CS 101 – Fall 2002, Mock
Introduction to Unix and the UAA Computing Lab

1. Introduction

This tutorial and assignment will orient you to the basics of using Unix and the computing facilities in the CS computer lab. The tutorial will assume that you will be logging in remotely over the Internet using a Windows-based machine. Contact your instructor for assistance if you are using a different machine or operating system and need help. If you happen to be in Anchorage, you can log in and perform the same operations from the Computer Science lab located in CAS 170A.

2. Getting Ready to Log In Remotely

Most of the Java programming will be done on a Unix machine named **mazzy.math.uaa.alaska.edu** located in the computer science lab. You will be given a username and password so that you can log in remotely to this machine over the Internet from home or some other computer that is connected online.

2.1 Using Putty

To connect to **mazzy**, you must first have some software installed on your computer. The first is an *ssh* client, where ssh stands for secure shell. This is a program that allows you to log in remotely to unix systems and type in commands from a text-based command line. Before ssh, most systems used the *telnet* program to connect to remote systems. However, telnet does not encrypt any data, allowing the possibility for a third party to intercept your passwords or other data you may type. In contrast, ssh will encrypt all data so that a third party cannot eavesdrop.

Once you are connected to the Internet, you can download a ssh client. I recommend you use PuTTY which is a free program that you can download from:

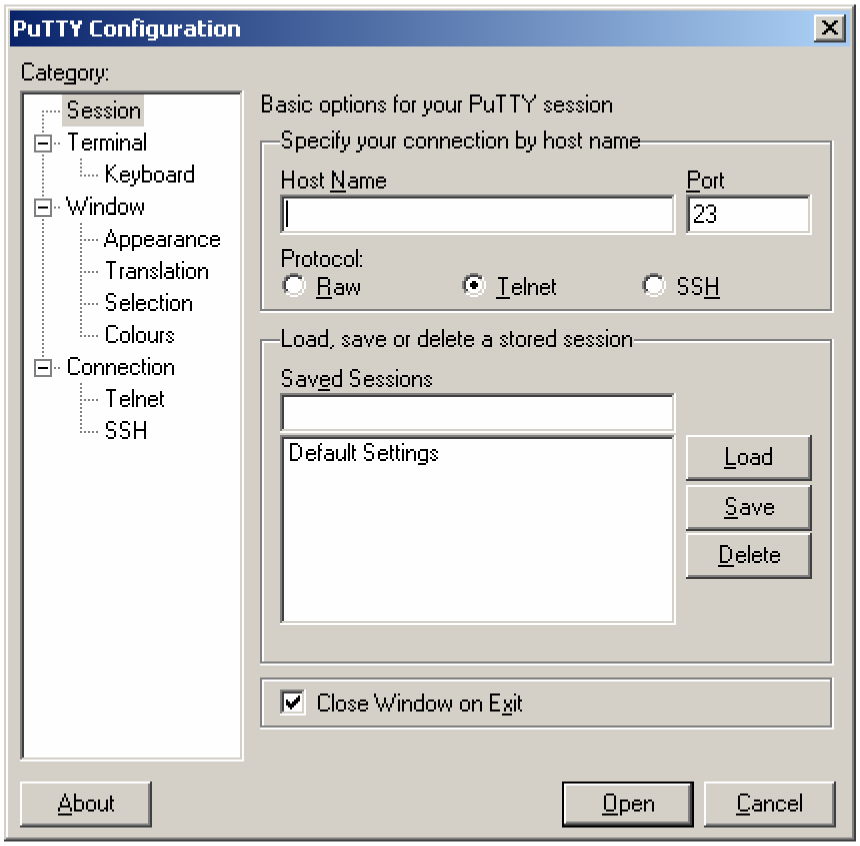
http://www.chiark.greenend.org.uk/~sgtatham/putty/

You only need the file **putty.exe**.

The program is simple to install – just download it and run it directly:

putty.exe

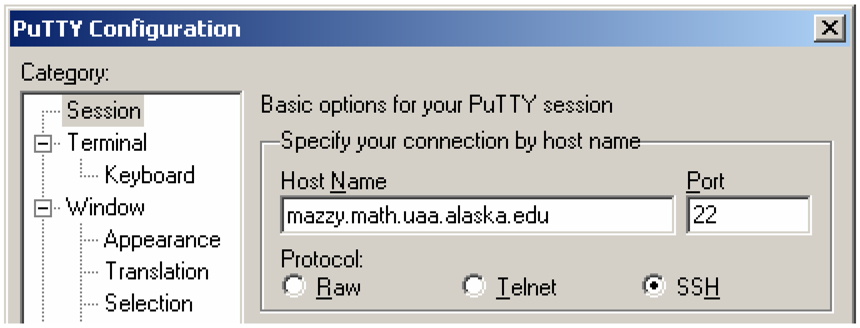
Upon starting up the program, by default the program will assume you are using telnet and will ask you to enter a host name:



To connect to mazzy, enter in the Host Name box:

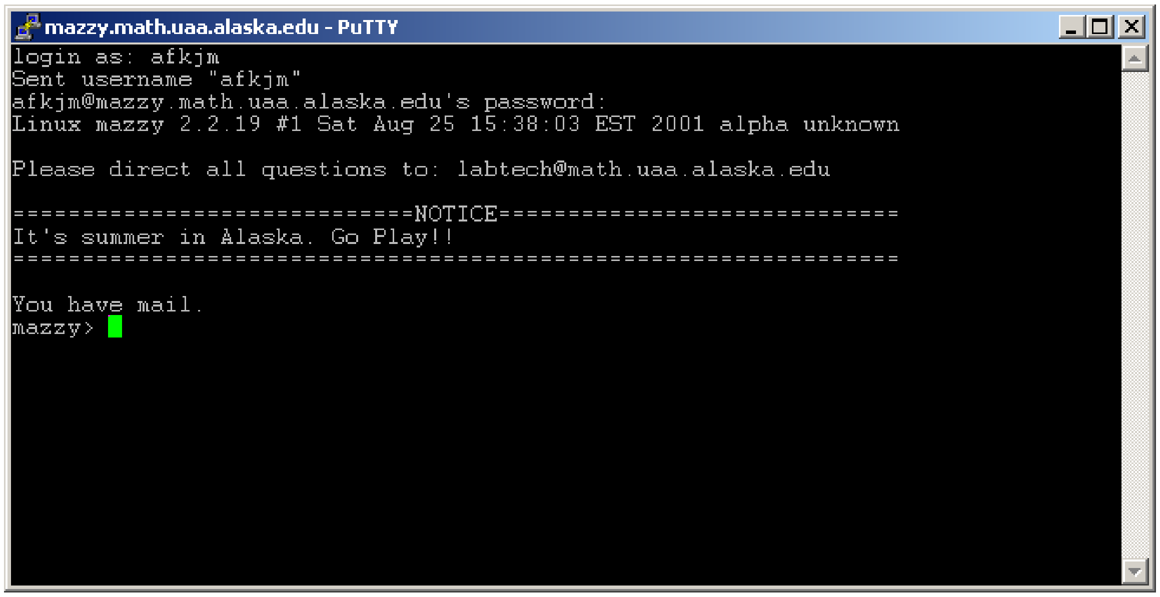
mazzy.math.uaa.alaska.edu

And then click the "SSH" button under Protocol:



Feel free to customize the colors in the window by selecting the options in the left hand pane. For quicker login, you can save your settings by giving the session a name and then click the "Save" button.

Finally when you are done, click the "Open" button to connect to mazzy. You will be prompted for your username and password:



```
login as: afkjm
Sent username "afkjm"
afkjm@mazzy.math.uaa.alaska.edu's password:
Linux mazzy 2.2.19 #1 Sat Aug 25 15:38:03 EST 2001 alpha unknown

Please direct all questions to: labtech@math.uaa.alaska.edu

============================NOTICE============================
It's summer in Alaska. Go Play!!
==============================================================

You have mail.
mazzy>
```

In this image, I have logged in using the username **afkjm**. I also entered my password, which is not echoed on the screen as it is typed. If the login is successful, you'll see a prompt indicating that you are online.

You will be emailed your username and password. If you have not received it, contact your instructor to get it!

2.2 Using WinSCP

Another program you will likely find useful is *scp*. Scp stands for secure copy, and is a way to copy files from one remote system to another. Just as with ssh, all data is encrypted so a third party can't eavesdrop and intercept the contents of the file.

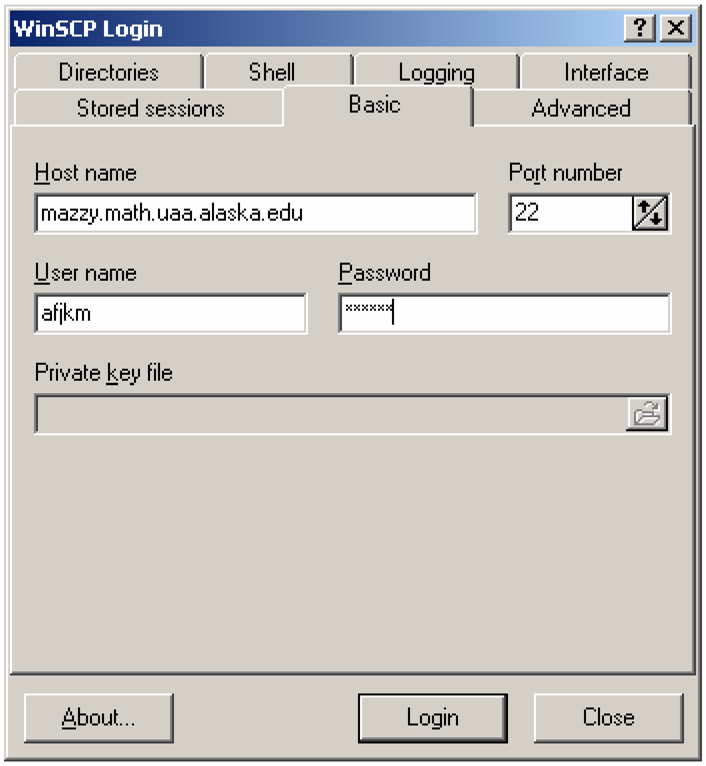
This program will be useful when you want to transfer files from one computer to another over the Internet. For example, you might develop files on your computer at home and want to upload them to mazzy. To do so, the easiest way is to use scp to copy the files.

With a Windows system, the program I recommend you use is WinSCP. This is another free program and it is available online from:

http://winscp.vse.cz/eng/

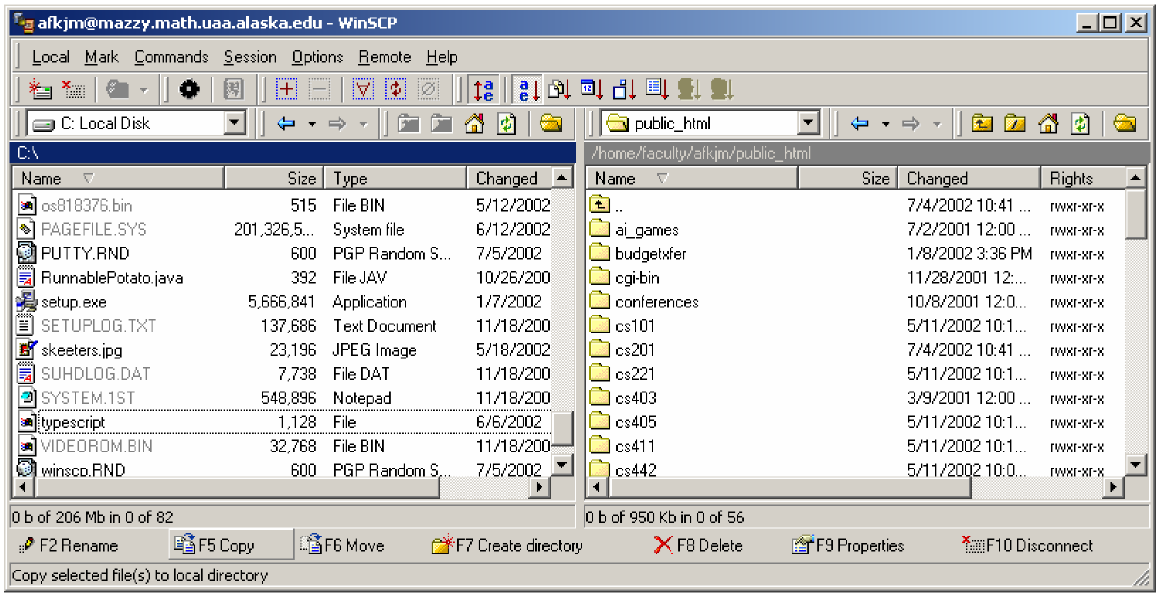
Download the file winscp.exe:

Upon running WinSCP, you'll be prompted for the machine to connect to:



Host name: mazzy.math.uaa.alaska.edu
Port number: 22
User name: afjkm
Password: ******
Private key file:

Enter mazzy.math.uaa.alaska.edu for the host name to transfer files from your Windows machine to mazzy. Also enter your username and password. Click login to start. After you are logged in you will see a window split in two sections as shown below:



C:\

Name	Size	Type	Changed
os818376.bin	515	File BIN	5/12/2002
PAGEFILE.SYS	201,326,5...	System file	6/12/2002
PUTTY.RND	600	PGP Random S...	7/5/2002
RunnablePotato.java	392	File JAV	10/26/200
setup.exe	5,666,841	Application	1/7/2002
SETUPLOG.TXT	137,686	Text Document	11/18/200
skeeters.jpg	23,196	JPEG Image	5/18/2002
SUHDLOG.DAT	7,738	File DAT	11/18/200
SYSTEM.1ST	548,896	Notepad	11/18/200
typescript	1,128	File	6/6/2002
VIDEOROM.BIN	32,768	File BIN	11/18/200
winscp.RND	600	PGP Random S...	7/5/2002

/home/faculty/afkjm/public_html

Name	Size	Changed	Rights
..		7/4/2002 10:41 ...	rwxr-xr-x
ai_games		7/2/2001 12:00 ...	rwxr-xr-x
budgetxfer		1/8/2002 3:36 PM	rwxr-xr-x
cgi-bin		11/28/2001 12:...	rwxr-xr-x
conferences		10/8/2001 12:0...	rwxr-xr-x
cs101		5/11/2002 10:1...	rwxr-xr-x
cs201		7/4/2002 10:41 ...	rwxr-xr-x
cs221		5/11/2002 10:1...	rwxr-xr-x
cs403		3/9/2001 12:00 ...	rwxr-xr-x
cs405		5/11/2002 10:1...	rwxr-xr-x
cs411		5/11/2002 10:1...	rwxr-xr-x
cs442		5/11/2002 10:0...	rwxr-xr-x

0 b of 206 Mb in 0 of 82

0 b of 950 Kb in 0 of 56

Copy selected file(s) to local directory

The window on the left shows files on the local Windows computer. Use the icons to select the folder or files you wish. In this case, I've selected teh file named "typescript".

The window on the right shows files on the remote system. Once again, use the icons to explore to find the folder or files you like. Click on the "Copy" button to copy files from one system to another. For example, if I clicked on "Copy" now, I would transfer the file named "typescript" from my local computer to mazzy and put it in the public_html directory.

After you are done transferring files, close the application to log out.

3. The CS Computer Lab

If you happen to be in Anchorage, here is some information about the CS lab itself. If you will always be logging in remotely, you can skip this section (although it may be useful to skim it).

The CS lab is located in Room 170A of the CAS building. The lab is split into "upper" (CAS 170A) and "lower" (CAS 170) labs, and contains approximately 25 PCs between both labs along with a laserprinter. Most of the PCs run either Windows 2000 or Windows NT 4.0, while approximately 5 of them run Linux (a type of Unix). Additionally, a number of servers are kept locked in the network closet. There is a lab technician in room CAS 171 who can help you if you are having trouble with the facility or have general questions. A sheet on the door will list their office hours. If you have programming questions or issues regarding the course, you should ask your instructor and **not** the technician. The preferred way of contacting your friendly lab technicians is to send email to labtech@math.uaa.alaska.edu with your request.

There are many variants of Unix, among them Linux, Ultrix, Solaris, and SunOS. All of them expose an interactive shell that allows you to run a variety of applications. If you log into one of the Linux machines in the lab, then one of the unix programs which runs is the window manager - it controls all of the icons, windows, and mouse clicking that you are used to with a graphical user interface. If you connect to the systems remotely, either from your home or from one of the Windows machines, most of your interaction will be through a text-based command line.

To use either a Windows or Linux machine in the lab, you must first log in. The login procedure requires your *username* and *password*. Your password should be kept secret - never divulge your password to anyone, or those people will be able to access your account files, or even pretend to be you to the internet community. You should have obtained your username and password from your instructor or from the lab technician in room 171. You should change your password to something else, and will learn how to do so in this tutorial.

When you have finished at the computer, it is important for you to log out. If you do not log out, someone else can come into the laboratory and access your files from the computer console. After you have finished you should log out (close all programs and log on as a different user), but you should not turn the computer off. You should **never** turn off a machine in the lab unless performing a reboot or maintenance.

3.1 Logging on and Using the Windows Machines

Once you are logged into Windows, you should see a welcome message logging you into the NT domain. You can just close this window. This indicates that you successfully have logged onto the network.

If you have not yet done so, the first thing you should do is change your password. To change your Windows password, hit control-alt-delete and a menu will appear. Click the "Change Password" button and you will be prompted for your old password and your new password. Make sure not to pick a password that is a simple English word, or something easy to guess. It should contain numbers and non-alpha-numeric characters!

Take a look at the programs in the start menu. You should see putty and WinSCP there, along with other programs you may likely use if you take other CS courses.

3.2 Logging on and using the Linux Machines

The Linux machines in the CS lab are primarily located on the far wall away from the door (**Note**: they will be moving into the lower lab, which is through the doorway from the upper lab). UNIX is a multi-user operating system which means that many people can use the same computer at the same time. These users may be located in different places, and can remotely access the machine through the Internet. Note that this is different from the typical Macintosh or PC, which normally only serves one user sitting at the console. Most of your work will be done on a single Unix server, mazzy.math.uaa.alaska.edu.

If you are familiar with Windows for the PC, you are already familiar with the window manager used on the Linux workstations, Gnome/GNUStep. You can resize windows by clicking and dragging from the corner of a window, and you can also use the menus located at the top of every window (File, Edit, etc). Similarly, to bring a window to the foreground (i.e. activate it out of your windows) you can click on it.

Try logging in on one of the Linux boxes using the password given to you. This will be the same password that you use for mazzy.math.uaa.alaska.edu. However, if you change your passwords, you can have a Unix password that is different from your Windows password.

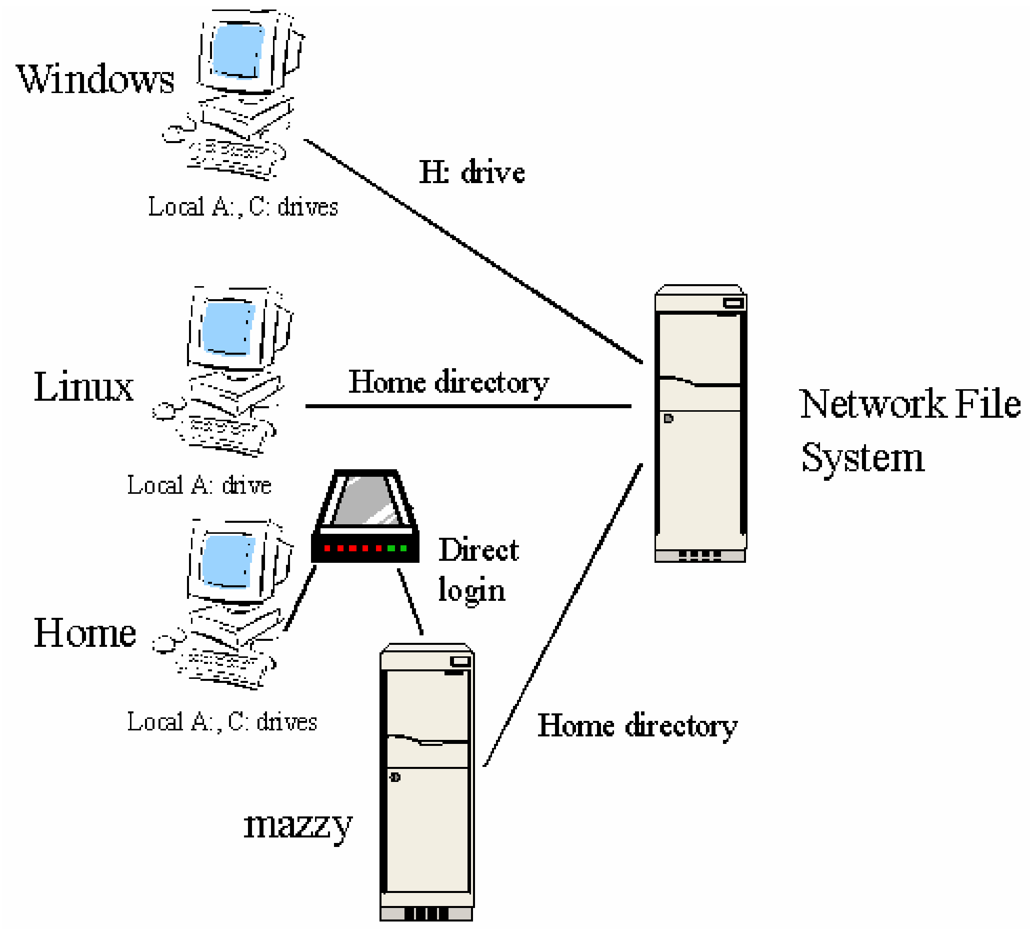
Once you are logged in, you can access a menu by clicking on the icon of a foot in the lower left corner. This will display different programs that you can run. One of the most common programs to run is to create a new X terminal window, or xterm for short (named after the graphical window system X). There are several options for opening up xterm windows under the Utilities and Xshells menu selections. There should also be an icon you can click on the menu bar to directly create an xterm.

This will create a new window where you will be able to enter unix commands, enter your programs, compile your programs, and run your programs. In short, almost all of the work you do will be through the Xterm. You may wish to create more than one Xterm window and switch back and forth among your windows. It is often useful to have one Xterm where you will be editing your files, another where you will run them, and perhaps even another where you can issue unix commands, allowing you to easily switch back and forth between different tasks. You may also wish to resize your window if you prefer a larger workspace.

At this point, you may enter unix commands into the Xterm window. Common commands will be described in the next sections. When you are finished with the computer, remember to log out!

3.3 CS Lab File System

The CS lab employs a networked file system. No matter what machine you log into, you have access to the same data files on the network. The figure below depicts the networking setup:



The network file system is capable of hosting your files no matter where you log in. If you log in on Linux or another CS unix server (like mazzy), then your home directory will automatically be set to the networked file system. This means that if you log into mazzy and create a file, you can log in later to a different Linux or Windows machine and access the same file. Consequently, any files you save will be accessible from any other system that shares the network.

If you log in on a Windows machine, by default you will have access to the local hard disk, the C: drive. If you wish to access the files stored on the network file system, they are accessible through a network drive mapped to the H: drive. If you need to move files back and forth between Unix and windows, the drive mapping is a great way to do it. Since you have your own directory on the network file system, this is a good place to store any of your files as opposed to storing them on the local C: drive. If you store files on the C: drive, anyone else that logs in on that machine will be able to access them. Consequently, you should only use the C: drive for temporary storage, and be sure to copy off and then delete anything you may put on the local C: drive.

4. Log into unix

Use putty to log into mazzy as described in section 2.

After you are logged in, the login prompt will show you the name of the machine you are using followed by an angle bracket, e.g. *mazzy.math.uaa.alaska.edu>* At this point unix is waiting for you to enter commands.

If you have not already done so, the first command you should issue is **yppasswd** to change your password.

To do this enter yppasswd as shown below:

```
mazzy.math.uaa.alaska.edu> yppasswd        You only type the part in bold,
                                           the rest is printed out
                                           by the computer.
Changing NIS password for yourlogin
Old NIS password:          <Type password here>
New password:              <Type new password here>
Retype new password:       <Type new password here again to verify>
```

Be sure to pick a password that you can remember but that nobody else will guess. After you have entered your new password, you will be asked to enter it again in case you made a typo (you won't be able to see what you are typing whenever entering passwords). If you ever forget your password, the consultant in the lab can help you set a new password, but this may take some time to take effect.

If you are using the CS Lab and logging in from one of the Windows machines, note that the unix password is separate from your windows password, so it is possible to have a different windows login password from your unix login password.

5. File Operations

Somewhat similar to MS-DOS, UNIX has a hierarchical file structure where you may change directories, copy files, delete files, etc. In Unix, the files are organized into a tree structure with a root named by the character '/'. There are four types of files in the file system: (1) An ordinary file contains a program or data, (2) a directory contains name and address information of the files in the directory, (3) a device file acts as a gateway between physical devices by representing the device (e.g. printer, speaker) as a file, and finally (4) link files are pointers to other files. In this course you will primarily be concerned with ordinary files and directories.

An example of a directory tree is shown below:

```
                 /
  _______________|_______________
  |       |              |       |
 etc     bin            tmp     dev
         _|_                    _|_
        |   |                  |   |
       ls ... cp             ucb .. lib
```

To view a listing of all files in the current directory, use the **ls** command (try these commands yourself, listed are samples from my own directory so you will get different results):

```
mazzy.math.uaa.alaska.edu> ls
GNUstep              MyProjects
bin                  cs101misc
class                public_html
```

To see what directory you are currently in, use the **pwd** command (print working directory):

```
mazzy.math.uaa.alaska.edu> pwd
/home/student/afkjm          (Your username/directory will be shown)
```

To change to a new directory, use the cd command:

```
mazzy.math.uaa.alaska.edu>  cd /usr/bin
mazzy.math.uaa.alaska.edu>  ls
(lots of files shown here)
info                   webpng
install-sid            wftopfa
ispell                 wrjpgcom
java                   xemacs
javashark              xemacs-20.4
jcf-dump
```

In this example, we've switched directories in the hierarchy. This directory contains several files, most of which happen to be directories containing utility programs.

Three common special symbols for directories are "..", "." and "~". ".." refers to the directory one level up in the hierarchy, "." refers to the current working directory, and "~" refers to your home directory (the directory you start out in when you first log in, and where your files are stored):

```
mazzy.math.uaa.alaska.edu>  pwd
/home/student/afkjm                                Current directory
mazzy.math.uaa.alaska.edu>  cd ..
mazzy.math.uaa.alaska.edu>  pwd
/home/student                                      Up one level
mazzy.math.uaa.alaska.edu>  cd ~
mazzy.math.uaa.alaska.edu>  pwd
/home/student/afkjm                                Home directory
```

To create your own directory, use the **mkdir** command. First, make sure that you are in your home directory:

```
mazzy.math.uaa.alaska.edu> cd     Changes to your home directory, same as cd ~
mazzy.math.uaa.alaska.edu> mkdir cs101
mazzy.math.uaa.alaska.edu> cd cs101
mazzy.math.uaa.alaska.edu> pwd
/your_path/username/cs101
```

You've just created a directory called cs101 in your home directory. If you wish you can create other directories to organize your files, e.g., you may want a directory for all Java programs, one for personal files, one for mail, another for programs, etc.

Try changing your current directory into my directory where I stored some cs101 files. You should see three files there:

```
mazzy.math.uaa.alaska.edu> cd ~afkjm/cs101misc  Changes into MY directory
mazzy.math.uaa.alaska.edu> ls
test_prayer      tricky_prof      FirstProgram.java
```

These files are text files. To view text files, you can use **cat** or **more**:

```
mazzy.math.uaa.alaska.edu> cat tricky_prof
(file will scroll by very fast)
mazzy.math.uaa.alaska.edu> more tricky_prof
(file will be displayed again, press spacebar for the next page)
```

To copy files, use the **cp** command. The format for cp is: *cp sourcefile destinationfile*. For example, the command "**cp file1 file2**" will create a copy of file1 named file2 in the current working directory. Try the following to copy the file "headlines" into your cs101 directory.

```
mazzy.math.uaa.alaska.edu> pwd
/home/student/afkjm/cs101misc           Make sure you're in my cs101 directory
mazzy.math.uaa.alaska.edu> cp tricky_prof ~/cs101          Copy file to your
                                                           cs101 directory
```

Change back to your cs101 directory and you should now see the file there:

```
mazzy.math.uaa.alaska.edu> cd ~/cs101
mazzy.math.uaa.alaska.edu> ls
tricky_prof
```

View the tricky_prof file to make sure it copied correctly. When you are satisfied that the cp command works to copy files, you will probably want to remove it. You can remove files with the rm command.

```
mazzy.math.uaa.alaska.edu> rm tricky_prof
mazzy.math.uaa.alaska.edu> ls
(tricky prof file deleted)
```

Be very careful when using the "**rm**" command! After a file has been deleted, it cannot be recovered! Consequently, you should make sure that you don't want the files you are deleting anymore when you remove them.

Some other useful commands you may want to experiment with:

mv file1 file2	Rename file1 to file2
rmdir dirname	Delete directory named "dirname"
cd	Change back to your home directory
who or **w**	See who is online
less filename	View filename, more powerful than more

diff file1 file2	Find differences between file1 and file2
g++ filename	Compile file filename using the C++ compiler
javac filename	Compile file filename using the JAVA compiler
java filename	Execute the JAVA-compiled program filename
pico filename	Edit filename with the pico editor
vi filename	Edit filename with the VI editor
joe filename	Edit filename with the JOE editor
emacs filename	Edit filename with the EMACS editor
enscript filename	Prints a text file to the laserprinter
man command	Show manual page for command
script	Begin saving what goes to the screen
exit	End a script or log out of a Xterm

To get online help for unix commands, you can use the **man** command, short for manual. All of the information in this handout is easily found online at your fingertips whenever you are logged in. The format is: **man command** which will show the online manual command for the command specified. For example, if you type "**man ls**" or "**man man**" you will be shown the commands for the ls and the man commands.

6. Introduction to pico

This section will introduce you to the text editor, pico. There are many other text editors for Unix; **vi, joe** and **emacs** are popular ones, and there is also the **notepad** on your desktop. For example, you could use notepad in Windows to access your unix files using the networked H: drive in the lab, or save a file on your local hard drive and transfer it remotely using WinSCP. The following is an introduction to the pico editor (pico is an acronym for PIne COmposer developed at the University of Washington), an explanation of how to use the editor, and an exercise for you to do using the editor.

Pico is an interactive, screen-oriented text editor. You will use pico to enter programs and modify existing programs. While relatively simple, pico may take a little getting used to if you've only used graphical word processing programs because everything must be done via keyboard commands. This tutorial presents the basics of pico and the commands which you will most often use.

Pico is not a word processor. It resembles one, in that it lets you type in and modify text. However, the way it lets you do this is quite different from traditional word processors. Try to forget about what you know about word processors and do not make any assumptions about pico.

6.1 Pico commands, basics

The editing commands are displayed at the bottom of the screen and are invoked using control-key combinations. In the display, the ^ character is used to represent the Ctrl key.

You make a choice by holding down the Ctrl key and pressing a letter key. For example, ^G means you hold down the Ctrl key and press the G key. Some commands prompt you for information. These prompts will be displayed at the bottom of your editing window, just above the command labels.

Starting Pico

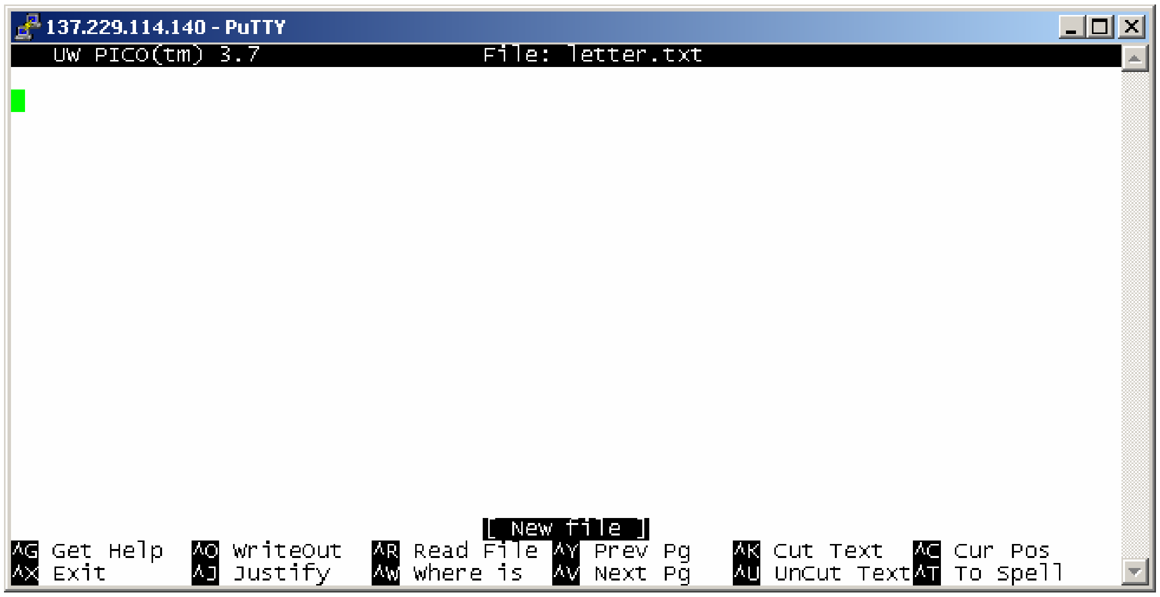
At your Unix shell prompt, type:

pico filename

replacing filename with the name of the file you want to create or edit. For example, to create a file and name it letter.txt, type

pico letter.txt

If the file already exits, Pico opens it for you to edit. If it doesn't exist yet, Pico creates it and places you in an editing buffer. If you start pico without giving a filename, you must provide one when you exit the program. Here is the window you should see if you started pico to edit letter.txt:



```
  UW PICO(tm) 3.7                    File: letter.txt

                                   [ New file ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Pg   ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg   ^U UnCut Text^T To Spell
```

Lines that continue beyond the edge of the display are indicated by a $ character at the end of the line. Long lines are scrolled horizontally as the cursor moves through them.

Pico displays a menu bar of commonly-used commands at the bottom of the screen. Pico accepts commands from your keyboard but not from your mouse.

To insert text into your Pico editing screen at the cursor, just begin typing. Pico inserts the text to the left of the cursor, moving any existing text along to the right. Each time the cursor reaches the end of a line, Pico's word wrap feature automatically moves it to the beginning of the next line. (Also see "Justify.")

To move the cursor, use the arrow keys or use ^f (forward), ^b (back), ^n (next line), ^p (previous line).

To delete the character to the left of the cursor, press BACKSPACE, DELETE, or ^h. To delete the character highlighted by the cursor, press ^d. To delete the current line, press ^k.

To save your changes part way through your editing session, use the command ^O (WriteOut) and a message similar to the following will be displayed near the bottom of your screen:

```
File Name to write : letter.text
```

Press Return to save your changes. You will be left inside the editor. At this point you can exit your editing session or continue with more changes.

To exit pico, use the command ^X (Exit). If you have not made changes to the file or if you have already saved your changes, you will be returned to your Unix prompt. If changes have been made but not saved, you will be prompted with the following:

```
Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) (y/n)?
```

If you respond with y the name of the file you are editing will be displayed. For the above example, you will see the following:

```
File Name to write : letter.text
```

Press Return and then you will have completed your editing session and be returned to your Unix prompt.

If you wish to change the name of the file, when the current name is displayed you can supply a new name by typing over the given name. If you give the name of an existing file you will warned:

```
File "memo.july" exists, OVERWRITE? [n] :
```

Deleting and moving lines

The command ^K (Cut Text) will delete the entire line that the cursor is currently on. You can then move the cursor to another position and use ^U (UnCut Text) to insert the previously deleted line of text at the current position. If a group of lines is deleted successively using the ^K (Cut Text) repeatedly with no other commands in between, they can be pasted back into the text at the current cursor postion, using ^U (UnCut Text).

When you use ^K (Cut Text) a message is displayed as a reminder than it is also possible to mark blocks of text for deletion or moving.

    Line Deleted. (Could also use ctrl-^ to mark text for cutting)

To mark blocks of text, use Ctrl ^ then move the cursor to the end of the text you are blocking (the marked text will be highlighted), and finally, use ^K (Cut Text).
NOTE: The Ctrl ^ command may not work with all Telnet programs. For example, using the Telnet client provided with Windows, the Ctrl ^ command does not work.

Paragraph justification

The command ^J (Justify) is used for paragraph justification. The paragraph that contains the cursor will be justified, or, if the cursor is between lines, the paragraph immediately below will be justified. Paragraphs are separated by blank lines, or by lines beginning with a space or a tab. Unjustification can be done immediately after justification by using ^U (UnJustify).

Searches

To see what line number you are on, use ^C (Current Position).

To search for a word or partial word, use ^W (Where is) and you will be asked to supply the search string. String searches are not case-sensitive. A search begins at the current cursor position and wraps around the end of the text.

File browser

The file browser is offered as an option in the ^R (Read File) and ^O (Write Out) command prompts. The option is labelled ^T (to Files). It is intended to help in searching for specific files and navigating your directories.
The browser displays file names with sizes and directory names. The browser always starts with your home directory and not with your current working directory. The current directory is displayed on the top line of the display while the list of available commands is displayed at the bottom of your screen. Note that these commands do not need to use the Ctrl key. Several basic file manipulation functions are supported: file renaming, copying, and deletion.

If you get disconnected

If you are disconnected while running pico, your current work may be saved before exiting. The work will be saved in the current filename with .save appended. If you have not yet named the file you were editing, it will be saved with the name pico.save.

Getting Help

More specific help is available in pico's on-line help which is provided by context-sensitive help screens. The command ^G (Get Help) is used to invoke the help system. A Unix manual page for pico is also available by typing man pico at your Unix prompt.

Options

Below are some options that are available by invoking the program as:

```
        pico option filename
```

+n	Causes pico to be started with the cursor located n lines into the file. (Note: no space betwen "+" sign and number)
-e	Enable file name completion.
-j	Enable "Goto" command in the file browser.
-g	Enable "Show Cursor" mode in file browser.
-k	Causes "Cut Text" command to remove characters from the cursor position to the end of the line rather than remove the entire line.
-rn	Sets column used to limit the "Justify" command's right margin
-v	View the file only (editing is not allowed).
-w	Disable word wrap, allows editing long lines.
-z	Enable Ctrl Z suspension of pico.

Command Overview

^a	Move to the beginning of the current line.
^e	Move to the end of the current line.
^v	Move forward one page.
^y	Move backward one page.
^w	Search for text (whereis).
^L	Redraw a garbled screen.
^d	Delete the current character.
^^	Begin selecting text. *
^k	Remove (cut) current line or selected text.
^u	Paste (uncut) last cut text at the cursor position.
^j	Format (justify) the current paragraph.
^t	Spell check the text.
^r	Insert (read in) a file into this file.

^o Save (output) the file.
^g View Pico's online help.
^x Exit Pico, saving the file.

6.2 An exercise using pico

You will now work with pico. Begin by typing a document: a letter to a friend. To begin, make sure you are in your home directory and then type:

```
        mazzy.math.uaa.alaska.edu> pico letter
```

You should now have a blank screen You are now free to type. Begin the letter, now. Make the letter brief, two or three sentences. The letter should contain your name and course number. Experiment with the commands in the previous section. Move the cursor around, jump to different lines, delete and insert text. When you are finished, write out and save your letter before quitting.

7. Printing your files

From time to time you may wish to print out files you are working on with mazzy. If you are not working in the lab itself, the way to do this is to download the file to your local computer using a program like WinSCP. Once the file is on your local computer, you can load it and print it locally using a word processing program (e.g. WordPad or NotePad).

If you are located in the CS lab itself, you can print files using the enscript command:

```
        mazzy.math.uaa.alaska.edu> enscript  –2r  –i 3  –Ecpp  letter
                        prints file named 'letter'  onto the laserprinter
                        Note the placement of spaces!  There is a space between i and 3.
```

You should use enscript to print out copies of all text files, including your source code or anything else you create or can edit.   By using the options "-2r –i 3 –Ecpp" the enscript program will save paper by condensing 2 pages onto one single page and printing in landscape mode. The "cpp" option will also format the printing so that it is easier to read C++ (and Java) code.   The "-i 3" command indents the text by 3 characters. There are other commands to print files, such as the **lpr** command, but I prefer the enscript format given above for purposes of readability and to save paper.

It is worth noting that entering the enscript command from unix will always print on the laserprinter located in the CS lab. This means that if you are logged in from somewhere else, for example, from home via modem, then if you enter the enscript command your output will print in the lab, not on your printer at home

8. Email

For email, you should have access to mail on the web (webmail.uaa.alaska.edu – see ITS in CAS120 for help). Alternately, there is also a very user-friendly software package called PINE on the Unix systems. This email software features on-screen menus and help; usually you need little or no assistance to use pine. To run the package, simply type **pine**. Spend a few minutes examining the various options. The most important ones for now are 'c' to compose a new mail message, and '^x' (control-x) to send a message you've composed. To include attachments, use ^j from the new message screen to select the file(s) to attach (for example, to turn in homework).

If you prefer to use a different email client, you can transfer any files to your local computer using WinSCP and then attach the files as you normally would.

9. Compiling a Java Program on Unix

This section will show you how to compile a Java program, introduce you to debugging, and how to run a program. Don't worry that you won't really know how the program works yet. This is just to introduce you to the process and give you an idea of some of the things we'll do in the class. First, start by copying the file FirstProgram.java into your cs101 directory (create this directory if it does not already exist):

```
mazzy.math.uaa.alaska.edu> cp ~afkjm/cs101misc/FirstProgram.java ~/cs101
mazzy.math.uaa.alaska.edu> cd ~/cs101
```

Make sure the file *FirstProgram.java* is in your current working directory. Before a program can run, it must be *compiled*. Compilation is the process of turning the program into code that the machine is able to understand. To compile the program, use the **javac** command. Be careful with the upper and lower case! Java is case-sensitive. You should get the following output:

```
mazzy.math.uaa.alaska.edu>  javac FirstProgram.java

FirstProgram.java:21: ';' expected
    System.out.println("Enter your name:")
                                          ^
1 error
```

There are errors! As you write programs, you will encounter errors like these frequently. To fix this program you must edit it and correct the error. Type:

```
mazzy.math.uaa.alaska.edu>   pico FirstProgram.java
```

Once again, don't be concerned about how the program works. You don't have to understand how it is working yet, although you will get a better idea of how this works by the end of the course. The compiler has informed you that the error is on line 21. Note that the number reported by the compiler is not always the actual line number with the error. However, in this case the error actually is on line 21. Referring to the commands in the introduction to pico, recall that ^C will tell you what line number you are on. Use the arrow keys to scroll down to line 21, checking which line number you are on using ^C.

Move to the end of line 21 using the ^E command and insert a ";" at the very end. The entire line should read:

```
System.out.println("Enter your name:");
```

Now save and exit the file via ^O and ^X. Recompile the program again:

```
mazzy.math.uaa.alaska.edu> javac FirstProgram.java
```

If you receive no error messages, then the compilation was successful. Compiling produces a file named **FirstProgram.class**. To run the program, type **java FirstProgram** (without the "class" part):

```
mazzy.math.uaa.alaska.edu> java FirstProgram
Enter your name:
naomi did i moan

Your name backwards is:

naom i did imoan
```

More instructions on compile and also on creating programs that run through a web browser (web applets) will be given later.

10. Producing a Typescript File

In turning in your Unix assignments, it will be helpful to make a "recording" of your programs output. While this is not required for all assignments, I appreciate receiving the output as it helps speed along the grading process!

Unix contains a command that allows you to create one file containing both the source code and the execution of your program. This command is **script**. The script command operates similarly to the record command on a VCR. When you record, the broadcast is captured on tape until the recording is terminated. When you issue the script command, everything that is displayed on your unix screen is captured in a file called *'typescript'* until you terminate the recording using the **exit** command. Once you exit, you can view your recording (the file called typescript) on the monitor and send it to the printer.

For this course the steps to follow:

1. Make sure everything works on the Unix end before executing the script command. If you are turning in a program, make sure the program runs. If you are turning in a text file you created, make sure the file is created first.

2. Once everything is working and ready to be turned in, type **script** to begin creating the typescript file.

3. Display the files you created. For example, enter **cat FirstProgram.java** to display the contents of the java program and save it in the typescript file. It will scroll quickly but it is all being captured in the typescript.

4. Compile and run any programs you created. Once again, all the commands and the output are being recorded in the file.

5. When finished, type **exit** . This stops logging to the typescript file. If you execute the **ls** command, it will now show that there is a new file named typescript created in the directory.

6. Type **more typescript** to view your file to make sure it is okay. If you backspaced, those characters might not display properly.

7. Turn your completed homework in by attaching the typescript file to an email!