Syntax

$ command -flags arguments separated by spaces

- Flags indicate command options
- Flags are optional and may be omitted
- Flags must be preceded by one or two dashes, depending on the flag and command
- Single-character flags can often be combined (e.g. $ ls -l -a can be written as $ ls -la)
- To pass an argument to Bash that contains spaces, delimit the argument with single quotes 'like this'
- " $ " at the beginning of a command is the prompt and should not be typed

Efficient Bash Usage

- Tab completion: Press the Tab key to auto-fill directory and file names
- Arrow keys can be used to see and run previous commands
- Ctrl-a will move your cursor to the beginning of the line, Ctrl-e will move your cursor to the end of the line
- Ctrl-c will cancel a command that has been typed but not entered (useful for quickly canceling text entry). If Ctrl-c is pressed when a command is running it will force-quit the command.

Paths

- Root directory : " / "
- User home directory : " ~ "
- Current directory : " . "
- Parent directory (relative to current directory) : " .. "
- Directories are separated by the forward slash character (e.g. /home/student/MIT/6.004/)
- Paths can be *relative* (to the current directory) or *absolute* (entire path specified starting with the root directory)
- Use **$ pwd** to print the current working directory

$ cd <path> : Change working directory

- $ cd with no arguments this will change the current working directory to the user's home
- $ cd – (cd with one dash) will undo the previous cd command

$ ls : List directory contents

Flags:

- -l : list file details
- -a : list all files (including hidden files)

$ less <path to file>: Display text file in isolated view with scroll support

- Press "q" to quit and return to the Bash instance

$ cat <path to file>: Print text file to the terminal

- Useful for concatenating files via the redirection operators > and >>

$ echo : Print string literals or variables to the terminal

- $ echo $? will print the exit code of the previous command; useful for debugging

$ file <path to file>: Shows file's type (text, image, video, etc.)

$ touch <path to file>: Create empty file

$ mv <path to source file> <path to destination file>: Move or rename file/directory

$ mkdir <path to directory> : Make a new directory

$ cp <path to source file> <path to destination file> : Copy a file

- The -R (recursive) flag can be used to copy directories

$ nano <path to file (optional)> : Basic text editor and scratch-pad

- Key notation: ^ refers to [Ctrl] , M refers to the Meta key ([Alt] / [Option])

$ rm <path to file or directory> : Remove file

- WARNINGS:
- When you delete files (or directories) with rm they cannot be recovered from the "Trash" or "Recycling Bin", only delete files if you are sure they will not be needed in the future
- Triple check rm commands and other dangerous commands before running
- Avoid running commands from the Internet without knowing their function
- When using rm make sure your paths do not have spaces between forward slashes

Flags:

- -r : remove recursively (will remove directory specified and all its contents)
- -f : force file removal, do not prompt before deletion

$ grep : Find text in a file or across files

Syntax: $ grep -flags(optional) 'text to search' <file or directory>

Flags:

- -i : case insensitive search
- -r : recursive (used to search within directories)

$ find : Find files or directories

Syntax: $ find <search directory> -iname '*name to look for*'

- Unlike grep, find matches name and iname exactly, so you must either specify the entire file name or use the wildcard character (*) that represents any text (including no text: " ")

Flags:

- -name : search by file/directory name, case sensitive
- -iname : search by file/directory name, case insensitive
- -maxdepth <N> : limit search to specified recursive depth
- -type d : find directories

$ type <command name>: Show command type and location

$ alias : Save a command shortcut

Syntax: $ alias alias-name='<command string here>'

- Do not insert spaces between the alias-name, the equality sign, and the initial quote
- Must be saved in ~/.bashrc (GNU+Linux), ~/.bashrc.mine (Athena), or ~/.bash_profile (Macintosh) to be persistent
- e.g. : alias athena='ssh –X {kerberos}@athena.dialup.mit.edu' will allow you to run $ athena

$ man <command name> : Show manual page for command

$ help <command name> : Show help for Bash built-in commands

$ whatis <command name> : Show one-line explanation of command

$ apropos <text> : Show commands related to <text>

$ exit : Close Bash session / terminate SSH connection

Backgrounding

- **You can make a command run in the background by ending it with one ampersand "&" (e.g. $ python3 &)**
- **If a command is already active it can be brought to the background by pressing Ctrl-Z followed by $ bg command**
- **Use the $ jobs command to show your processes. Their job number will be in brackets to the left of the process name. e.g. "[1]- Stopped emacs"**
- **To return process N to the foreground run $ %N where N is the job number**
- **To terminate process N run $ kill %N**