A Somewhat-Quick Introduction to Using CS61A Computing Resources

By Casey Ho

I. The long and short intro

In CS61A, you will be using a Unix operating system to do just about everything. Unix is different from Windows in many respects, but don't let that scare you away from trying it out. You will be able to do most things using graphical interfaces, but from time to time you will also need to learn how to type in text commands. This document heavily focuses on the latter- if you have used MS-DOS, then some of this will be familiar. It's meant to be an introduction (albeit a lengthy one), so most descriptions are concise. It should, however, cover almost all your needs for the duration of the entire class. You are strongly encouraged to take some time to get familiarized with console commands- it won't take long before they're second nature since most are mnenoics and straightforward to use.

If it turns out you don't like the setup we have given you, you always have the option of doing a larger portion of the work on your home computer. That will require some extra work to get started. Be forewarned also, you will still need to work with Unix to some degree to get your work turned in.

If you have had some prior experience with Unix in CS3, we still recommend that you go ahead and scan through this document; the class relies much more on Unix and does not employ web-based tools like UCwise.

II. Running programs and doing your work

Much like MS-DOS, Unix is command prompt based. When you first log into your account and finish answering questions, you won't have much of a screen to look at. If a terminal isn't already open, you can right click on the background to bring up a menu and select "xterm" or "terminal" to bring up the console. This console will be where you do most things.

```
h50 [1] >                              (The prompt a.k.a your friend)
```

There's not much to note here, except the "h50". This means the server you are working on is located in Heart Field Annex 50, which is important if you try to print documents (you want to know where your printer is!). So what can you do from here on out? To execute a command, type it into the window and hit enter. Try typing "**ls**" and hitting Enter. This will show the current files in your home directory (more on this later). It should be pretty empty for now.

```
        h50 [1] ~ > ls
        core@  mail/
```
Alright, let's try something a little more complicated. Type "**emacs**" into the prompt and hit Enter. A new window should pop up with a menu bar and some other interesting things. This is Emacs, the text editor that you'll be using to do your work. It's something of a fancy Notepad, designed for programming work. We have a separate tutorial that details how to work with it. You are of course free to use whatever text editor you want if you dislike Emacs- VIM and Pico and both available. (Some TAs have never used Emacs so they can't help you out there too much- ask a lab assistant).

You might have noticed that when you typed "emacs" in the console, the window popped up and the console stopped accepting inputs. This is because emacs is bound to the console, and the console will not execute a new command until Emacs is closed. You can deal with this by either opening a new console or typing "emacs &" instead when you start emacs. The ampersand (&) will tell the console to allow you to execute more commands later on (it's like you said emacs-“and”-something-else-later). You can use ampersands with other programs, but only use it with programs that display in a new window- otherwise weird things will probably happen.

"**firefox**" will open up Firefox, a web browser. "netscape" and “mozilla” are also available, but they are older versions and Firefox is all the rage these days.

And because we're all Internet addicts, "**gaim**" will open up Gaim, an instant messanging client. It has almost all the features that AIM has- you can also use it to connect to other networks like MSN, Yahoo and ICQ. It’s much better than quickbuddy :)

One last thing to note: You have multiple desktops available for your use. This is akin to having multiple monitors- you can open Emacs in one workspace and keep Gaim in another window (or risk being distracted ;)). Play with the arrows on the bar at the top of the screen to switch desktops and move windows around.

III. Using the filesystem to store your work

When you first log into your account or open a new terminal, your console will be set to your home directory. Unix directories are much like folders in Windows- they can contain sub-directories and various files. Your home directory is your personal workspace- only your work will be in it and no other students will be able to access it. Dealing with your files will consitute most of the commands you type into the prompt I recommend that you create a new directory in your home directory for each assignment that you work on (i.e. hw1, hw2, proj1, proj2, etc.) to keep everything organized. Here are some commands that you can use:

ls [location] - **L**i**S**t the contents of a location, or the current directory by default
cd [location] - **C**hange **D**irectory to given location
cp [source-location] [new-location] Make a **C**o**P**y of a file in a new place
mv [old-location] [new-location] **M**o**V**e a file from one place to another
mkdir [location] - **M**a**K**e a **DIR**ectory with the given location
rmdir [location] - **R**e**M**ove a **DIR**ectory with the given location
touch [location] - Create a blank file with the given location
rm [location] - **R**e**M**ove the specified file

Locations can be specified in a variety of ways. The most standard method is to just type a bare file or directory name like "hw1.scm". There are also a variety of aliases you can use to reach particular locations.

./ (The current directory)
../ (The parent directory)
~/ (Your home directory)

Hmmm, we could use a demo of all of this. Let's start out with that "ls" command again.

```
h50 [1] ~ > ls
core@  mail/
```

Looks like there's nothing much to see yet. Let's make a directory for homework 1.

```
h50 [2] ~ > mkdir hw1
h50 [3] ~ > ls
core@  hw1/   mail/
```

Alright, let's change our current directory to homework 1. As described in the section on submitting homework, you'll want to do this before submitting or otherwise the submit program will ask you a lot of annoying questions (it thinks everything in the current directory might be part of your homework, which in the above example includes your mail!).

```
h50 [4] ~ > cd hw1
h50 [5] ~/hw1 > ls
```

It's a new directory, so our work isn't in here yet. Lets create a new file to store it.

```
h50 [6] ~/hw1 > touch lab1.scm
h50 [7] ~/hw1 > ls
lab1.scm
```

Oh whoops, this is our homework directory, not our lab directory! Let's rename the file by moving it.

```
h50 [8] ~/hw1 > mv lab1.scm homework1.scm
h50 [9] ~/hw1 > ls
homework1.scm
```

That's better. We can edit this file now by going into Emacs by typing "emacs homework1.scm &". Lets go back to our home directory to see what else needs to be done.

```
h50 [10] ~/hw1 > cd ~
h50 [11] ~ > ls
core@  hw1/   mail/
```

As you can see, we've made it back to where we started just by using the tilde (~). Typing "cd .." would also have worked. Okay, we're done with the homework1.scm file. Let's go ahead and delete it.

```
h50 [12] ~ > rm ./hw1/homework1.scm
rm: remove ./hw1/homework1.scm (yes/no)? yes
```

Note that "./" was typed to say "look for the hw1 directory in the current directory". Saying "yes" to the rm command also might get annoying at times, but you'll never know when it will save your rear.

IV. Submitting homework

The submit command is your friend! You will need to use this whenever you turn in an assignment. NOTE: You MUST use submit every time, we do not have scripts that automatically scan your directories like in some other classes. To use it, type the following:

> **submit** [assignment name] (Example: "submit hw1")

where assignment name is something like "hw1", "hw2", "proj1" depending on what you are submitting. If you type in "submit" with no arguments the list of possible assignments will be displayed.

Be in the directory that contains your code when you submit, otherwise the submit program will ask you if you want to submit all sorts of other files from other assignments, etc.

As for submitting assignments, please be sensible in what files you submit; obviously, you should not turn in homework assignments when you use "submit proj1". Each assignment should be turned in individually. Try to limit the number of files you submit- it is particularly helpful to the readers if you submit all your code in one file like "hw1.scm" as opposed to 10 different files each with one question. You also don't need to submit files that look like "hw1.scm~" or "#hw1.scm#". If you never explicitly created those files, then they are just backup copies used by Emacs. Try to use sensible filenames for all your projects. Each project will include files with some initial code given to you- you do not need to rename them (i.e. turning "twenty-one.scm" to "my-twenty-one.scm"). Lastly, make sure these files can be loaded properly. Just run STk and double check as follows:

```
h50 [1] ~/proj1 > stk
STk> (load "twenty-one.scm")
STk> (exit)
```

If there are no errors, then everything will likely be okay. Occasionally, some students make a mistake with the order in which they define things- stk will inform you what the problem is if you need to change it.

You can also resubmit files as often as you want (so long as it isn't past the deadline). This is particularly helpful if you want to submit a semi-working copy and don't know if you have enough time to finish the rest. If you do take advantage of this, please resubmit ALL files, not just one part.

Once you've typed "submit hw1", everything should have been taken care of on your side. If your grade for a while after everyone else has received theirs, it might not hurt to fire off an e-mail to your reader. (your reader is the one listed under glookup)

V. Checking your grades

You can check your grades using the command "**glookup**". Just type it in and you will get a report on all your assignments that have been graded so far. You usually don't need to worry about extrapolated totals- glookup almost never gives a good estimate. Just be sure to keep track of your performance on the assignments for which you received grades. If you are worried about how you are doing, be sure to talk to your TA. If you want to complain about an individual grade, send an email to the reader that graded your assignment (their login will be listed in glookup). Give the reader a few days to respond before complaining to the professor or a TA; they will defer grading decisions to the reader.

VI. Reading e-mail and getting in touch with your TA/reader/professor

"**Pine**", the basic e-mail and newsgroup client, is very important in cs61a where material will be covered rapidly. If you need to get in touch with me or get general help/advice, e-mail and newsgroups will be your speediest mode of communication- you don't need to wait for office hours. Typing "pine" in the console will bring up a menu. You use both arrow and letter keys to navigate around the menu; the commands you can execute are always listed at the bottom of the screen.

One thing to note is that sometimes the command looks like "^M", i.e. it has a carat at the start. This means you should press both the control key and the other letter, in this case Control-M.

If you have a general question that needs to be answered, try posting it to the newsgroup instead to get an answer. The TAs promise to check the newsgroup as often as they check email, which is daily at the very least, so you're better off with the chance that someone else will answer the question sooner. The newsgroup is basically a forum; there will be threads of questions and their responses.

To set up pine to access the newsgroup, do the following:

1) Type "pine" at the prompt
2) Press "S" to enter setup
3) Press "C" to enter config
4) Change "personal-name" to your name- we want to know who you are!
5) Change "nntp-server" to "news.berkeley.edu"
6) Press "E" to exit setup (say yes to save these changes)
7) Go to "Folder List"
8) Go to "News on news.berkeley.edu/nntp"
9) Press "A" to add a newsgroup and enter in "ucb.class.cs61a" as the value
10) You're done! Just hit enter on ucb.class.cs61a to read postings.

In the future, to get back to the newsgroup just repeat steps 7, 8, and 10. Please adhere to the etiquitte mentioned in the class handout on the newsgroup; the last thing we want is a flamewar adding numerous useless postings. You should answer questions from other students if you can do so; your help will be much appreciated. Also, try to see if an answer has been posted before asking a question. It helps reduce clutter from redundant postings.

Most students want to view the newsgroup with the most recent postings first. If you are one of them, go back to the config menu (steps 1 to 3) and change sort-key to “reverse arrival”.

You can also set up a news client (Eudora, Outlook, and Netscape Mail all have functionality to do this I believe) to read the newsgroup just like any other Internet group. You may need to use a server instead of news.berkeley.edu if you are off-campus.

Also, you can read the newsgroup using just a web browser at http://groups.google.com/group/ucb.class.cs61a?hl=en. This solution may not necessarily contain the latest postings (google may not update for several hours), but it’s a convenient way to do a very quick initial search for postings without having to log into your instructional account.

And lastly, it may be good to get into the habit of constantly monitoring for new e-mails using your favorite client. You can access your class account e-mail using IMAP also from Outlook, Eudora, Netscape, etc. Instructions are at:

http://inst.eecs.berkeley.edu/connecting.html#email

Please, please, please check your e-mail at least once a day! Time-critical information may be sent to you via it!

VII: Email part Two: getting your e-mail sent elsewhere

Pine should be easy enough to use, but it can be a pain to have to check more than one e-mail account at a time. You can have your cs61a e-mail sent to another account by typing in the following command:

```
        h50 [1] ~ > echo ‘myemail@berkeley.edu’ > .forward
```

If you ever want to use a different e-mail address, just edit the “.forward” file in Emacs. If you want to save a copy of your e-mail on your account AND send it elsewhere, your .forward file should have

```
        \cs61a-xx, myemail@berkeley.edu
```

in it.

VIII: Some shortcuts for advanced users

When typing in commands into the console, it may seem tedious at times to write out long names. Don’t worry! You can also use wildcards and tab-completion to specify names, if you don't want to type as many characters.

Tab completion is by far one of the niftiest tools you can use to navigate the filesystem- if you've tried using the commands you can see that it may take a while to type out everything. Say you're trying to change your directory to "homework1". All you need to do is press tab after you have typed "cd ./h" and the system will fill in the rest for you if it can. In this case, say there is another directory named "homework2" next to "homework1". After you press tab, your prompt will change to "cd ./homework". Because there are no other directories that start with the letter h, such as "hello", the computer knows you can only complete the statement using "omework" and types it for you.

If there was a conflicting directory like "hello" then pressing tab will have no effect.
If you see this happen, press tab twice in succession (like double-clicking). The names of the files that can be used with what you have typed so far will be printed, so the terminal will inform you that "homework1", "homework2", and "hello" are all options for "cd h".

Tab completion will also work for typing in the name of a program. Just don’t type the directory specifier (e.g. “./” in the examples above) and the computer will attempt to fill in the name of a command instead.

If you want to affect a number of files at once, you can use wildcards. Typing an asterik is a synonym for all files. If you say "rm *" this means remove all files (DONT DO THIS if you're not sure!!!...I've done it before to my home computer and had to spend many hours restoring my hard drive...you may not be as lucky). A more common usage is to say "rm hw1*"; this means to remove all files whose names begin with "hw1". This is useful for getting rid of batches of files.

The "cp" command can be used for copying more than one file at once. "cp -r" will copy recursively (I hope you understand what this term means by the end of the class)- this is necessary to copy whole directories, especially directories within other directories.

Similarly, "rm -r" will delete files recursively. Beware, this is nearly as dangerous as "rm *", so double check that you know what you're deleting.

IX. The Last Resort

If you are confused with any of the commands listed here, you can just type "**man** [command]" to see a description of what the command does and a full explaination of how to use it along with its options. Man stands for the command "MANual". For example, “man ls” will tell you all the wonderful different ways you can list the files in the directory- you can sort by time, filesize, show more details and hidden files, etc etc etc.

X. Quick Reference Sheet

The Bare Necessities

emacs	Main text editor
stk	Scheme interpreter
mozilla	Web browser
pine	E-mail and newsgroup reader
gaim	Instant messanging

Grades and assignments

submit [assignment]	Submit the specified assignment
glookup	Display current grades

Files and directories

ls [location]	**L**i**S**t the contents of a location, or the current directory by default
cd [location]	**C**hange **D**irectory to given location
cp [source] [location]	Make a **C**o**P**y of a file in a new place
mv [old-location] [new-location]	**M**o**V**e a file from one place to another
mkdir [location]	**M**a**K**e a **DIR**ectory with the given location
rmdir [location]	**R**e**M**ove a **DIR**ectory with the given location
touch [location]	Create a blank file with the given location
rm [location]	**R**e**M**ove the specified file
less [file]	Show the contents of a file
head [file]	Show the start of a file
tail [file]	Show the contents at the end of a file
pwd	Display the **P**ath to your **W**orking **D**irectory
find [location] **|** grep [name]	Displays the names of files that contain a given name
grep [term] [files]	Displays line numbers of files that contain a given term
quota	Display how much of your disk quota space you are using

Printing

a2ps [file]	Prints a file
lpr [file]	Prints a file in actual size (uses more paper)

Miscellaneous

passwd	Change your password
man [command]	Display information on how to use a command