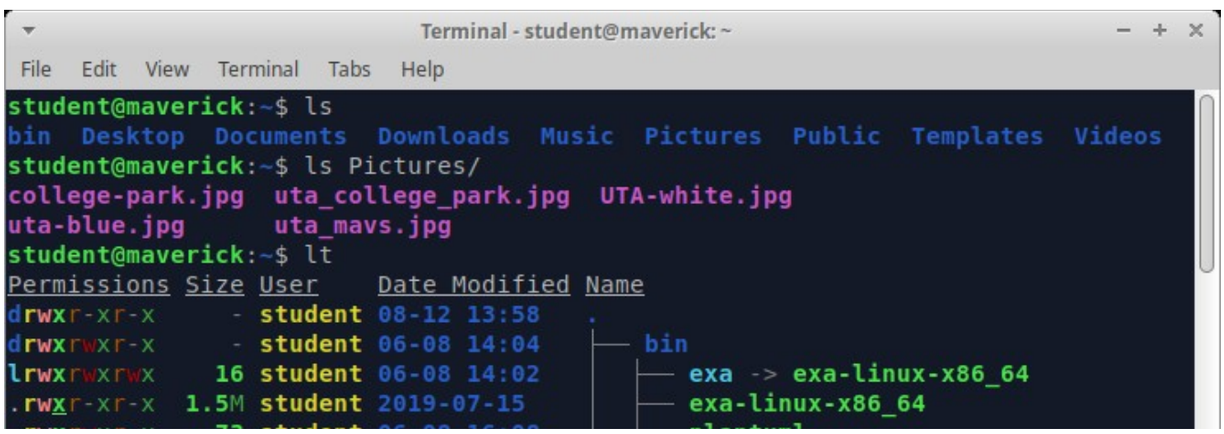
Using bash in 5 Pages

(the "Bourne-Again SHell")

1 Starting bash

- In Linux, press **Ctrl-Alt-t**. Or double-click *bash* on the desktop. Or **Start** > **Accessories** > **Terminal Emulator.** Or ⊞ > **"term"**.
- In Mac OS X, select **Applications** > **Utilities** > **Terminal**.
- In Windows, you have 2 options – git bash or run bash under Linux. Or both!
  - Install the Windows Subsystem for Linux (WSL 2), which actually runs a full Linux kernel under Windows. It's more powerful, but cannot script Windows activities.
  - Install git under Windows (see *Git in 5 Pages*), which includes a bash instance, and launch via **Start** > **Git Bash**. This includes many (though not all) bash tools, but can also be used to script Windows activities.



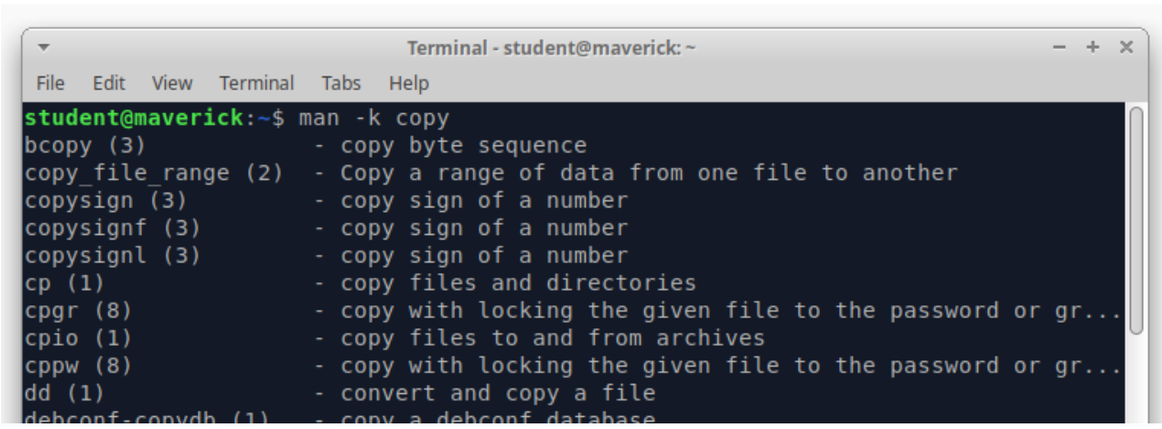
Ubuntu Terminal / Bash Tips and Tricks

- Change your password with the **passwd** command.
- Use **View** > **Zoom In** (**Ctrl-+**) to make text bigger, **View** > **Zoom Out** (**Ctrl- –**) smaller.
- The **up-arrow** key will step through previous commands, which may be edited and re-entered. Or **Control-r** ***search-term*** **Esc** will search earlier commands for one matching *search-term*. Repeatedly pressing **Control-r** searches again from that point until you find the right command.
- The **mouse scroll wheel** and the **scroll bar** on the right review previous work.
- Select text, then **right-click** > **Copy** (or **Ctrl-Shift-C** or **Ctrl-Ins**) to copy text (such as earlier command output) to the clipboard. **Right-click** > **Paste** (or **Ctrl-Shift-v** or **Shift+Ins**) pastes the text from the clipboard onto the command line for editing and submission.
- You can also ***select*** **text to copy it** to the special "X buffer", and in any window (or just within git bash) ***middle-click*** **to paste it**. This is faster than the usual method, but only works for text.
- Type part of a command, parameter, or filename, and press **Tab** to complete it.
- **Control-Z** stops the current command. Use **fg** ("foreground") to continue it, or **bg** ("background") to run it in the background, returning you to the bash prompt. Or add an **&** to the end of a command to run it in the background from the start. Select **File** > **Open Tab** or **Control-Shift-T** to open a new terminal in a tab.
- Add **.bash_aliases** from Appendix A to your home directory for enhanced commands (in blue).

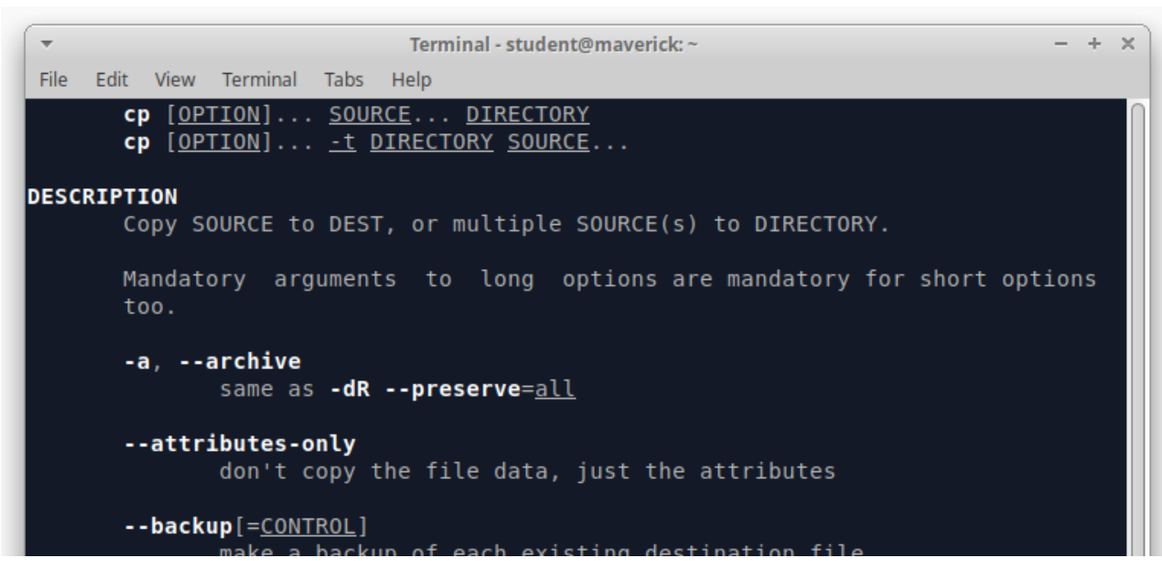
2 Getting help in bash

1. **An alphabetized list of bash commands is available on-line at http://ss64.com/bash/.**

2. **man -k [topic]** lists all commands with brief summaries related to the specified topic.[1]

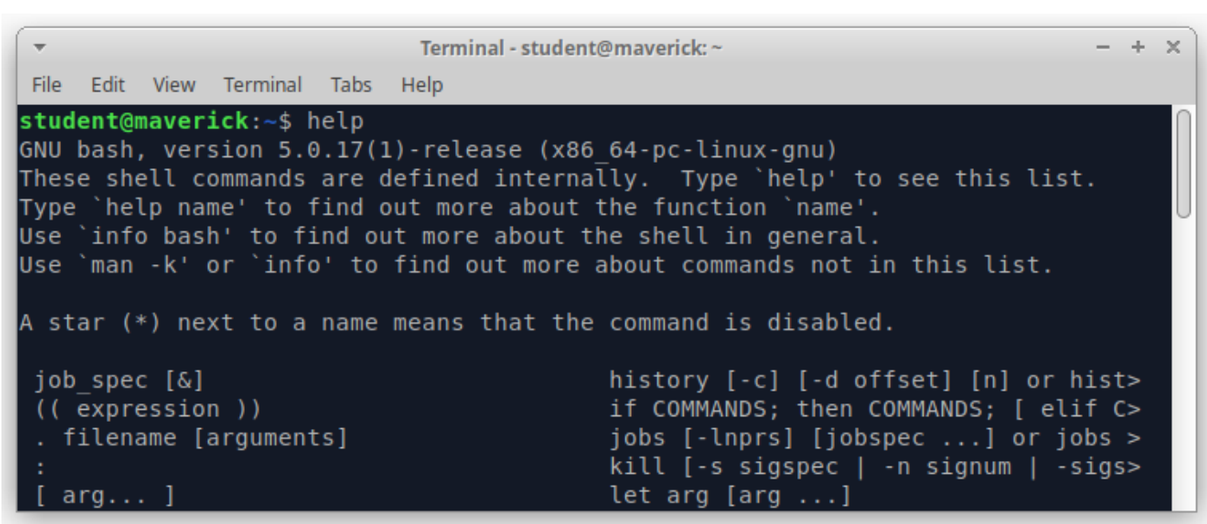


3. **man [command]** (e.g., **man cp**) displays a concise, interactive manual for any command. [1]
   Try **man man** for the manual page on man**.** [1]



- **Page Up** and **Page Down** or the **mouse wheel** will move through the manual.
- **/[word]** (slash followed by a word) searches for the first occurrence of "word". **n** successively moves to and **Shift-N** back to each occurrence of "word", wrapping from bottom to top.
- **q** will quit the manual.

4. For a list of bash keywords, try **help**. For details of a keyword, type **help [keyword]**.



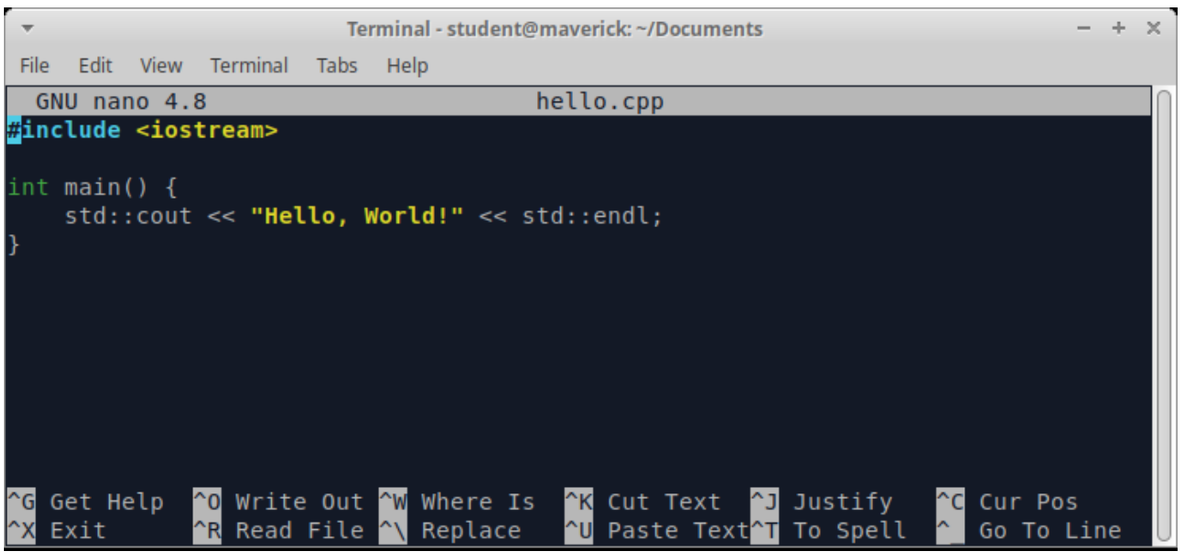
1 This doesn't work for me in git bash.

3 Editing a text file

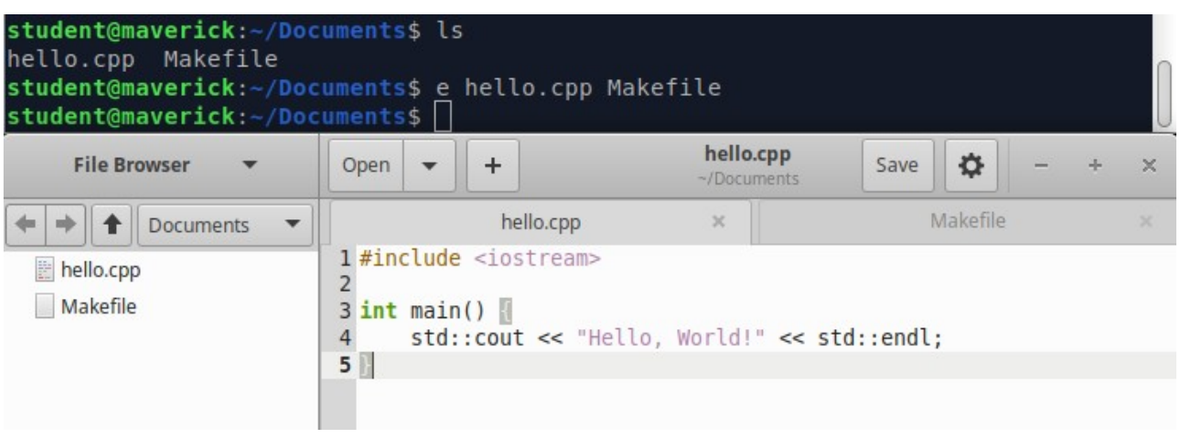
- Typing the editor name followed by one or more filenames opens them for editing. The default text editor for Mac is **TextEdit** and for Windows is **notepad**, but I recommend something better. The default text editor in Linux varies but are generally fairly capable – try **gedit** (which I'll likely use in class), **gnome-text-editor**, **kate**, **leafpad**, **mousepad**, **pluma**, or **xed**.

  Popular portable GUI editors include Sublime, the new kid on the block Zed (now also on Mac and Windows!), or (with care) Visual Studio Code (but read *VS Code in 5 Pages* carefully). Some Windows users love Notepad++ [2] (compatible with Wine on Linux). Some Mac users love Espresso or BBEdit.

- To edit within the terminal, try nano, which is less similar to typical Windows or Mac editors. Save using Control-o (shown as ^O below), exit using Control-x (^X below). A brief tutorial is available. The more adventurous might try vi / vim (I may use it a bit in class) or emacs, both of which are highly productive but with a remarkably steep learning curve.



- If using .bash_aliases in CSE-VM, **e [filenames]** *usually* opens any file in its default app.[3] **eall** will open all Java, C, and C++ files with Makefile and build.xml in the default editor. **ec [class]** will open the C++ [class].h and [class].cpp files in the default editor.



- To change the default application for a file type (e.g., text/plain for Java files), it's easiest[4] to launch the file manager, find a file of that type, right-click it, and select **Open With > Other App**, select the desired application, enable "Use as default", and click Open (or the similar process for Windows or Mac).

---

2 You may need to create an alias to use Notepad++ from the command line in git bash. See instructions in the "Text Editor" section of https://pow123.github.io/UWI-Mona/setup.html.

3 **e** is a bash *function* that relies on the xdg-open command. Use **type e** to see the code for function e. This doesn't work in git bash.

4 If you *really* want to do this from the Linux command line, the command is **xdg-mime**. But it's a bit complicated to use correctly. Lots of reading on the Internet is recommended, as well as a backup of your virtual machine.

4 Navigating directories and using files

Commands in blue text are unique to the CSE-VM virtual appliance or those using the .bash_aliases in Appendix A.

- Paths are separated with forward slashes (/home/student/), not backslashes as in Windows or colons in Mac OS. No drive letters exist – all paths start with a slash (a "unified file system").
- **ls** will list the files in the current directory (like dir in Windows' command line)
  - **ls -l** will display a "long" listing with extra information
  - **ls -a** will show all files, including those that are hidden (e.g., start with a period)
  - **lx** (in the provided appliance) gives a colorful "long" listing with even more info, and **lt** will show a 3 level deep "tree" of directory contents (based on **exa** – type **man exa** for info)
- **mkdir [name]** will create a new directory with the given name (same as cmd.exe).
  mkcd [name] (in the provided appliance) will create (if needed) and change to the directory.
- **cd [directory]** will change to the specified directory (same as cmd.exe).
- **pushd [directory]** will change to the specified directory, but remember the current directory.
  - **popd** will return to the most recently remembered directory.
- **rmdir [directory]** will remove a directory, but only if it's empty (as in Windows' cmd.exe).
  rm -fr [directory] will remove a directory *and all of its contents*, no questions. **Be careful!**
  rm [file] removes a file permanently (no trash can).
- **mv [directory] [new_name]** will move a directory (++or file) to a new name. Unlike Windows, this is instantaneous if on the same physical device.
  cp -r [directory] [new_directory] will copy directory and all of its contents to new_directory.
  cp [file] [new_name] will copy a file to a new_name, which may be a filename or directory.
- **locate [partial_name]** will list all files on the computer that contain the partial_name.
- **grep [string] [file(s)]** will search the filenames and list the lines containing the string.
  pdfgrep [string] [file.pdf] will search a PDF file.
  pdfgreps [string] will search *all* PDFS found recursively starting in the current directory. This may be useful for searching the provided lecture, homework, and exam prep PDFs. :-)
- **cat [file(s)]** concatenates (types) the contents of all listed files to the console.
  - **head [file]** shows the first few lines of the file. **tail [file]** shows the last few lines.
  - **less [file]** pages through the file one screenful at a time, with Page Up and Page Down.
- **chmod a+x [file]** will make a file "executable" (like a .EXE in Windows).
  chmod in general sets file permissions.
- **scrot -s file.png** will allow you to screenshot a window or area you select to file.png.
  e file.png views the screenshot. Full screenshot options for all operating systems:

Area to Capture	Bash	Linux GUI	Windows	Mac OS
Entire Screen	scrot hello.png	PrtScn	Win+PrtScn	Shift+Cmd+3
Current Window	scrot -s hello.png	Alt+PrtScn	Alt+PrtScn	Shift+Cmd+4+space
Select Region	scrot -s hello.png	Shift+PrtScn	Win+Shift+PrtScn	Shift+Cmd+4
File Location	./hello.png	Via dialog	Click preview	On the desktop

5 Combining commands via pipes and redirection

- **javac X.java ; java X** compiles and runs class X. The ; executes the left command, and when it exits, executes the right command. (**java X.java** will work for *simple* cases, NOT in general.)
- **java X > output.txt** sends the standard output (via System.out) to the file named output.txt.
  java X >> output.txt appends the out text to the (new or) existing file named output.txt.
  java X > output.txt 2> errors.txt sends the error output (via System.err) to errors.txt.
- **java X < input.txt > output.txt** feeds input.txt to X's in & writes X's out to output.txt.
- **java X | tee output.txt** sends the standard output (via out) to both the console and output.txt.
  The | (pipe) connects out from the left program to in of the right program.

6 Loops, conditionals, and programmerish features

- **for f in $(ls) ; do mv $f $f.txt ; done** renames (moves) all files in the current directory to the same name with .txt appended. $([command]) is replaced by bash on the command line with the standard output of [command]. $f recalls the value of the f variable.
- **for i in $(seq 1 10) ; do echo $i ; done** counts from 1 to 10, once per line. echo (like print) just repeats its parameters to standard out.
- **while read line ; do echo $line >> myfile.txt ; done** appends each line of text entered at the console to the text file myfile.txt until EOF (end of file), which is control-d.
- **while :; do echo "This is the song that never ends" ; done** repeats the annoying song forever.
- **javac X.java ; if [$? -eq 0]; then java X ; fi** compiles X.java and then runs it only if the compile succeeded.
  - **div** displays a colorful separator to help find the first error message in a build
  - **notify** "build complete" shows and speaks a notification
- **zip -r [directory]** creates a ZIP archive of the named directory named directory.zip.
  - **unzip file.zip** unzips the zip file to the current directory.
  - The name of the current directory is a dot (."), and the parent is two dots ("..").
- **diff -y X.java Y.java** displays all differences between the two files, side by side.
  In between, "<" and ">" show added lines, "|" changed lines, and " " unchanged lines.
  - **meld X.java Y.java** is a windowed diff app. Use **meld dir1 dir2** for directories.
- **ps** lists all processes (commands) with their process id ("pid") running in the current bash shell.
  ps -ef lists all processes / pids running on the computer.
  top periodically lists the "heaviest" processes running on your computer (**q** exits).
- **kill [pid]** terminates the process with the specified process id (the "pid").
  kill -9 [pid] terminates the process with the specified pid with extreme prejudice.
  xkill terminates the next GUI program you click. **Be careful!**
- **type [command]** lists the pathname, alias, function, or other info for the command specified**.**
- **sudo [command]** executes the command as the administrator. **Be careful!**
  sudo apt install [program] installs the requested program from the Ubuntu app store.
- **backup** makes a perfect timestamped copy of the current directory alongside it in the parent. This will include a snapshot of the local git repository, if any.
- **cloc *.java *.xml** counts the lines of code in your program.
- **time java X** prints how long your program runs before exiting

7 Appendix A – Custom bash Environment

You may add the blue commands from this document to your own bash environment by copying or appending the contents of https://github.com/prof-rice/cse-vm/blob/master/bash_aliases to the file **~/.bash_aliases** on your own machine. The ~ means your home directory. Note that the filename must start with a period (.), which in bash means a "hidden file" not normally shown in directory listings (although **ls -a** will reveal them: **-a** means "all").

This file will add the following commands to your bash environment (at the time this was written).

- **doc** – Change to the ~/Documents/
- **dl** – Change to the ~/Downloads/
- **usb** – Change to the /media directory (where flash drives are usually mounted)
- **dev** – Change to the ~/cse1325/ directory (your local git repository)
- **prof** – Change to the ~/cse1325-prof/ directory (my git repository)
- **e [*filenames*]** – Open *filenames* in the associated application
- **ec [*class*]** – (For C++) Open *class*.h and *class*.cpp in the default editor
- **eall** – Open all Java, C++, and Python files in the default editor in alphabetical order, with Makefile and build.xml in the far right tab(s)
- **lt [*directory*]** – List a colorful 3-directory deep file hierarchy with all file attributes (including git) – requires that **exa** be installed
- **backup** – Duplicate the current directory, appending -YYMMDD-HHMMSS to the directory name
- **mkcd [*path*]** – Create the specified directory (including multiple subdirectories) and change to it
- **paths** – List the $PATH with one directory per line
- **pdfgreps [*string*]** – Search all PDF files in the current directory hierarchy for "string" – requires **pdfgrep** be installed
- **c17 [*files*]** – Compile the specified files with the C++ version 17 compiler (**g17 [*files*]** includes gtkmm)
- **div** – Print a colorful and distinctive divider to make finding the start of error messages easier
- **notify [message]** – Display and speak the message, usually to alert you compilation is done
- **m** – (C++ only) Display **div**, run **make**, then **notify** "Build complete" or "Build failed"
- **j** – (Java only) Run **ant** if build.xml is available, otherwise run **javac *.java** (this is very useful for graders, by the way!)