CS 201 – Lab #1
Introduction to Unix and the UAA Computing Lab

1. Introduction

This first lab is a little different than the rest of the labs. It primarily serves as a tutorial to orient you to the basics of using Unix and the computing facilities in the CS computer lab. Work through the tutorial and at the end there are some questions to turn in. The tutorial will assume that you will be logging in using a Windows-based machine in the Computer Science lab located in SSB 170A although it is also possible to log in remotely from your home computer if you install the needed software. Future labs will consist primarily of writing Java code.

2. The CS Computer Lab

The CS lab is located in Room 170A of the SSB building. The lab is split into "upper" (SSB 170A at the top of the ramp) and "lower" (SSB 170 at the bottom of the ramp) labs, and contains approximately 30 PCs between both labs along with printers and a scanner. You are free to use any of these machines in either lab, except for the machines in the far northeast corner of the lower lab, which are reserved for special projects.

All of the PC's in the lab run Windows XP, while some of them are set up to dual-boot into Linux (a type of Unix). Additionally, a number of servers are kept locked in the network closet. This class will assume that you are working on a Windows machine.

There is a lab technician in room SSB 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. To complicate

matters, at this point you will be given two separate passwords: one for logging into Windows, and another for logging into the Unix systems.

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.

2.1 Logging on and Using Windows

To log into the Windows XP machines your login is set to your UAA email username, for example "askmp". If you do not know your username then you can look it up at the following ITS webpage: http://technology.uaa.alaska.edu/computer/ID/

Your default password is “uaa” followed by your first, middle and last initial in uppercase, followed by the last four digits of your student ID (i.e. the number on the back of your wolfcard, although sometimes it may be your Social Security Number).

The domain should read "UAAACTDIR"

Contact the lab administrator if you have problems logging into the Windows machines.

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. Connecting to Unix via Windows

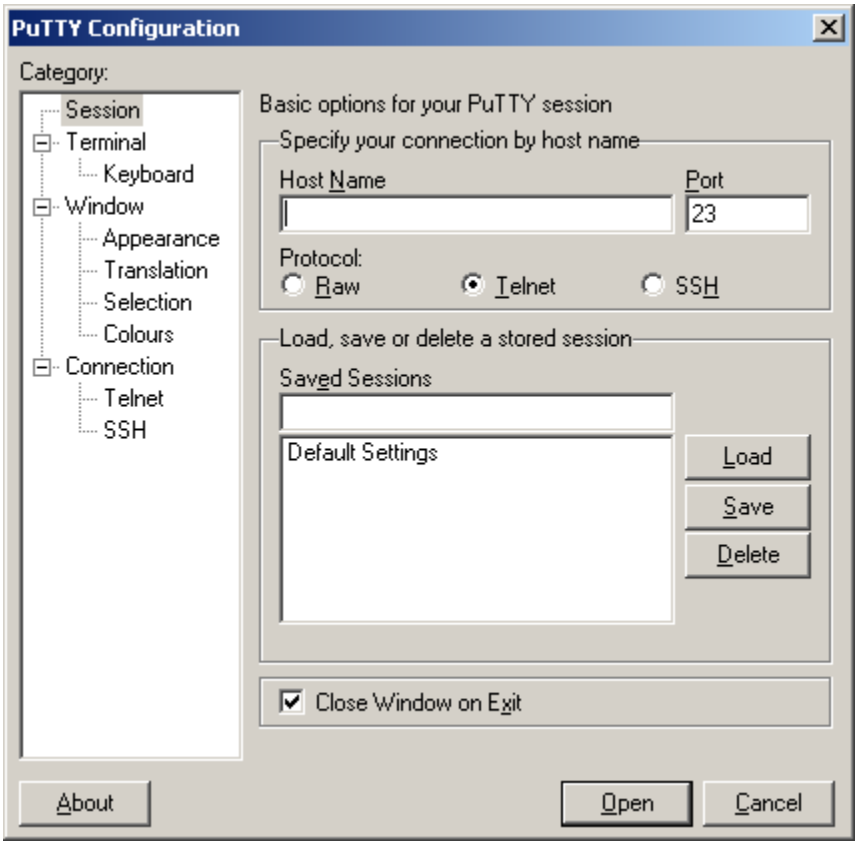
Some of the Java programming will be done on a Unix machine named **bigmazzy.math.uaa.alaska.edu** located in the computer science lab server closet. The unix username is the same as your UAA webmail username. However, your password defaults to your student ID number. Contact your instructor if you are unable to log into **bigmazzy**.

3.1 Using Putty

To connect to **bigmazzy**, 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 logged in on one of the Windows machine, look for "putty" from the start menu. Its icon looks like the following:

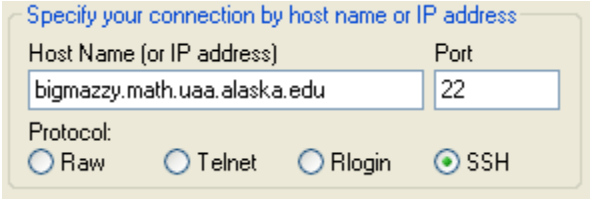
Upon starting up the program, by default the program will assume you are using telnet and will ask you to enter a host name:



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

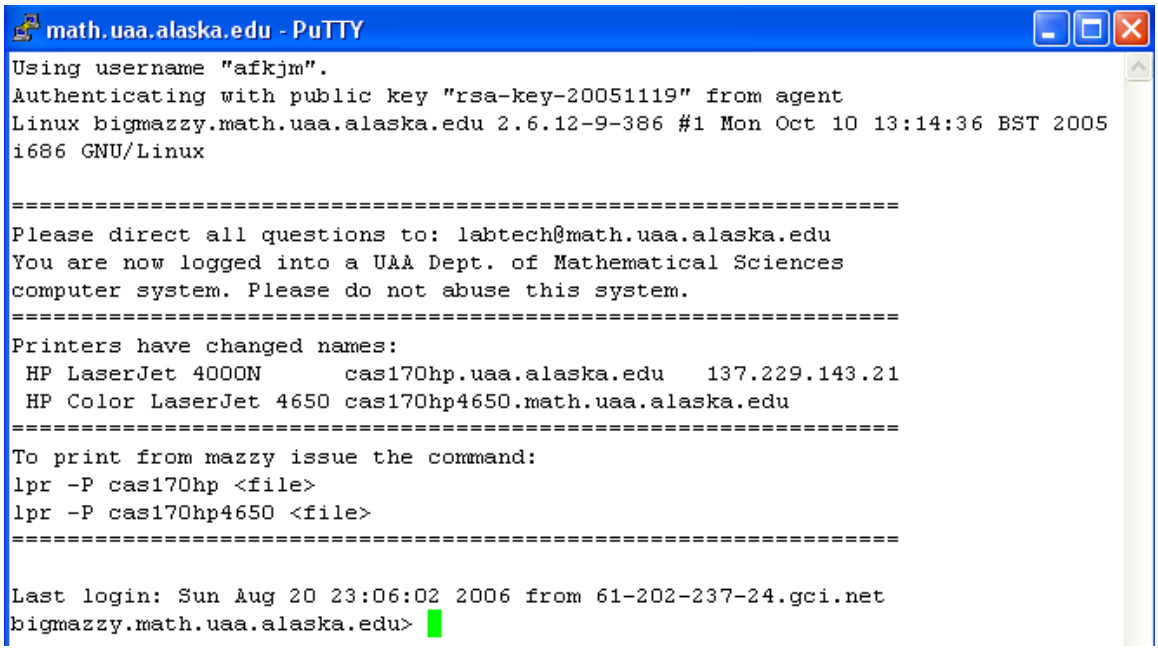
bigmazzy.math.uaa.alaska.edu

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 knoya. You will be prompted for your UNIX username and password:



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.

If you have not received it, contact your instructor to get it! If your instructor does not have it, then you will need to see the lab techs to obtain or reset your account.

3.1.1 Using putty from home

If you would like to use PuTTY from your home machine to connect to bigmazzy you can download it free from the link on the course webpage.

You only need the file **putty.exe**. The program is simple to install – just download it and run it directly.

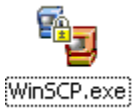
3.2 Using WinSCP

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.

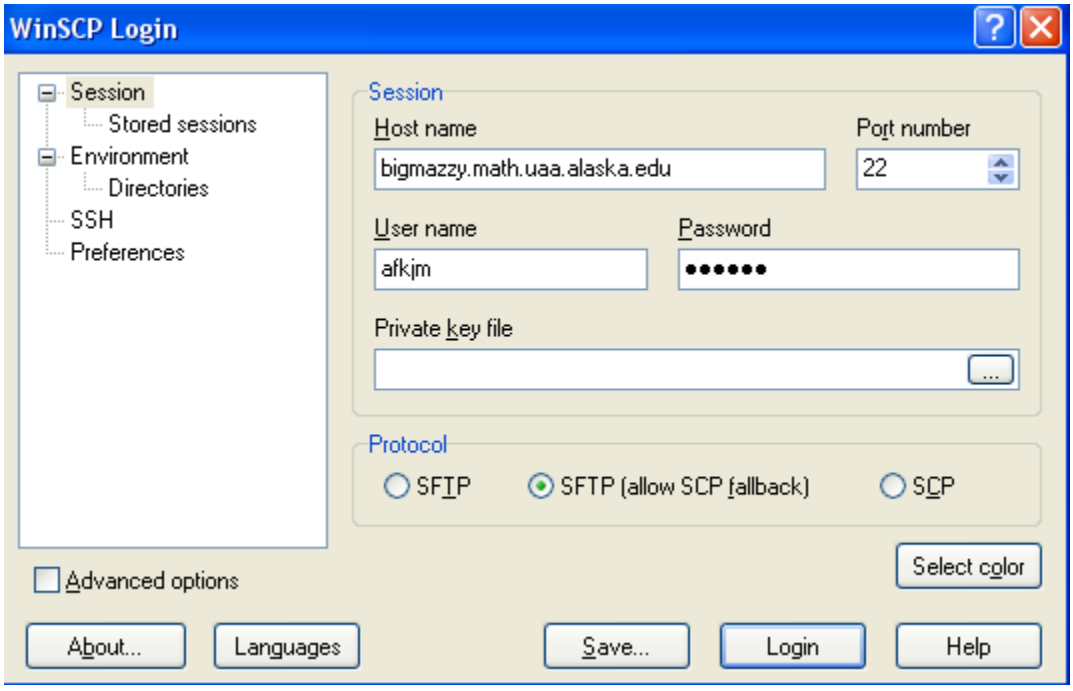
This program will be useful when you want to transfer files from one computer to another over the Internet. For example, you might develop files on your computer at home and want to upload them to knoya. To do so, the easiest way is to use scp to copy the files.

With a Windows system, the program I recommend you use is WinSCP. This is another free program and it is available online from the link on the main course web page.

Download the file winscp.exe if you wish to run this program from your home machine. It is already installed on the CS Lab machines (and is actually not needed due to the H: drive mapping).

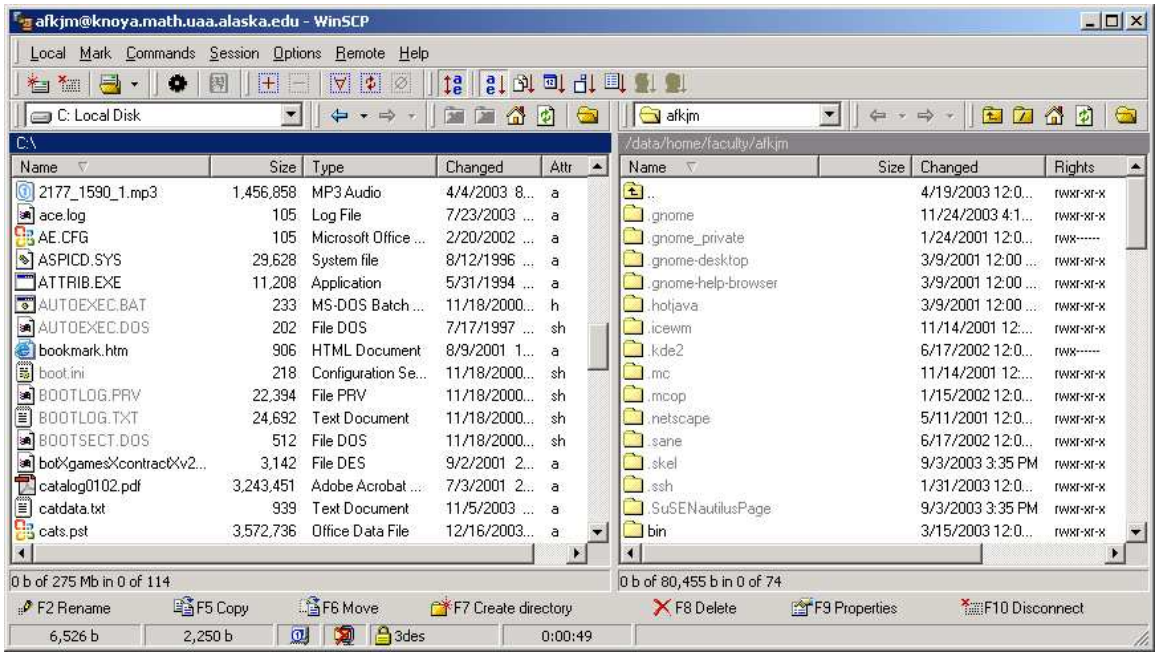


Upon running WinSCP, you'll be prompted for the machine to connect to:



Enter bigmazzy.math.uaa.alaska.edu for the host name to transfer files from your Windows machine to bigmazzy. 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:



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 bigmazzy and put it in the public_html directory.

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

4. Changing Your Unix Password

Use putty to log into bigmazzy as described in section 3.

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. *bigmazzy.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 **passwd** to change your password.

To do this enter passwd as shown below:

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

If the passwd command doesn't work for your account, try running **yppasswd** instead.

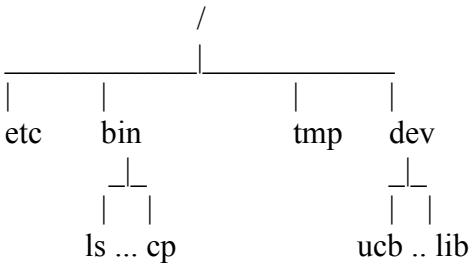
Be sure to pick a password that you can remember but that nobody else will guess. After you have entered your new password, you will be asked to enter it again in case you made a typo (you won't be able to see what you are typing whenever entering passwords). If you ever forget your password, the consultant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

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:



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):

```
bigmazzy> ls
GNUstep             MyProjects
bin                 cs201misc
class               public_html
```

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

```
bigmazzy > pwd
/home/faculty/afkjm          (Your username/directory will be shown)
```

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

```
bigmazzy > cd   /usr/bin
bigmazzy > 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):

```
bigmazzy > pwd
/home/faculty/afkjm                    Current directory
bigmazzy > cd  ..
bigmazzy > pwd
/home/faculty                          Up one level
bigmazzy > cd  ~
bigmazzy > pwd
/home/faculty/afkjm                    Home directory
```

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

```
bigmazzy > cd     Changes to your home directory, same as cd ~
bigmazzy > mkdir  cs201
```

bigmazzy > **cd cs201**
bigmazzy > **pwd**
/your_path/username/cs201

You've just created a directory called cs201 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 cs201 files. You should see three files there:

bigmazzy > **cd ~afkjm/cs201misc** *Changes into MY directory*
bigmazzy > **ls**
test_prayer tricky_prof FirstProgram.java

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

bigmazzy > **cat tricky_prof**
(file will scroll by very fast)
bigmazzy > **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 "tricky_prof" into your cs201 directory.

bigmazzy > **pwd**
/home/faculty/afkjm/cs201misc *Make sure you're in my cs201 directory*
bigmazzy > **cp tricky_prof ~/cs201** *Copy file to your cs201 directory*

Change back to your cs201 directory and you should now see the file there:

bigmazzy > **cd ~/cs201**
bigmazzy > **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.

bigmazzy > **rm tricky_prof**
bigmazzy > **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
emacs filename	Edit filename with the EMACS editor
lpr –P cas170hp 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. 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

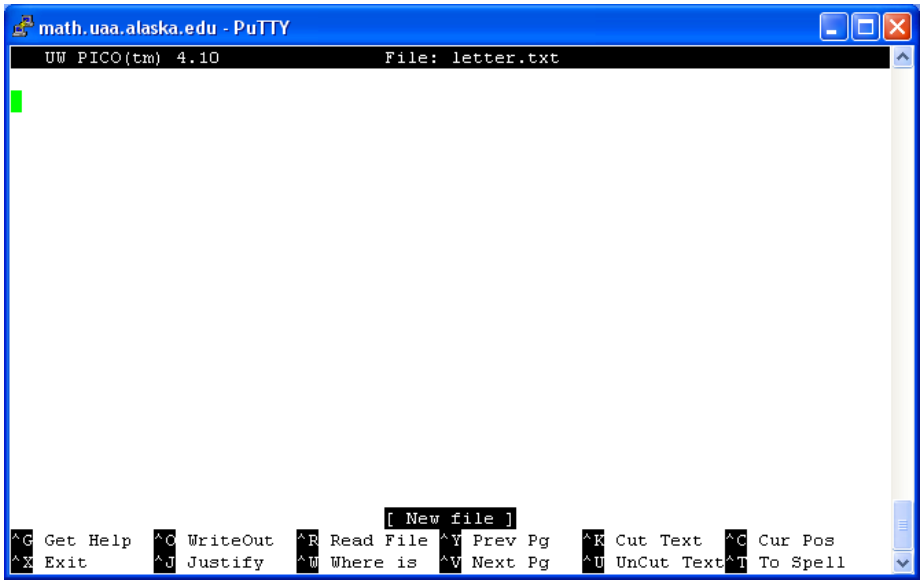
At your Unix shell prompt, type:

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:



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 under 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:

bigmazzy.math.uaa.alaska.edu> pico **letter.txt**

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 bigmazzy. For most of you, the best 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 **lpr** command:

bigmazzy> **lpr -P cas170hp4650 letter.txt**
prints file named 'letter.txt' onto the laserprinter named cas170hp4650. Note the placement of spaces!

It is worth noting that entering the lpr 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 DSL or modem, then if you enter the lpr 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 (http://webmail.uaa.alaska.edu – see ITS in SSB120 for help).

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 cs201 directory (create this directory if it does not already exist):

```
bigmazzy> cp ~afkjm/cs201misc/FirstProgram.java ~/cs201
bigmazzy> cd ~/cs201
```

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:

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

FirstProgram.java:20: ';' expected
     s = scan.nextLine();
                        ^
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:

```
bigmazzy >   pico FirstProgram.java
```

Once again, don't be concerned about how the program works 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 20. Note that the number reported by the compiler is not always the actual line number with the error. In this case the error actually is on line 19 – pretty close to the line with the error reported by the compiler but not the same! Different compilers and programs may or may not give you the exact location of errors, so you'll have to rely on your programming skills when it comes to debugging.

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 19 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:

bigmazzy> **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):

bigmazzy > **java FirstProgram**
Enter your name:
naomi did i moan

Your name backwards is:

naom i did imoan

Later we'll use an IDE to do most of our programming, but it's a good idea to see how to directly compile and run your programs for now.

Exercises

Save the answers to these exercises into a text file or word processing document and submit your answers via blackboard under the lab #1 assignment

1. View the file in ~afkjm/cs201misc named "tricky_prof". What was the 95 point final exam question?

2. Write a short letter using pico (or some other unix editor such as vi). If you did the exercise in the section on using pico, this can be the same letter.

3. What is the difference between the commands:

ls ~afkjm and ls afkjm

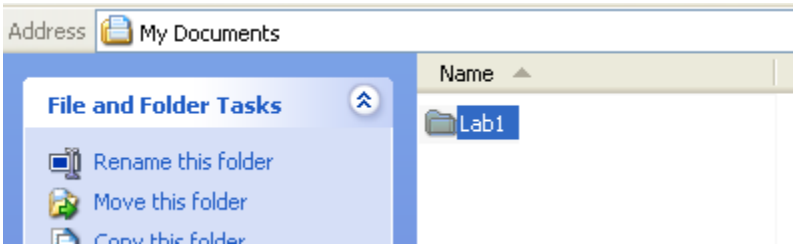
4. Give the unix commands to (1) give a listing of all files in a directory, (2) display the contents of a file all at once, (3) print source code of a file (hypothetical) named "foo.java" on the CS lab laserprinter from unix.

To Turn In

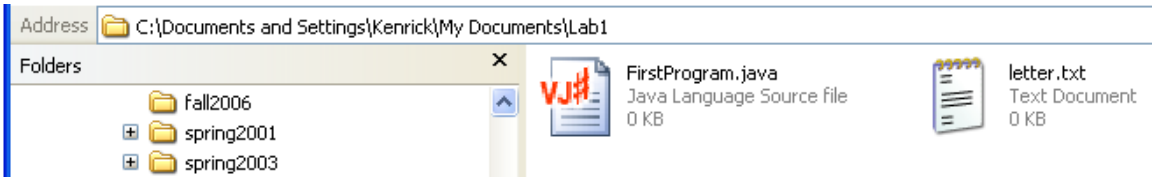
Normally you won't have to turn anything in for your labs. I'll just see that you completed the assignment and mark you down right there in the lab (unless you're working from home).

However, to make sure that you know how to submit your work via Blackboard, I ask you to submit some files for this lab as practice. This portion of the lab assumes that you are logged in on a Windows machine.

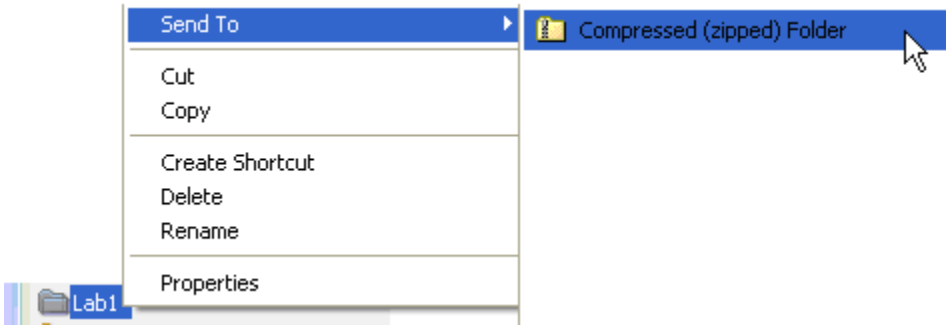
1. First make a folder that will contain all the files you will submit. You have permission to create folders in your **My Documents** folder. For example I made a folder here called **Lab1**:



2. Copy your working file **FirstProgram.java** and your **letter.txt** from bigmazzy to your Windows machine using **winscp**. Copy them into your **Lab1** folder:



3. The easiest way to package these files together is to right-click on the **Lab1** folder and select "Send To --> Compress (zipped) Folder" (this only works on Windows XP, if you are using another OS then you'll need WinZip or another compression program):



4. This will create a Zip compressed file called **Lab1.zip** that will contain all of the selected files. The default name is the same name as your folder, with a .ZIP extension (e.g. Lab1.zip in my example).
5. Send the zip file you created to me by logging into blackboard and submitting it under Assignments. Go to http://uaaonline.alaska.edu and log in using your UAA account name and password.
    a. Click on "Submit Assignments"
    b. Select "View/Complete Assignment" for Lab #1
    c. Under "Comment" type in something about your lab (e.g. "Here's lab #1")
    d. Click "Browse" and navigate and select the zip file you created
    e. Click Submit
6. You are done! You will submit your assignments in the same manner. Note that your submission will be time stamped and you can come back later and see your grade. For future labs, you will not need to send your labs in Blackboard, I just want to check this first one to make sure it went through, the proper files were sent, and that you know how to complete the submission process. All of your homeworks should be submitted this way.