Introduction to Unix for Engs 22

E. W. Hansen and M.J. Fromberger; rev. 6/02 C. Sullivan

Introduction

The **operating system** of a computer is the software responsible for organizing all of the computer's basic functions, such as processing user commands, maintaining files on disk, managing memory, and communicating with external devices such as printers and other computers. You are already familiar with the Macintosh operating system and/or Microsoft Windows.

Unix is yet another operating system, widely used in "high end" workstations by companies like Sun, Hewlett-Packard, and Silicon Graphics (SGI). Unix was originally developed in the 1970s by a small group of computer scientists at Bell Telephone Laboratories and achieved wide popularity because it was simple, elegant (at least, it was elegant then), and portable to a wide variety of computers. **Linux** is a recent operating system very similar to Unix which has become popular on PCs. Superficially, Unix resembles the old Microsoft DOS operating system (or rather, DOS resembles Unix) — the user types commands rather than clicking on icons. This can make it difficult, or at least unpleasant, for Mac and Windows users to learn Unix. Fortunately, you can do all of your work in Engs 22 with only a few commands. Furthermore, Unix supports a graphical user interface system, called **X-windows** (developed at MIT), which makes it a lot easier to use.

Thayer School has a network of SGI workstations located in Rooms 225 and 227, plus some Linux machines in room 227. You have been assigned an account on the Thayer School network for Engs 22, if you did not have one already. You can use any of the SGIs or Linux machines; your files reside on a server and are accessible from any machine. You can also access the same set of files on the same server from a Thayer PC if you log in with the same username and password—the files are in the

Dartmouth College supports a separate campus-wide network of Unix workstations, known as **Northstar** or **UWE** (**U**nix **W**orkstation **E**nvironment). There are no Northstar workstations in Cummings. Accounts for the Thayer machines are separate from accounts on Northstar. Files can be copied from one network to the other, but are not shared transparently.

This handout is intended to be a quick reference to get you started using Unix on the Thayer network. After reading it, if you want more information you may obtain help by talking to one of the system administrators in Room 225 Cummings, or by using online help:

- For Thayer School computing, go to: `http://thayer-help.thayer.dartmouth.edu`.
- For general information on using the SGI workstations, run the `insight` program, which makes several on-line "books" available.
- For help with a particular Unix command, type `man <command>`, which brings up a **man**ual page for that command.

The basic Unix commands are platform-independent; they work the same on any brand or model of computer running Unix. However, the details of window manipulation and editing may vary, particularly from brand to brand. This handout assumes you are using an SGI machine.

Log in

The first thing you do when beginning to work with a Unix workstation is log in to the system. In response to the `login:` prompt you type your user name *in lower case*, followed by `<return>`. Unix will then ask you for your password. When you type it in, it will not be displayed. If you get it wrong, Unix will respond with `login incorrect` and will repeat the `login:` prompt. Passwords are case-sensitive, so you must be careful to follow correctly the capitalization you used when you created your password. If you become convinced that you don't know your password (or that Unix forgot your password), contact a consultant or administrator to get it straightened out.

You will immediately be placed into a graphical environment. It may ask you if you would like to create a new desktop environment, share a desktop environment, or copy one. Select "Share" — this will make it so that any changes in the way that windows are arranged on your screen will be carried over (shared) if you log onto a different SGI workstation later.

Logging out

It is important to log out when you leave the computer. This makes sure that nobody can alter your files while you are away, and it also enables others to use the computer. You may not "reserve" a computer by leaving yourself logged in, and if a system administrator discovers you have done this, he will log you out.

To log out on the SGIs, simply select Log Out from the Desktop menu. A dialog box will pop up asking you to confirm that you want to log out now. Select "Yes" and you'll be logged out.

Window manager

Unix is a holdover from the dark ages of computing, before graphics terminals were commonplace, and it shows. In an attempt to modernize the Unix interface, the X-Windows system was invented. Roughly speaking, X is to Unix as the original Microsoft Windows was to DOS — a layer of software covering up the underlying clunkiness of the command-line interface. X is the graphical user interface that most modern workstations come equipped with.

X is a very powerful system. With it, the "look and feel" of a computer is highly configurable by the user. In addition, X allows multiple connections to machines anywhere on the Internet. You can run a program on a machine halfway around the world and get graphical results on your screen here. It is also rather daunting to first-time users, which is why companies like Silicon Graphics have invested heavily in additional software layers, called **window managers**, built upon X, that make their systems easier to use. You can set many options in the window manager, from desktop layout to color schemes for the windows.

Each application you run will bring up one or more windows. You can also bring up a window which enables you to type regular Unix commands. On an SGI, this window is called a "winterm" and is created by selecting *Desktop* → *Open Unix Shell* from the toolbar, or by typing `winterm <return>` in the *Console window* (the main first command window that opens, with a grey background in the usual default configuration). Do not use the Console window for general work. Usually you can just iconify it (see below) to get it out of the way. (To open a terminal window on one of our Linux machines, click on the terminal icon (with a foot in front of it) at the bottom of the screen.)

For the most part, the window manager behaves in a way that you would expect. If you hold down the mouse button with the cursor on the title bar of a window, you can move the window around the screen. If want to change the size of the window, position the cursor at the edge you desire to change. When you're in the exact position, the cursor will change shape. Hold down the mouse button and drag to resize the window. There are three buttons at the top of each window. The one farthest to the left has a menu that drops down when the button is clicked. The other two buttons iconify the window (shrink it to an icon), and expand the window to full-screen.

There is one very confusing difference between an X-based window manager and the MacOS or MS Windows environment which you must understand. In X, the mouse pointer must be located within a window in order for you to type in it. It isn't enough to just click in the window. If, when you type, nothing seems to happen, check the position of the mouse pointer. The "active" window, the one containing the mouse pointer, is always shown in a slightly different color than the others.

Unix file system structure

On the Mac and PC, you keep documents in folders, and put folders in folders to make a hierarchical organization of your documents. Unix likewise uses such a "tree-structured" file system, but instead of documents and folders represented by icons, the file system is viewed as lists of files called **directories**. Each file is in a directory, and directories may have subdirectories. The highest-level directory, where all the system stuff lives, is called the "root directory", and its name is simply '/' ("slash"). Every other file and directory on the system has a unique

path from the root of the filesystem to the file or directory itself. For example, your home directory on the Thayer SGIs is `/home/y/yourname`. where *y* = the first letter of your login name (*yourname*). This path can be abbreviated `~yourname` (note the tilde). If you have used DOS or Windows, you'll note the similarity, but beware: Microsoft uses the backward slash (\), while Unix uses the forward slash (/).

Unix uses two abbreviations for commonly used directories. It refers to your current working directory as ".", and its parent directory as "..". Hence, if you are working in your home directory, "`..`" is the directory `/home/y/`, "`../yourname`" is a redundant way of referring to your home directory itself.

To manage your files and directories, you can type UNIX commands in the winterm shell, or you can use the graphical browser that is opened from the Toolbar, under Desktop, Access Files.

There are several Unix commands which operate on directories. These are (you don't actually type the < > in `<dirname>` or the `[ ]` in `[directory]`):

`pwd` — tells you your **p**resent **w**orking **d**irectory (the one you're in).

`ls [directory]` — **l**i**s**ts the contents of the named directory. If no directory is given, it defaults to your pwd. Like all Unix commands, `ls` can be modified with certain flags. Two of the most important forms are:

- `ls -l` gives the **l**ong form of a listing, including protection codes (read / write permissions), owner, size, and date last modified.
- `ls -a` shows **a**ll files in a directory, including "hidden" files whose names begin with "." (e.g., `.login` or `.cshrc`).
- `ls -al` does both (long form for all files)

`cd [directory]` — **c**hange **d**irectory from your pwd to the named directory. If no directory is given, it defaults to your home directory.

Examples: If you are in your home directory, then

- `cd ..` changes you to `/home/y`
- `cd /home/h/hername` changes you to a friend's directory.

`mkdir <dirname>` — creates (**m**a**k**es) a new **dir**ectory called "dirname". If "dirname" is a full path specification (such as "`/home/e/eric/es22`"), then the directory "es22" is created as a subdirectory of "`eric`". If "dirname" is just a single name without slashes (such as "`es22`"), then the directory "es22" is created as a subdirectory of the pwd.

`rmdir <dirname>` — **rm**oves a **dir**ectory. This can only be done if the directory is empty of files.

Filenames

Filenames can be any practical length (up to 1024 characters, in practice), and usually have extensions which indicates what kinds of files they are. For example, "name.cpp" is source code for a program written in the C++ language, and "name.m" is a MATLAB "M-file".

Frequently used Unix commands which operate on files are (you don't actually type the < > in <filename>):

`cat <filename>`	literally, "con**cat**enate", outputs the contents of a file to your screen. `cat` is actually a much more powerful command than this. For details, see the manual (type `man cat <return>`).
`more <filename>`	whereas `cat` spews characters until it reaches the end of the file, sometimes scrolling stuff off the top of your screen, `more` divides its output into pages, enabling you to go through it carefully. To go the next page, hit the space bar. To go only one line, hit the return key.
`cp <old> <new>`	(**cp**y) makes a copy of "old" called "new".
`mv <old> <new>`	(**mv**e) renames "old" to "new". The difference between `cp` and `mv` is that after `cp`, "old" still exists, but after `mv`, "old" is gone.
`rm <filename>`	**rm**oves a file. `rm` has some powerful options, which you should not use unless you really know what you are doing.

Unix has two "wildcard" characters which simplify the naming of files in commands. "*" matches any number of characters and "?" matches any single character. For example, "design.*" refers to all files named "design" in a directory, regardless of their extension (e.g., design.c, design.o, etc.). "paper?.txt" matches files named "paper1.txt", "paper2.txt", etc. You can copy the contents of an entire directory somewhere else by using `cp <olddir>/*.* <newdir>`. You can selectively list only the C source files in a directory by typing `ls *.c`. With practice, you'll get the hang of it. But beware: when used alone, "*" and "*.*" basically mean "everything", and should be used with discretion — the commands `rm *` and `rm *.*` remove every file in the present directory!

Editing files

The preferred editor (on both the Thayer and Northstar networks) is called **`nedit`**, and it behaves more or less like a word processor on your personal computer. Another editor, which is more powerful but a bit harder to learn, is called **`emacs`**. It has an extensive help system (just pull down the Help menu and look for the tutorial). You can also skip these and just use MATLAB's built-in editor.

Printing files

The basic printing command is `lpr`. At the Unix command prompt, type

`lpr -P`*`printer_name file`*

where *printer_name* is the name of the printer you want to use, and *file* is the name of the file you want to print. The common printer names are `lw225-1` and `lw225-2`. Both are in 225 Cummings. Another convenient printer is `lw` (in the student lounge on the second floor of Cummings, above the Great Hall).

lpr is normally used for printing text files or PostScript files. MATLAB has its own printing command for graphics, but m-files are in fact just text files and so are printed using **lpr**.

Example. To print the file "foo.m" on the printer lw225-1, type `lpr -Plw225-1 foo.m <return>`. You can also use a shorter form, `lpr -P225-1 foo.m <return>`.