# CLA CS-101 (B1)

## Introduction to Computers

Fall 1995

## Homework # 2

## Due on Tuesday 10/3/95

$\boxed{1}$ In this problem, we will take you in a quick tour through some of UNIX's simple commands. The solution to this problem consists of your written answers to the `boldfaced questions` in all parts below, and the printout that you will obtain from part l.

a. Log in as described in the "logging in" section of Problem #6 in Homework #1.

b. At the UNIX prompt type $\boxed{\text{ethics}}$. Read and keep in mind what comes out on the screen! **Where should violations of the stated guidelines be reported?**

c. At the UNIX prompt type $\boxed{\text{man cal}}$ to find out about the "cal" program. **What is so special about September 1752?** Also, use "man" to look at the "script" command. **How does the -a option modify the behaviour of the "script" command?**

d. **How many people in Boston University have "Malloy" as their last name?** Use the $\boxed{\text{ph malloy}}$ command to find out. **What is the phone number of the CS Department Administrator?**

e. At the UNIX prompt type $\boxed{\text{script problem-2-1.solution}}$ to start a script file of your session.

f. At the UNIX prompt type $\boxed{\text{date}}$ to get the date and time.

g. Find out who is using **CSA** by typing $\boxed{\text{finger @csa}}$ at the UNIX prompt.

h. Display the calendar for the year in which you were born using the "cal" command. For example, if you are born in 1974, type $\boxed{\text{cal 1974}}$. **What day of the week were you born?**

i. Use the $\boxed{\text{calc}}$ command (interactively) to compute 12+23/7 and to compute (12+23)/7.

j. Using the $\boxed{\text{webster}}$ command, display all the 13-letter words in the English language that start with an "x" and end with an "a". Also, find out all the words in the English language that start with a "w", end with "nt" and contain a "c". Also, look up the meaning of "automaton".

k. End the script of your session (that started in step e. above) by typing $\boxed{\text{exit}}$.

l. Get a printout of your script file from one of the printers in MCS B24. At the UNIX prompt type $\boxed{\text{lpr -Psiskel problem-2-1.solution}}$ to get the printout on printer "siskel". **What would be the command to get the printout on printer "ebert"?**

m. **How many lines, words, and characters are there in the script you obtained in part l?** Type $\boxed{\text{wc problem-2-1.solution}}$ to find out.

**2** In this problem you will get a taste of the basic editing capabilities of Emacs.[a] By the end of it, I hope that you will feel comfortable using Emacs as an editor. Your familiarity with Emacs is extremely important for you to be able to keep up with upcoming attractions (I mean homeworks!)[b] There is nothing to hand in for this problem. However, the results of this problem are needed for the next problems.

a. Start an Emacs session by typing emacs at the UNIX prompt.

b. From within Emacs, create a file called *foo* using the CTRL-X CTRL-F command. In this file write an essay of about 1 full screen (about 30 lines), in which you describe what you expect to learn in CS101, your opinion about the material covered and the style of the lectures, and whether you like to see any changes in the course contents. Save your essay every now and then using the CTRL-Xs command. Also, if you want to check the spelling of a particular word from within emacs, position the cursor on that word and then type ESC-xspell-word followed by the "return" key. Get out of Emacs using the CTRL-X CTRL-C command. Remember to save the files you are editing before getting out. If you don't, Emacs will ask you if you want to save your files before getting out.. Remember, if you don't save it, you loose it!

c. Start another session on Emacs (same as step a.) and retrieve the contents of the file *foo* using the CTRL-X CTRL-F command. Did you find your essay intact? (It'd better be!) If you need to make changes go ahead..

d. Without leaving Emacs, create a new file using the CTRL-X CTRL-F command. Call the new file *foobar*. In this file describe in few words what each of the following commands does: CTRL-X CTRL-F, CTRL-P, CTRL-N, CTRL-B, CTRL-F, CTRL-V, ESC-V, CTRL-L, CTRL-D, CTRL-K, CTRL-Y, CTRL-T, CTRL-X CTRL-C, ESC->, and ESC-<.

e. Without leaving Emacs, type CTRL-Xb.[c] Emacs will ask you whether you want to switch buffers. If you press the "return" key, Emacs will go back to editing the *foo* file. So, go ahead and press the "return" key to switch buffers.

f. Repeat step e. to go back to editing *foobar*. Now, add to *foobar* a description of what you think CTRL-Xb is doing.

g. Get out of Emacs using the CTRL-X CTRL-C command.

h. There is much more to `emacs` than what is described here. Some of these features will be discussed in the lectures or in the sections.

---

[a] If you want to spend more time learning about Emacs, you may want to try the teach-emacs command, which will take you in a tour of many of Emacs' features.

[b] The symbol CTRL- is used to indicate the use of the "control" key. For example, CTRL-A means press the "control" key and then (while still pressing it), press the character "A". The sequence ESC-means pressing and releasing the "Escape" key. For example ESC-v means press the "Escape" key, release it and then press the character "v".

[c] Notice that this is different from CTRL-X CTRL-B

**3** In this problem set you will learn how to read and send Electronic Mail. There are many programs that one could use to send and receive Electronic Mail. The program we will be using is called `pine`.

a. Start a `pine` session by typing $\boxed{\text{pine}}$ at the UNIX prompt.[a]

b. The main menu of `pine` will be displayed and will give you a number of choices. Also, at the bottom of the screen, `pine` will display all the options available to you by typing the highlighted character. For example, typing the character $\boxed{\texttt{q}}$ will make you quit `pine`.

c. From the main menu of `pine` type the character $\boxed{\texttt{c}}$, which will allow you to send your first email message. Once the character `c` is typed, `pine` will clear the screen and display an empty email message. The top of that message includes four fields that tell `pine` what to do with the email message you will compose.

d. The first thing that `elm` will need to know is the addressee, i.e. *who* should receive your message. Following the `To:` field on the very top of the page, type in the e-mail address of the user whith whom you want to correspond electronically. Well, your first message will be addressed to me. So, on that line, type my e-mail address, which is $\boxed{\text{best@cs.bu.edu}}$.[b]

e. Next `pine` will need to know if Carbon-copies of your message should be sent to other recipients. What about sending a copy of the message we are composing to the teaching fellow? And, for your own records, what about sending a copy of the same message to yourself! Here is how.. Following the `Cc:` field type $\boxed{\text{rip@cs.bu.edu}}$, followed by a comma, and then your own E-mail address... Done!

f. The `Attchmnt:` field is used to describe any attachements (i.e. enclosures) to be sent with the message. For now, let's leave that field empty.

g. Next `pine` will need to know what is the Subject of the message. Following the `Subject:` field type $\boxed{\text{Problem-3}}$ followed by your full name... Done!

h. Now you are ready to compose the message. To do that, move the prompt below the line that says `----- Message Text -----` and start typing whatever you want to send me (you have already learned how to use Emacs as an editor in the previous problem, and most of the commands you learned will work with `pine`)... What I want you to send me is the essay that you prepared in problem 2 (part b.) This essay was saved in a file named *foo*. To include the contents of this file in the message you are composing type $\boxed{\text{CTRL-R}}$, which instructs `pine` to include a file into the message you are composing. As a result, `pine` will ask you for the name of the file to be inserted. Following the `Insert file:` prompt at the bottom of the screen type $\boxed{\text{foo}}$ and then press the "return" key. Now, you should have your essay inserted in the message.

i. Go to the end of the message and sign it – Well, just write your firstname or nickname.

j. Now, you are ready to send the message. To do so, just type $\boxed{\text{CTRL-X}}$. `Pine` will want you to confirm that you indeed want the message to be sent. To confirm, simply type $\boxed{\texttt{y}}$ (for "yes"). Congratulations!.. You have just finished sending your first electronic message...

k. Repeat the above process. This time send more messages (at least 4 or 5) to yourself – NO MORE MESSAGES TO ME OR TO THE TEACHING FELLOW. In the body of your messages just type any text. Also, you might want to exchange messages with friends of yours taking CS101-B1.

j. Get out of `pine` by chosing the "Quit" option from the main menu.

---

[a] The first time you start `pine`, a notice may be displayed regarding the creation of a directory in your account where `pine` will store messages you will be receiving.

[b] Notice that if you just type `best`, the system will assume (by default) that you mean `best@cs.bu.edu`. This is another instance where a "default" value is useful. Can you recall other instances you came across?

**4** In problem 3, you learned how to send messages using `pine`. In this problem, you will learn how to read messages sent to you.

a. Start a `pine` session.

b. As soon as you start, `pine` automatically will check your electronic mailbox, which is stored in a special `folder` called the `INBOX`. If you have any mail, the number of messages will be displayed on the upper right corner of the screen. To view these messages, type `I`. As a result, `pine` will give you a list (or Index) of all messages in your `INBOX` folder. To view any one of them, simply move to that message (using the up and down arrows) and press return. To get back to the index of all messages, type `I`.

c. While reading a message, or while browsing the index of messages, `pine` allows you to delete a message (by pressing `D`), forward a message to another recipient (by pressing `F`), or reply to a message (by pressing `R`).

d. Once you have tried these commands, get out of `pine`.

e. There is much more to `pine` than what is described here. For example, you can create a folder where you store all the messages from your best friend or another folder where all the messages from the TF are stored, Some of these features will be discussed in the lectures or in the sections.

**5** In problem 2, you created two files *foo* and *foobar* from within Emacs. These files are stored in your main directory. As we discussed in class, to be better organized, it's a good idea to make filenames meaningful. Also, we learned that it is a good practice to keep our files organized in directories to make future access easier. In this problem, we will use some Unix commands to accomplish these goals. The solution to this problem is the script you obtain from step (h) below.

a. Start a script using the `script` UNIX command (see problem 1 for "how to do a script").

b. Using the `mkdir` UNIX command, create a new directory under your main directory and call it "ps2" (for problem set 2).

c. Using the `cp` UNIX command make a copy of the file *foo* into the "ps2" directory. Name the new copy *course-evaluation.txt*.

d. Using the `cd` UNIX command, change directory to "ps2". Make sure you are there by using the `pwd` command.

e. Using the `cat` UNIX command, take a look at your essay (now named *course-evaluation.txt*).

f. Repeat steps c. and e. for *problem-2-1.solution* and *foobar*. Your copy of *foobar* should be given a meaningful name (for example, you may want to rename *foobar* to be *emacs-commands-reference.txt*).

g. Finish the script by typing $\boxed{\text{exit}}$ at the UNIX prompt. Get a printout of your script.

*Have Fun!*