

```
$ tar -cf archive.tar [files] && gzip archive.tar
```

Most versions of `tar` are smart enough to recognize when a file is compressed, e.g. `$ tar -xf archive.tar.gz` will first `gunzip` and then extract the files.

# find

Ever lost a file? Use `find` to help you figure out where you put it.

```
$ find [dir] prints out all the files and directories in dir
```

You can add flags to specify certain types of files as well:

`-name something` will find all files whose name is `something`

So to find all text files, use `find . -name '*.txt'`

Use `-iname` instead of `-name` to perform a case-insensitive search.

`-type f` will match only files, `-type d` will match only directories.

`test1 -a test2` matches only if both tests are true

`test1 -o test2` matches if either are true

`! test` matches if `test` is false

Other tests include file size, read/write/execute permissions, modification time, and more.

## find actions

By default, `find` just prints out all the matching files, but it also allows you to take some other action for all the matching files as well.

`-delete` will delete all the files that match the criteria (e.g. `$ find . -name '*.bak' -delete` to delete all backup files)

`-exec command {} \;` will execute an arbitrary command on each file. `command` is some arbitrary command to run.

`{}` will be replaced with the filename in the command.

The command is terminated by a semicolon, but this must be escaped (`\;`) or quoted (`' ; '`) to avoid it being interpreted by the shell.

For example: `$ find files/ -name '*.mp3' -exec mv {} songs/ \;` will move all your MP3's in the files directory or below to your songs/ folder.

`-ok` is like `-exec`, except `find` will prompt you for confirmation before each command runs.

## scp

scp (**secure copy**) is a tool used to transfer files between two computers. scp uses SSH for transferring the files.

Basic usage: `$ scp user1@host1:/path/to/file  
user2@host2:/path/to/file`

If one of the files is on your local computer, you can leave out the `user@host` for that computer.

Example: upload a file to Grace:

```
$ scp code.c bender@grace.umd.edu:~/projects/4/
```

Example: pull down a project from Grace:

```
$ scp -r bender@grace.umd.edu:~/projects/4/ .
```

Note the `-r` flag is required to **recursively copy** a directory

# wget and curl

wget and curl are used to download files off the internet.

\$ wget www.example.com/file.txt will download file.txt located at www.example.com

\$ curl www.example.com/file.txt prints the contents of file.txt to stdout. Redirect it to a file to save it.

Both tools offer a lot more, such as compression, different protocols, recursive mirroring of websites, etc.

See <http://daniel.haxx.se/docs/curl-vs-wget.html> for a comparison.



## bc

bc stands for **b**asic **c**alculator - a simple utility for basic math.  
bc supports basic arithmetic operations, and using variables and functions.

Add the `-l` flag to add support for floating point division, plus some math functions like sine (`s`), cosine (`c`), arctangent (`a`), natural log (`l`), and exponentiation (`e`)

bc can be used interactively, or have expressions piped to it:

```
$ echo '5 * 20 / 3' | bc # prints 28
$ pi=$(echo '4*a(1)' | bc -l) # stores variable pi
```