Comp 200
Barland

Getting Started: Owlnet

Handout #2
97.Jan.22

This handout provides introductory information on how to use Owlnet, send e-mail, and read news.

1 Getting an Owlnet Account

You will need an owlnet account for Comp200. If you have not done so already, do the following to get your own owlnet account.

1. Sit down at a computer in any computer lab on campus (e.g. Ryon 102), and get the login prompt. This prompt will probably be there when you sit down.
   - If the screen is blanked, just jiggle the mouse or press a key to end the screensaver.
   - If the screen has nothing but some message like "`chub, 2:34 pm, idle 1h 23m (Press ^C)`" then press control-C: i.e. type "c" while holding down the control key, in the same manner you would hold the shift key down to capitalize a letter.
2. Normally you would type in your login name in response to the "login:" prompt. But since you don't yet have an account, type "apply" (followed by the return key). (The computer may or may not further ask for a password; if it does, press the return key, and sigh at the computer's petulance.)
3. Follow the directions on the screen, and answer the slew of questions. One thing you'll be asked is for your *login name*; some people think it's cool to have login names like "empress" or "slugface", but be warned that whatever login name you choose, you'll be stuck with it for the rest of your stay at Rice! You'll also be asked for a password (which you can—and should—change regularly); heed what it says about keeping your password secure. It is a violation of Owlnet policy to give out your password to anybody. It will also ask for your password. A few obvious precautions: never write your password down (on paper or on-line), never let a friend know what it is, never forget it.

Within 24 hours, you should have your owlnet account; your e-mail address will be *yourLoginName*`@rice.edu`.

2 Register your account for `comp200`

In order to tailor your Owlnet account for Comp 200 a bit, there is a one-time step to do: "register" for Comp 200. Log in and type `register comp200`. It will ask you two questions, to which `yes` is the correct[1] answer.

[1]One of the questions is a (justifiably?) paranoid question from Owlnet, asking if you want to see *exactly* registering comp200 will do. Feel free to choose the option of looking at the scripts, to get a glimpse of (an inelegant) programming language, the C shell.

One effect of registering is that you create your own subdirectory named `comp200`, which is initially set to be unreadable by other students. You are encouraged to do all your files for this class inside that directory. (For an explanation of subdirectories, read Section **??**; for a bit more about file protection see Section **??**.) Registering also customizes your UNIX environment for Comp200 a little. You need only `register` once, though if you do it several times nothing evil will happen.

3 Netscape

Netscape is a window-based program that displays *hypertext* documents from the World-Wide Web. Hypertext documents can contain text, pictures, sound, movies, and (perhaps most importantly) links to other documents. The World-Wide Web (WWW) is an interconnected collection of these hypertext documents. The act of browsing the World-Wide Web of documents is called "Web surfing" or "net surfing."

To run Netscape, either

- Move the mouse cursor into the background, hold down the right button, select Netscape, and release the button. Or,
- Type `netscape &` into a command shell[2], followed by Return.

After a few moments, a new window titled Netscape will appear on your screen. You may need to click in the window to make the Netscape information disappear and to see the Comp 200 home page.

The first time that you use Netscape, it will display a license agreement. Read it and click on the Accept button. Netscape may subsequently display a window stating your account has not yet been initialized for Netscape. Click on the OK button to initialize your account; Netscape will respond by creating some special files in your account (their names start with a period, so they are ignored as uninteresting by many UNIX programs) and displaying the Owlnet "home page".

In a hypertext document from the World-Wide Web (WWW), phrases highlighted by color and underlining are linked to other pages; click on them to see that reference (no matter where it is in the world). Try it! To return to the previous page, simply click on Netscape's most useful button: the Back arrow near the top left of the window.

You can directly access an arbitrary Web document by clicking the Open button at the top of the window or by just typing into the "Location" edit box near the top of the Netscape window. For instance, the URL for Comp 200's home page is `http://www.owlnet.rice.edu/~comp200`. Fortunately, you rarely have to specify an explicit location (or URL in WWW jargon) while net surfing.

If you find a document of particular interest, you can remember it by creating a bookmark for it: pull down your Bookmarks menu, and select Add Bookmark. Any time

[2] The `&` at the end of the command line means "don't wait for me to finish this program—Netscape, in this case—before letting me issue another shell command."

you want to revisit that page in the future, you can just select it from the Bookmark menu (where it is now listed). You may wish to add the class home page to your bookmarks right now.

To exit Netscape, select Exit from the File menu.

After lab, you should look through the Comp 200 home page, which contains office hours, homework, a link to the course newsgroup, etc. The Comp 200 home page is displayed whenever you start Netscape and is also available by clicking on the Home button near the top of the Netscape window.

3.1 Caching

One thing to watch out for: after Netscape fetches a page, it *caches* it—that is, it saves a copy of the page in *your* directory (inside a directory `.netscape`). This is usually convenient, since if you view the same file in a few minutes or tomorrow morning, Netscape doesn't need to have it re-transmitted from some other computer; it just retrieves the version it cached.

However, caching is a double-edged sword: Netscape is not smart enough (grrr...) to double-check with the remote site that the file hasn't changed in the mean time. If you look at a page that changes frequently, you may need to force Netscape to reload the file by clicking on the Reload button. (If you want, you may alter Netscape's caching behavior through the Options menu.)

3.2 Viewing Postscript files

In Netscape, go to the class's home page, and click on the link "Handouts", and select the entry for "Handout 2", which happens to be a file in the Postscript format. A separate program (Ghostview) is used to display postscript files. Note that Postscript is a rather bulky format, and sometimes you have to be patient to see the file; also Owlnet has Netscape configured so that Postscript files residing inside Owlnet cannot be viewed from outside Owlnet.

Once viewing the Postscript file, you can use the space bar (or the return key) to page forward, and backspace to page back.

4 Reading News

Newsgroups are electronic bulletin boards where everybody can post and read articles on certain topics. There is a newsgroup specifically for this class, `rice.owlnews.comp200`. You should read the newsgroup every day or two, as helpful hints and clarifications are posted there regularly. Also, if you have a question relating to the class, post it to the newsgroup. Similarly if you have an answer for somebody else's question!

In newsgroups, the idea is that once you've read a message, you probably don't ever want to see it again (unless you do something special). So your newsreader "marks" an article as read, and next time you read news it only presents you with unmarked articles.

There are thousands of other newsgroups which you can participate in. We recommend that you read a group for a while before posting articles to it, so that you understand the group's flavor and don't do something inappropriate. (For instance, `alt.computers.religion` is not about automating church offices, but is a forum for humorous "religious wars" between proponents of different varieties of computers, such as IBM vs. Mac.) You should also read the FAQ (Frequently Asked Questions) for the group so you don't repeat a common request, and consider reading `news.announce.newusers`, which contains other information for new users of netnews.

There are many different news-reading programs you can choose from. Two leading contenders are Netscape and Pine, each of which makes reading news analagous to their way of reading mail.

4.1 Reading News with Netscape

From the comp200 web page, follow the link "The Newsgroup". This opens a separate Netscape window, with three subwindows, or panes: the list of newsgroups (upper left pane), the list of articles in current newsgroup (upper right pane), and the current article itself (lower pane).

- The current article: Space pages down, and backspace pages up.
- The list of articles in current newsgroup: Click on an article to read it; to see the next article there is a keyboard shortcut of Alt-down: Hold down the Alt key (which on my keyboard happens to be *not* the one labelled Alt, but instead the one with a diamond box on it; go figure) while pressing the down-arrow key, which is 2 on the numeric keypad.

  An unmarked (unread) article is shown in bold face, and has a green diamond between the Sender and Subject, while a marked (read) article is in normal-font, with a dull grey dot instead of a green diamond. As you read articles, you see them change from unmarked to marked.

  How to do something special, to be able to read articles again later: you must change an article from "marked" back to "unmarked".

  - If you just read the article now, it is still is still listed in the upper-right pane (in dull grey). No problem: to designate it as "unmarked", just click on the grey dot immediately to the left of the Subject. This will unmark the article, the diamond will become green again. Clicking once more will mark the article again.

    You can also flag articles in Netscape with a small flag icon; this is different than marking them un/read.
  - But what if you want to unmark an article that you read yesterday? Yesterday's marked articles don't even show up in the list-of-articles, so you can't immediately use approach(1) above. Solution: Go to the "Options"

menu, and select "Show all articles", rather than "Show only unread messages". Now you can scroll up and read previous articles, perhaps even unmarking some of them for the future.

There are also buttons you can use to mark an entire newsgroup as read, or mark just a certain (presumably inane) thread as read.

Articles also "expire" after a while. For instance, you don't see articles posted three years ago, simply because the news server (the local rice computer that gets/posts news from other computers on the net) doesn't store articles forever. How old an article has to be before it expires varies; rice class-newsgroups tend to stay around several semesters, while articles in alt.sandwiches.pouting expire after about a week.

- The list of all newsgroups: Click on a newsgroup to read it. Subscribed newsgroups are shown with a check to the right of their name, and unsubscribed newsgroups just have a dot; click on that symbol to *toggle* (exchange the on/off status) that particular newsgroup's subscription status.

  Note that initially, only subscribed newsgroups are shown (and the class newsgroup, since you selected that one in particular to start the newsreader). Under the "Options" menu, you can modify this by selecting either "Show all newsgroups" or "Show subscribed newsgroups".

  Finally, note that some newsgroup categories (like `alt`, `rice`, etc.) may have a triangle to the left of there name. When this triangle is pointing down, the view is expanded to show all newsgroups within that category; click on the triangle to toggle whether it points down or not.

To post your own message to a newsgroup, make sure it is selected, and click the "To: News" button in the upper left corner. You'll be put inside an editor to compose the message; when done you can either click a "Send" button to post the message, or you can select Close from the File menu.

A couple of helpful hints about altering the arrangement of information within the pane:

- The relative size of the different panes really bugs me. Fortunately, this can be changed: look for a small, beveled square between two panes: by dragging this, you can re-size the panes.
- You can change the width of individual columns within a pane: move the mouse along the column titles, and when the cursor changes between columns, click and drag.
- You can rearrange the order of columns by dragging a column header to where you want it to be.

Note that under the File menu of Netscape's *News* window, Close just closes the News window only, while Exit actually means exiting *all* Netscape windows.

4.2 Reading News with Pine

(If you are happy reading news from Netscape, you can skip this section.) Pine is a program commonly used for reading and composing e-mail (see Section **??**). But you can also set it up to to read news, by doing the following:

1. From the main menu, choose *S*etup.
2. When it asks which setup task you want, choose *C*onfig. After doing this, you should see a list of options, most of which probably say "No Value Set".
3. Using the arrow keys, highlight the item "nntp-server", about the fourth item on the list. ("nntp" stands for something archaic, like "network news transfer protocol").
4. If you're like me, you'd think "return" would tell Pine you want to change the highlighted value. I try this, but no, it tells me to choose the option *A*dd Value. So choose that.
5. Type in `news.rice.edu` as the news server.
6. Exit Pine; the next time you start, one of the options you will have will be to read news.

When you choose to read newsgroups in Pine, it will give a list of those groups to which you are currently subscribed. Choose *A* to subscribe to the class newsgroup or others. To get a list of, say, all newsgroups, start to type in `rice.`, and then see that typing `^X` (remember, that's control-X) shows you a list of all matching groups.

5 What and Where is Owlnet? UNIX? X?

You may have noticed that the previous sections glossed over some underlying points, and left some non-obvious terms (like "UNIX") undefined. The remaining sections provide some of that background.

Owlnet refers to the campus-wide collection of computers for use by students, faculty, and staff. Some labs at which you can use these computers are Ryon 102 (the SW corner), and Mudd Lab. 103 Mudd Lab is also where Owlnet's Consulting Center office is located; out front they have many helpful handouts there—check them out some time.

UNIX is the *operating system* for many Owlnet workstations. An operating system is the program which keeps track of low-level details of the computer's operation, such as where files are are physically located on a hard drive, what other programs are currently (waiting to) run, and so on. It also provides a way for you—the user—to tell the computer what you want it to do. In the case of UNIX, your wishes are communicated via a *shell*, which is just a window into which you type commands[3]. Some actual UNIX commands you need to type are covered in Section **??**.

5.1 X Windows

For details, see the handout *Introduction to the* UNIX *Operating System* available from the consulting center in mudd. But the basics of getting started are pretty intuitive if you have ever used a Macintosh or PC.

To log in to your account, go to any Owlnet computer, type your login name, and (to convince the computer you are who you claim to be) your password.

If you are on a computer running X Windows, you will see a window appear in the center of the screen. This window happens to be an xterm ("X terminal), and is running a UNIX shell. When the cursor is positioned somewhere over the xterm, its border will be slightly emphasized, and the shell is waiting for you to issue commands (see below). If you are not on a computer running X windows, but instead are on some other type of terminal, that is essentially as if you are restricted to one xterm (and no mouse), and the next paragraph won't apply.

But before typing anything, just play around a bit with the window manager:

- To move a window around on the screen, click on the window's title bar with the *middle* button.
- To resize a window, click on the small "baseball diamond" in the upper-right corner of the window with any button.
- Finally, try clicking on the background (the "root window") with the right button, to get a menu of programs to run. In particular try choosing `xterm` from that menu: after a moment, the shadow of an xterm window will pop up; move the shadow to where you want the window to appear, click, and voila!

Logging Out

When you are done using the computer, it is important you log out, so nobody else can mess with your account (see Section **??**). Be sure you have saved any files you were working on, then type `exit` in the console window (upper left corner of the screen). *Never turn the computer off.*

5.2 Issuing UNIX Commands

"Having two xterms is great", you say, "but what are they good for?" Remember that the xterm is running a shell, and a shell is your way of telling UNIX to do things. The shell is politely waiting for you type something any UNIX command; it lets you know

[3]You can also start some programs using only a series of mouse clicks; it is a *window manager* that handles this. Technically, shells and window managers are not part of the operating system ("OS"), but just interfaces for you to give commands to the OS.

this with its *prompt*, which is probably something like `great-gray%` , where the text before the `%` is the name of the particular machine you are logged onto.

We'll try a couple of UNIX commands: Move the mouse over an xterm (so that it whatever you type is seen by the xterm), and type a `ls` followed by the return key. The command `ls` lists your files; try it. You were probably underwhelmed, since you haven't created any of your own files, and probably just saw one or two files put there when your account was created. For a better time, list some other files—perhaps those in the class directory: `ls /home/comp200`. After finishing each `ls` command, UNIX again presents you the prompt, and dutifully waits.

Try a different UNIX command: `finger`. This command shows a list of people logged onto your particular computer at this moment (which is probably just you, if you are on a workstation). Like the previous command, `finger` can be more interesting if you give it an *argument*. Try `finger` *yourLoginName*`@rice.edu` instead[4].

5.3 The UNIX command paradigm

The general paradigm for issuing UNIX commands is simple, but it helps to be explicitly aware of it:

1. the UNIX shell gives you a prompt;
2. you type in one or more words;
3. the first word had better be the name of a UNIX program[5]
4. UNIX finds that program and starts it running, giving that program the other words you typed in on the command line (if any),
5. when the program finishes, UNIX goes back to step 1.

Note that UNIX is famous for being terse: if an error occurs (such as you type in something that isn't a valid command) it will let you know, but if everything works great, UNIX doesn't explicitly acknowledge that; it just gives you the next prompt.

The other words you type in, after the program name, are up to the program to interpret how it likes. Often, those words are the names of other files to work with (as in `ls /home/comp200`). It is not uncommon to also give *flags* to the program, to tailor how it works. For instance, rather than straight `ls`, you can try `ls -l`, which tells `ls` to file information in the long format. Try `ls -l /home/comp200`. Another flag, `-a` is used by `ls` to mean "all": try `ls -a`, and you'll find you actually have more files than you realized!

[4]If you want to change the information shown about you, try `finger @rice` to see how to contact the registrar about changed.

[5]In particular, if you type in the name of a (text) file to UNIX, it will give an error "Permission denied", since a text file is not a program, and you don't have permission to run it as a program.

5.4 Viewing and Editing Files

So far we've seen how to list the names of files, but how about seeing what's *inside* of them? To just send the contents of a file to the screen, you can use the program `more`: try `more .login`, or (if you want to see where it gets its name from) `more /usr/dict/words`.

To actually create files of your own and edit them, you want to use a text editor. There are a variety of text editors out there—`pico`, `emacs`, `vi`, `aXe`—and any will suffice. I recommend starting with `pico`, though; you can learn it as you go, and it shares some commands with `emacs`, a real howitzer of an editor that we'll be using later in the semester.

For your Scheme programs, we'll be using `drscheme`, which has its own built-in editor.

For practice, try `pico grok`. After typing this, `pico` will take over the entire window, and you are now editing a file named `grok` (which, since it didn't exist before, is automatically created). You can just type words into the file, look at the bottom of the screen to note that `^O` writes out a file, and `^X` exits. After exiting the editor, you'll be...where? Right—the UNIX prompt again. Now you can try `ls` and `more grok` to greater satisfaction. If interested, pick up the owlnet handout on using pico, and/or the handout on using emacs.

See the last page of this handout for a reference sheet of some of the most common UNIX commands. Also, see the owlnet handout "Introduction to the UNIX Operating System". Happily skim over the parts that don't seem particularly useful or interesting; then in a week or two, look them over again, and different parts will make more sense.

6 UNIX File Organization

You've probably seen graphical representations of files systems on a Mac or PC: icons of files and folders, and the concept of traversing the file system by opening folders within folders to reach a file.

The UNIX file system has this same essential structure, except that to use a file/folder, you have to type out the name of the file/directory, instead of selecting an icon. Your account has its own directory, your *home directory*, in which you can store all your files, create your own subdirectories, etc.

UNIX also has the concept of the *current (or working) directory*. (This concept is also present in the Mac/PC desktop ("active window"), but is not as prominent.) In your shell, at any moment you are "sitting" in some directory. When you log in, you start at your home directory; you can change directories with the `cd` command, for instance `cd /home/comp200`. You can see that you're in a different directory by running `ls`—when you don't specify what files to list, `ls` lists the current directory.

Finally, note that there are two ways to specify a file: using its complete name (which starts with a `/`, as in all examples so far), or the relative name, i.e. relative to your current directory. For instance, after the above command to change your current directory,

`/home/comp200/Projects/sound-library.ss` and `Projects/sound-library.ss` both name the same file. When using relative pathnames, it's helpful to know how to climb *up* the directory tree: In every directory[6], there is a strangely-name file `..` which is the *parent* directory.

7 Keeping Your Files Private

By default, anybody on owlnet can read your files! You must take special measures to make your files unreadable by others. Read pp. 22–23 of the owlnet handout "Introduction to the UNIX Operating System". The quick fix is to make *someFile* unreadable is `chmod go-rwx` *someFile*. The (not entirely useful) mnemonic for this is: `ch`ange `mod`e so that `g`roup and `o`thers do not (`-`) have permission to `r`ead, `w`rite, or e`x`ecute the file named *someFile*. Of course, *someFile* can be a directory, in which case all the files inside the directory are safe from viewing by others.

When you `register comp200`, the directory `comp200` it created is set up to be unreadable to others; you are encouraged to do all your files for this class inside that directory.

In all your classes, you shouldn't leave your homework files unprotected. Even in classes like Comp200, where you are encouraged to work *with* others on the homework, it is still forbidden to just copy somebody's file and try to figure out what they did. (For comp200, if somebody helps you on figuring something out, you should still be the one typing in your file. If the help was anything non-trivial, be sure to credit them. If you work jointly with other student(s), the graders are particularly happy—just turn in one copy of the homework with both (all) your names. None of the above necessitates copying a file.) If there is any doubt of what is allowed, *please* ask the professor for that particular course.

Note that the system administrators can look at all files regardless of permissions, and may actually do so if they suspect a threat to owlnet (file) security, or if you are under investigation by the Honor Council. You can use `crypt` to encrypt and decrypt files; see `man crypt` for details.

Finally, a wise word from the Owlnet User's Guide:

> Leaving your workstation unattended is dangerous to your personal files, your reputation, and to system security. People have taken advantage of such unwary users by erasing their files, sending rude mail to third parties, and setting up ways to do these things again in the future. Obviously, such actions are unacceptable and will be punished; however, punishment of the malicious user comes after the damage is done.
>
> When you are running a windowing system (such as X Windows), a program is available through a menu option to lock a workstation's screen. You should use this program whenever you are going to be away from your station for

[6] except one!

a short period of time. The lockscreen programs available on the system will log you out of your workstation at the end of a twenty minute period. This should be enough time to go to the restroom, pick up a printout, or get a labby's attention. Users are not allowed to use their own versions of lockscreen programs.

8 E-mail

As with reading news, there are several different UNIX programs for sending and receiving electronic mail. One of the most self-explanatory is Pine. To invoke Pine, type `pine` at a UNIX command prompt, or move the mouse to the root background, click the right button to produce a menu of programs, and select the pine command in the submenu entitled mail. Pine gives you step-by-step instructions on how to read and send mail. Try it out, sending mail to the person next to you in lab, or your roommate, or even yourself!

You may also use Netscape as a mailer, by clicking on the envelope in its lower-right corner. This is handy if you are often running Netscape, but you can't use it from a text-only terminal (as you can Pine).

Sometimes, it is even handy to send email using the simple program `mail`. To send mail to (say) `president@whitehouse.gov`, simply type at the UNIX prompt `mail president@whitehouse.gov`; type in the Subject: line as prompted, and then just start typing the body (remembering to hit the return key at the end of each line). To finish the mail and send it, type a dot `.` on a line by itself. (To cancel a message press control-C (`^C`).) You can also use `mail` to read your mail (type `mail` at the UNIX prompt, and then "help" at the mail prompt), though again Pine is easier to learn.

9 Printing Files

You can print files from the UNIX prompt with the `lpr` command, for example `lpr myHomework.ss`. This should be set up to send the output to a printer in your current lab. If you want to specify printing to a particular printer anywhere on campus, use the `-P` flag: `lpr -Pryonps myHomework.ss` prints the file `myHomework.ss` to the printer "ryonps", located (strangely) in the Ryon lab. For a list of printer names, their locations, and costs of printing, type `lprloc`. You are charged through the registrar.

Other related commands are `lpq` to view the print queue, and `lprm -`, the hyphen meaning to remove *all* your pending print jobs from the queue[7] Both of these use the `-P` flag in the same way `lpr` does, e.g. `lpr -Pryonps -`.

[7] In order to remove a print job from the queue, you must issue the command from the same machine where you started it.

10 Using `tcsh` Instead of `csh`

Finally, one slightly advanced but very handy (and not often explained) thing I would suggest doing, is change your default shell from `csh` ("the C shell"), to a newer spiffed-up version called `tcsh` ("the T shell"). It is 100% compatible with C shell, and has a passel of nice new features. The main things it adds are (1) the ability to use the arrow keys to go back, edit, and re-issue previous UNIX commands, and (2) using the tab key for filename completion[8] (i.e., halfway through typing a long filename, you can press the tab key, and tcsh will fill in the rest of the filename if it can). See `man tcsh` for a list of more differences.

To change your default shell, use the program `chsh`. It will then ask you for your password, and for the new shell you want, to which you should type `/usr/local/bin/tcsh`. You only have to do this once; it will take a day or so to take effect.

[8] You can also get filename completion in `csh`, but I'd recommend using tcsh anyway.

UNIX Survival Commands

UNIXcommand	short description
`ls` [*file...*]	list directory/files; `-l for long listing`
`cp` *src targ*	Copy (duplicate) the file *src* to the file *targ*
`mv` *src targ*	Move (rename) the file *src* to the file *targ*
`more` *file*	display contents of *file*
`pico` [*file*]	edit the contents of *file*
`cd` *dir*	change directory to *dir*. Note that the (peculiarly-named) file `..` happens to be the name for “the parent directory of this current one”.
`mkdir` *dir*	make a brand new directory named *dir*
`chmod o-rwx` *file*	change mode: make it so others cannot read, write, or execute your *file*[9]
`lpr -P`*pname file*	Print out *file* to the printer named *pname*; the file had better be something printable (either plain text, or on laser printers you can also use PostScript files (ending in `.ps`).
`rm -i` *file*	forever permanently irrevocably delete *file*; the `-i` flag asks for confirmation first[10]
`alias` *new oldCmd*	rename a command
`!mo`	repeat the last command starting with the letters “mo .

In particular, I like to improve `ls` by saying[11] `alias ls ls -F`. The `-F` causes `ls` to show directories with a slash `/` and the end, and programs with a splat `*`.

[11] In fact, I like it so much, I have this line at the end of my `.cshrc` file, so it is in effect every time I log in.