

**FLCC AXP Linux System
Users Guide
Fall 2002**

Table of Contents

Topic	Page
Introduction	1
Logging in and out	2
Connecting to the AXP on the Internet	3
Commonly Used Shell Commands	4
Linux File Conventions	13
Transferring and Printing Files	14
Printing Linux Labs	15
Linux Network Services	16
Troubleshooting Problems	17
System Help	18
Quick Reference Guide	19
VI Editor Reference	20
Insert Mode	20
Command Mode	21
: Command Line in Command Mode	22

Introduction to FLCC's AXP Computer

Serving as FLCC's main academic computing facility is a minicomputer manufactured by Compaq Services formerly known as **Digital Equipment Corporation**. This minicomputer is an **Alpha DS20**. The operating system for this minicomputer is **Red Hat Linux v7.1**.

FLCC's **AXP** Minicomputer is connected to the world's largest network, the Internet. Internet e-mail, FTP, Secure Shell, and Lynx are available Internet utilities on the system.

Access to the AXP is available from all FLCC networked microcomputers, and from other computers if you are connected to the Internet and have installed a secure shell utility (such as Putty, which is used on FLCC systems).

This "User's Guide" is intended to give you the necessary information to access the AXP system and complete your academic assignments requiring the use of this computer. It is not intended to be a comprehensive manual describing all aspects of the facility. If more in depth information is needed concerning a particular subject, please make use of the system's extensive HELP utility or ask your instructor for assistance.

HARDWARE	SOFTWARE
667 MHz Alpha AXP CPU	Red Hat Linux v7.1
512MB Internal Memory	Apache web server
~80GB storage for student data	ProFTPD FTP server
40/80 DLT Tape drive	Open SSH secure shell server
CD-ROM Drive	Qmail SMTP server
100mbps Ethernet	Squirrel Mail web-based email

Account Application Procedures

You will receive an account on the AXP Linux system when you logon to a Main Campus computer as Newuser on the Student Domain. This process also assigns a fresh password for existing users who have forgotten their password. Help with the Newuser logon is available in the Computer Helpdesk (B388).

Please note that you must be on Main Campus, and show photo ID to the Computer HelpDesk, in order to receive a new account or a fresh password. For security reasons, these things cannot be done from remote systems.

Academic Computing Policies

FLCC's complete statement of computing policy is available online at http://www.flcc.edu/offices/computingservices/academic_cpu_policy.html

A printed copy is available in the documentation rack outside the computer Helpdesk

Conventions Used in this Guide

IMPORTANT: Linux *does* have a case-sensitive command line interface. In other words, Linux treats commands typed in upper case differently than commands typed in mixed case or lower case text.

User-typed entries, like commands and file names, appear in boldface type.

Names of keys are represented in angle brackets... e.g. the <Enter> key. When two keys should be pressed in combination (for instance, the <Ctrl> (control) and <Z> keys), they are shown as <Ctrl+Z>. More explicitly, press and hold the <Ctrl> key down, and while doing this, press <Z> once. Cursor movement keys are referred to as <UP> (up arrow), <DOWN> (down arrow), <LEFT> (left arrow) and <RIGHT> (right arrow)

Generic names that accompany commands, for instance, **cp *filespec***, are shown in italic and bold, indicating that the user is to substitute an actual file specification for the italic/bold word. Optional parameters in command lines are enclosed in braces, for example **ls {*filespec*}**

To indicate an unspecified number of optional parameters, an ellipsis in braces {...} is used.

The dollar sign (\$) is the standard system prompt. When you see this prompt, that means that Linux is ready for a command.

Logging In and Logging Out

Login Procedure (on FLCC computers):

1. After logging on to the computer, Double-click the **INTERNET AND OTHER SYSTEMS** folder on the desktop, then double-click the **CONNECT TO New AXP** icon.
2. At the prompt "Login As " type the username given on the account sheet you received for your computer account, then press <Enter>.
3. At the Password prompt type your Linux password then press <Enter>. Your Linux username and initial password are on the account sheet you get from the Helpdesk when you request a computer account. You will not see the password on the screen as you type it. It is a good idea to change your password immediately when you first log in, using the **passwd** command. Passwords must be at least 6 characters long and should not be common English words. We suggest that you combine two words that you can remember easily (e.g. "Farmcow", "westny", etc.).

If the login succeeds, you will see the \$ prompt meaning Linux is ready for a command. If the system returns a "User authorization failure" message, then your login was not successful. You may try your password again by pressing <Enter>, and then going back to step 3.

Logout Procedure:

To log out, type **logout** at the \$ prompt and then press <Enter>. Be sure to log off the PC before you leave, also.

IMPORTANT

If you leave the computer, or turn it off without logging out, you provide a wide-open opportunity for illegal system use. Another person could then use the system with your account and copy or alter important data.

Connecting to the AXP via the Internet

All Windows systems at FLCC connect to the AXP Linux system using a shortcut icon in the "Internet and Other Systems" folder on the desktop. If you are on an Internet capable computer at home or elsewhere and you want use the AXP, you may do so using a Secure Shell program that supports Linux terminal emulation. You should connect to the host **axp.flcc.edu** using **SSH**.

Any program that supports the SSH protocol should work. A good free program that works well is called "PuTTY". PuTTY can be downloaded and used free of charge, from:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

If you prefer, you can copy putty.exe from drive I: on any FLCC computer.

We recommend PuTTY as it is free, and one of the easiest programs that we have found to use. SecureCRT from Vandyke software (<http://www.vandyke.com>) works well too. **But**, SecureCRT is *not* a free product, as PuTTY is.

To use PuTTY on non-FLCC systems, double-click on the executable Putty.exe

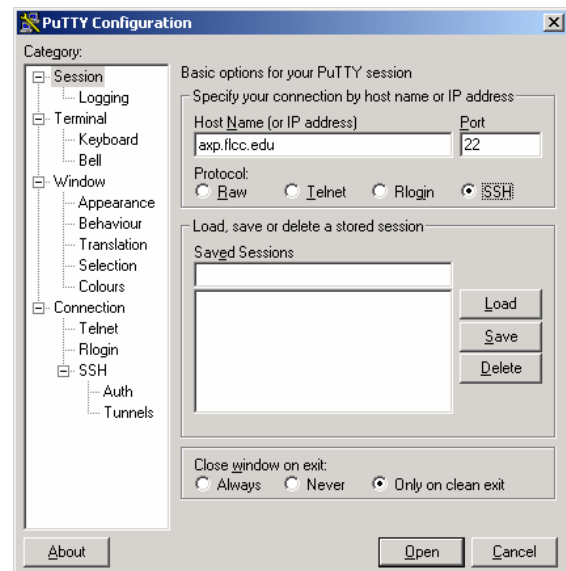
In the "**Host Name (or IP address)**" type **axp.flcc.edu**

Click **SSH** for the protocol. The port number will change to 22, the correct default for SSH.

Click **Open** to connect to axp.flcc.edu (the AXP Linux system)

You can save your PuTTY configuration by naming the settings and call them back later by running putty.exe @settingsname (see PuTTY's online Help).

For example, at the college we saved the configuration settings as NewAxp. The shortcut (in the Internet and Other Systems folder) runs c:\putty\putty.exe @NewAxp to start PuTTY with those saved settings.



Commonly Used Shell Commands

The next section of this guide contains the syntax and meaning of some of the more commonly used commands with the Linux Bash shell. You are encouraged to try them all, if you wish. If you need more help with any of the commands, you can use the Linux manual pages by typing “man” followed by the command that you need assistance with (**\$ man ls** for example).

Use the <UP> and <DOWN> arrow keys on your keyboard to display a command you have previously used so you can easily repeat it (after changing it, if desired).

c++ Compile a program written in C or C++

Syntax

\$ c++ ./sourcefilename -o executablename (./ is needed to specify current directory location)

Example

\$ c++ ./sample.cpp -o sample.exe

Compiles a source file named sample.cpp in the current directory (./) to create an executable output file named sample.exe in the current directory

To run an executable, include the "current directory" specifier. For example, to run the file created above, type ./sample.exe then press <Enter>

cal Displays a calendar

Syntax

\$ cal {MonthNumber} {YearNumber}

Examples

\$ cal *(shows a calendar for the current month)*

\$ cal 2006 *(shows a calendar for the year 2006)*

\$ cal 4 2003 *(shows a calendar for month 4 in the year 2003, ie., April 3003)*

cat Displays the contents of a file

Optionally number the lines or redirect output to another file.

Syntax

\$ cat filespec {-options}

Example

\$ cat program1.cpp *shows the entire contents of program1.cpp on the screen*

\$ cat program1.cpp -n *shows the entire contents of program1.cpp on the screen
with line numbers*

To see only the first or last lines of a file, use the head or tail commands.

To see a file one screen at a time use the more command.

cat **Combines multiple files into a single file using redirection**

Syntax using cat to combine one or more files into another file (concatenating) using the > or >> redirection options

\$ cat input-filespec {...} > outputfilespec *concatenates all specified files into another file
replaces any old output file of the same name*

\$ cat inputfile {...} >> outputfile *concatenates all specified files to the end of another file
adds onto any old output file of the same name*

Examples for concatenating files

cat prog3.cpp -n prog3lines.txt *copies prog3.cpp into a new file prog3lines.txt
with lines numbered.*

\$ cat dat1.dat dat2.dat dat3.dat > alldat.dat *copies dat1.dat dat2.dat and dat3.dat into
the new file alldat.dat*

\$ cat dat5.dat dat6.dat dat7.dat >> alldat.dat *copies dat5.dat dat6.dat and dat7.dat onto
the the end of the file alldat.dat*

cd **Changes to a new directory.**

Syntax

\$ cd directory-name *(moves into a directory, or back to home directory with no arguments)*

Example

\$ cd newdir *moves into the directory named newdir in the current director)*

\$ cd .. *moves into the parent of your current directory*

\$ cd *goes back to your home directory from wherever you are*

chmod **Changes the protection for a specified file or files.**

Syntax

\$ chmod {code} filespec

Example

\$ chmod 700 final.dat *Symbolic codes for permissions are in the online manual*

clear **Clear the screen**

Places the cursor at the top. Previous commands may still be recalled with up-arrow.

Command line recall and editing

A previously entered command can be recalled by pressing up arrow or down arrow as many times as needed to locate the desired command entry.

After locating a command, to reissue the command as is, just press **<Enter>**

To change the displayed command before you execute it:

Move around in the command line to the spot you want to change something:

<←> (left arrow) moves left 1 character.

<→> (right arrow) moves right 1 character.

<Alt-B> moves back (left) one word

<Alt-F> moves forward (right) one word

<Ctrl-A> moves to the beginning of the line.

Edit the command as needed:

**** deletes the current character

<Backspace> deletes previous character

<Ctrl-K> deletes the line from the current position to the end

<Alt-D> deletes the next word

Other keys insert the characters you type just before the current position.

cp Makes a copy of a file to a different name or location

Syntax

\$ cp input-filespec output-filespec

Examples

\$ cp fileA.txt fileB.txt	<i>makes a copy of fileA.txt the copy is named fileB.txt in the current directory</i>
\$ cp test1.dat datadir	<i>makes a copy of test1.dat in the directory datadir the copy is named test1.dat in the directory datadir</i>
\$ cp mydat.dat datadir/pg2.dat	<i>makes a copy of mydat.dat in the directory datadir the copy is named pg2.dat in the directory datadir</i>

Syntax for copying multiple files

\$ cp input-filespec {...} directoryname *copies all specified files into another directory*

Examples for copying multiple files

\$ cp *.txt textfiles	<i>copies all files with a .txt extension into the directory textfiles</i>
\$ cp pr4*.* proj4dir	<i>copies all files whose names start with pr4 into the directory proj4dir</i>
\$ cp pr2.dat pr2.cpp pr2.exe proj2dir	<i>copies pr2.dat, pr2.cpp and pr2.exe into the directory proj2dir</i>

date Displays the current date and time

Syntax and Example

\$ date *system displays the current date and time in 24-hour format*

exit **exits from the current process**

If the only running process is one instance of the shell, exit logs the user out of the system.

Syntax and example

\$ exit

finger **Displays a list of users on the AXP.**

Can also be used to look up a username

Syntax

\$ finger {search-spec} *search spec may be a personal name or a username on the system*

Example

\$ finger *lists all users who are logged on now*

\$ finger mary *Shows users whose name or username contains "mary"*

head **Displays the first few lines of a file on the screen**

Syntax

\$ head filespec {-options}

Examples

\$ head proj1.cpp *displays the first few lines of proj1.cpp*

\$ head proj1.cpp -n20 *displays the first 20 lines of proj1.cpp. The -n is required before the number if you want a specific number of lines.*

ls **Lists names and other information about the files in a directory**

The default is all files and directories in the current directory

Syntax

\$ ls {filespec} {-options}

Options can be combined in a single string after the dash. Some command options for the ls command

<i>ls Option</i>	<i>Effect</i>
-a	Displays all files, even hidden ones
-l	Displays all relevant information about specified file(s) in long format (in columns on the screen).
-s	Displays the file size in blocks for the specified file(s).
-h	Shows file sizes in Kilobytes instead of blocks
-sortorder	Sorts the output by specified field, which may be time modified (-t), size (-S), extension (-X) or other codes.

Examples for the ls command

\$ ls *lists all normal files and directories in the current directory.
Directories are shown in a different color.*

\$ ls *.txt *lists files with a .txt extension in the current directory*

\$ ls -l *columnar list of info about all normal files in the current directory*

\$ ls *prog2*.* -l *columnar list of filenames starting with "prog2" in the current directory*

\$ ls -al *all files and directories in the current directory, including hidden ones (-a)
displayed as long format in columnar list (-l)*

\$ ls -lt *list of files in the current directory sorted by time modified (-t)
displayed as long format in columnar list (-l)*

\$ ls -lX *list of files in the current directory sorted by file extension (-X)
as long format in columnar list (-l)*

\$ ls *.exe -lht *list of files in the current directory with a .exe extension
displayed as long format in columnar list t (-l)
showing size in kilobytes (-h)
sorted by time modified (-t)*

logout

Logs the user out of the system.

Syntax

\$ logout

man

Displays information from the online manual about a specified topic

.Syntax

\$ man *command_name*

Example

\$ man rm

This displays the manual page for the “rm” command and returns a : prompt. Press <Enter> one or more times at the : prompt to see more lines describing the command.

To exit the manual and return to the \$ command prompt, press **q** at the : prompt

mkdir

Creates a new directory for cataloging files

Syntax

\$ mkdir *directoryname*

Example

\$ mkdir newdir *Create a directory named newdir in your current directory*

more

Displays a file on the screen until the screen is full

Syntax

\$ more filespec

Example

\$ more prog1.cpp *shows one screenful of the file prog1.cpp*

After the first screen is displayed, press <Enter> to see the next line or **q** to quit from the more command.

mv

Move/rename - rename a file or move it to another directory

Syntax

\$ mv input-filespec output-filespec

Example (renaming a file)

\$ mv test.dat final.dat *renames the file test.dat to the new name final.dat.*

If final.dat already existed it will be overwritten without warning.

Example (moving a file into a subdirectory)

\$ mv test.dat datdir *moves the file test.dat into a subdirectory named datdir*

\$ mv test.dat datdir/final.dat *moves the file test.dat into a subdirectory named datdir with the new name final.dat*

passwd

Changes your password

Allows for a password to be changed by the current user of the account. Passwords must be between 6 and 32 characters in length. Passwords ARE case sensitive. Passwords must consist of alphanumeric characters (letters and digits) only, and cannot have any blank spaces in them.

Syntax

\$ passwd

Example

\$ passwd

Changing password for jsmith3

(current) UNIX password: *Enter your old password (invisible)*

Enter new UNIX password: *Enter your new password (invisible)*

Retype new UNIX password: *Enter your new password again (invisible)*

passwd: all authentication tokens updated successfully

pico

Opens a file for reading and writing with the Pico editor.

Syntax

\$ pico filespec *(file will be created if it does not already exist)*

Example to edit or create a file named final.dat

\$ pico final.dat

To exit pico, the command is <Ctrl + X>. It will prompt you to save your changes. Keys for commonly needed commands are shown at the bottom of the editing screen.

ps process listing

Displays information about a process and any subprocess in the current process tree.

Syntax

\$ ps {process-id}

Example

\$ ps

This shows all process listings for the current user

Example

\$ ps 1024

This shows information about process #1024

pwd Shows the Present Working Directory

.Syntax and Example

\$ pwd *system shows /home/student/jsmith3 if you are jsmith3*

rm Remove (delete) one or more files from a user directory

Syntax

\$ rm filespec

There are also several flags that can be used with the “rm” command. Such as “-r” for recursive, to remove all specified files in the current directory and subdirectories, “-i” for interactive, which prompts for the removal of each file, and “-f” which stands for “force” to remove all specified files without prompting.

Example 1

rm test.dat *delete the file test.dat*

Example 2

\$ rm -i test.dat *deletes test.dat after prompting*
rm: remove 'test.dat'? *type y or n then <Enter> in response to this prompt*

Example3

rm *.dat *delete all files with the .dat extension (best used with -i)*
rm *.dat -i *delete all .dat files with prompting for each one*

Example 4

\$ rm -f test.dat *force deletion of test.dat with no prompting*
this is the system default at FLCC

rmdir **remove (delete) a directory**

Syntax

rmdir *directory-spec* *{options}*

Examples

rmdir testdir *deletes the directory testdir which must not contain any files or directories*

rmdir junk -Rf *deletes the directory junk AND all its contents - use with caution*

sort **sort the contents of a file**

Default screen output may be redirected to a file.

Syntax

\$ sort *filespec* *{options}* *{ > OutputFileSpec }*

Examples

\$ sort testdat.dat *Shows the file testdat.dat in sorted order on the screen*

Does not alter the file testdat.dat

\$ sort testdat.dat > sdat.dat *Sorts the file testdat.dat into a new file sdat.dat*

Replaces any existing file named sdat.dat

\$ sort testdat.dat -r *Displays the file testdat.dat in reverse sorted order*

tail **Displays the last few lines of a file on the screen**

Syntax

\$ tail *filespec* *{-options}*

Examples

\$ tail proj1.cpp *displays the last few lines of proj1.cpp*

\$ tail proj1.cpp -n20 *displays the last 20 lines of proj1.cpp*

vi **Starts the vi editor**

syntax

\$ vi *filespec*

Examples

\$ vi oldfile.txt *brings up an existing file named oldfile.txt for editing*

\$ vi newfile.txt *creates a new file named newfile.txt to work on*

While in the vi editor, you may be in **INPUT** mode (which can also replace text) or **COMMAND** mode (which also provides command line entry with **:**).

To exit from vi, get into the command mode with **<ESC>**, then enter **:wq** to write the changed file and then quit. If you want to quit without saving your changes, enter **:q!**

For more details, see the VI Editor Reference in the Appendix.

who**Displays information about the current interactive users***Syntax***\$ who***Examples***\$ who** *lists currently logged in users, their terminal, and their logon time***\$ who am i** *lists your own username, terminal and logon time*

Linux File Conventions

Linux files are represented using the syntax *filename.filetype* or *dirname/filename.filetype* to refer to files in another directory. To refer to a file in the current directory, as needed by some commands, use *.* for the directory (for example, *./sample.exe* will run the file *sample.exe* in the current directory).

Examples

prog2.cpp *c++ source code for prog2*
prog2.exe *executable program image for prog2*

The *filename* is the name given to the file upon creation. Do not use a hyphen as the first or last character of a filename. Give the file a name that is meaningful to you. In the example, **prog2** is the filename and identifies the file to the user.

The *filetype* identifies the structure or type of data in the file. The filetype can have up to 39 characters, including the hyphen and underscore. With certain commands, if you omit the file type, the system applies a default value. It is recommended that you use file types in naming all files. In the examples **.cpp** identifies the file as c++ course code and **.exe** identifies the file as an executable image.

File Type Conventions

.DAT	Data file
.EXE	Executable program image file created by the linker
.CPP	Input source file for the C++ compiler
.TXT	File containing text
.HTML	File containing HTML code

Directories and file organization

Since you only have one account, and you may be taking multiple classes that use the AXP, it is a good idea to organize your files into subdirectories that represent the classes you are enrolled in. You should create a separate directory for each class.

Creating Directories

First, decide on directory names for each class (directory names have the same restrictions as filenames). Next, log on and create the directories using the following syntax:

\$ mkdir directory-specification

Example

\$ mkdir csc115 *Creates the directory csc115*

Now, when you give the **ls** command, you will see the directory **csc115** listed in a different color.

Transferring Files (Uploading and Downloading)

Using FTP

These instructions are only for FLCC computers. You may use your own FTP program in a similar way from home if you are on the Internet.

- 1) Log on to the computer.
- 2) Open the **Internet and other Systems** icon on the Windows desktop.
- 3) Double-click on the **FTP File Transfer** icon.
- 4) Make sure that **PROFILE NAME** says **NewAXP**. Click "**Ok**" and then type in your AXP username and password when prompted

Select the correct directories on your own disk and on the AXP by double clicking on the directory name. Click on **ASCII** for plain ASCII (text) files, or **BINARY** for images or other non-text files. Select the file(s) you wish to transfer (hold down the <Ctrl> key to select multiple files). After the files are selected, click the arrow pointing to the appropriate place (right arrow to transfer TO AXP, left arrow to transfer FROM AXP).

Using e-mail

You can transfer a file by e-mailing it to yourself as an attachment. Your FLCC student e-mail account provides 10 MB of storage for your e-mail and all other files you have stored on the AXP Linux system. You cannot send files whose size would cause your disk quota to be exceeded.

Printing Files

To print a Linux AXP file on your Windows PC, first transfer it to your PC using FTP. Then open it in Microsoft Word or Windows Notepad, then print it on your connected printer.

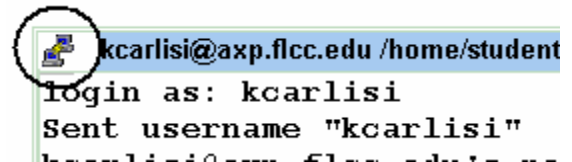
Linux Labs: Printing Instructions

In order to save (or print) what you did during a Linux session, you should start a new Putty session which uses more lines of scrollback before you begin capturing your session. Then, when you are finished with what you want to record, you can copy the entire captured session to the Windows clipboard so you can paste it into Notepad.

1. Log on to the AXP:
 - From desktop at school
 - Click on Internet & other Systems
 - Click on "New AXP: icon
 - Login to AXP

Start a New Session configured with more lines of scrollback

2. Click on the Putty icon at the left of the Title Bar



3. Select "New Session..."

Enter:

Host Name: "axp.flcc.edu"

Select option "SSH" under Protocol.

Click on "Window" in the Category Pane.

Change **Lines of Scrollback** from default of 200 to a large number.

(A choice of 1000 Lines for the Discovery Exercises(1-26) at the end of Chapter 1 in the Linux text captured the complete session which printed on 12 pages. Students are required to highlight and number the commands that correspond to the exercises in the book).

Click on Open to apply the changes.

Logon to this NEW session with your AXP Username and Password.

Do all the things you want to "record" in your captured session.

At End of Capture Session:

1. Click on the Putty icon at the left of the Title Bar.
2. Select "Copy All to Clipboard".
3. Open Notepad in Windows.
4. Type your name, the date, and the exercise information.
5. Paste the clipboard contents into Notepad.
6. Save with a descriptive name on your U: drive and Print if desired.
7. Back on the AXP, logout (or Exit) from your second session.

When you are all done using the AXP, logout (or Exit) from your original session, also.

Linux Network Services

Introduction

The AXP is connected to the Internet (the world's largest computer network). The following section briefly describes the available Internet utility programs on the AXP. More detailed descriptions are available online.

Lynx - A text-based World Wide Web browser.

SSH - Provides remote login capabilities to local and off-site computers on the Internet.

FTP - Provides ability to move files between computers.

Squirrel Web Mail

A utility that allows students to check their FLCC Student email via a web based interface over the internet. See the handouts "Student Email Basics" and "Student E-mail Beyond the Basics" for more information.

SSH

Secure Shell provides remote login capabilities to local and off-site computers on the Internet. It is similar to the telnet protocol in how it works, with the main difference being that the data is encrypted before being sent across the network.

Using SSH

Syntax

```
$ ssh remote-host -l username
```

Example

```
$ ssh axp.flcc.edu -l jsmith3
```

Exiting SSH

There are many different methods of ending a Secure Shell session. Many hosts that you log into will accept **EXIT** or **QUIT** as commands to close the remote connection. If you have trouble at the system prompt, type **QUIT** to exit.

FTP command interface

FTP provides the ability to move files between computers.

Using FTP

Syntax

```
$ FTP remote-host
```

Example

```
$ FTP watsun.cc.columbia.edu
```

Exiting FTP

There are three different ways of ending an FTP session:

At the remote host's prompt, type **LOG OUT**

At the FTP prompt, type **EXIT** or press <Ctrl+Z>

At the FTP prompt, type **DISCONNECT** and **EXIT**

FTP Commands

DIR	Lists the names of remote files and other information about them. The remote files can be either UNIX or Open VMS.
GET <i>filespec</i>	Copies remote files to the local host. Wildcards and pathnames may be used.
HELP	Displays information about how to type FTP commands.
CD	Sets your default directory on either the remote host or the local host (use the /LOCAL qualifiers). On a UNIX remote host, the lower case command 'cd' performs the same task.
SET TYPE <i>filetype</i>	Configures FTP to transmit or receive image or text files - IMAGE for images, and ASCII for text.

Troubleshooting Problems

Password

If you forget your password, repeat the "Newuser" logon process you used to get your account. Your new password will be on the account sheet you receive in B388 after you show photo ID to the attendant. You cannot get a fresh password unless you are on the Main Campus.

Mail

Most problems with MAIL have to do with incorrect addressing of messages. If you have a problem, you can ask the lab aides, other students in your class, or ask your instructor. If you think there is a system problem, contact the system administrator (Brian Miles, ext. 7301). If you are not receiving new mail, it may be because you've exceeded your disk quota.

Disk Storage Space

Users have a limited amount of disk storage space. If you discover that you have run out of disk space, then you must delete files you do not need. If, after deleting unnecessary file, you find that you still need more space for your class-related files, you can make a request for more disk space from the system administrator via MAIL to the **ROOT** account.

Cannot delete a directory

If you have to delete a directory, you must first delete all of the files and subdirectories that exist below the directory that you want to delete (that is to say, the directory to be deleted must be empty), unless you obverride with -Rf. The fast way to delete a directory and all its contents is to enter:

\$ rm -Rf dirname

WARNING: Keep in mind that this will delete the directory without prompting. Once it is gone, you can't get it back, so please be careful.

Stuck in a Loop

If your program seems to be stuck in an infinite loop, try pressing <Ctrl+C>

System HELP

Linux has an extensive utility of manual pages. You can get help while you are logged on and at the command line by using the following syntax:

\$ man *keyword* where *keyword* is the command or function you need help with

Example:

\$ man cp

If you cannot seem to get the help you need from the man utility, ask one of the lab aides at the Help Desk (B388) to help you. In general, the lab aides are not allowed to help you do things like debug programs, solve a logical problem, or help you get rid of syntax errors. They can, however, help you with non-application specific problems like getting logged on and off, helping you with printing problems, and showing you where to find the available documentation.

Where to go for more help

If you are still having trouble, contact your instructor. If you feel that the problem is a system error, then after reviewing the problem with the lab aide and your instructor, you should ask your instructor to contact the system administrator.

Quick Reference Guide

Connecting to the AXP Linux system

On any FLCC computer, double-click "Connect to New AXP in the Internet folder
From elsewhere, connect to `axp.flcc.edu` using the SSH protocol on Port 22

Logging In

Login As: *yourUsername* <Enter>

Password: *yourpassword* <Enter>

Logging Out

\$ `logout` <Enter> or \$ `exit` <Enter>

Changing your password

\$ `passwd` <Enter>

Changing password for `jsmith3`

(current) UNIX password: (characters don't show up) <Enter>

Enter new UNIX password: (characters don't show up) <Enter>

Retype new UNIX password: (you must enter exactly the same new password) <Enter>

If the command was successful, you should see the following:

passwd: all authentication tokens updated successfully

Creating a directory

\$ `mkdir dirspec`<Enter>

Changing into a directory

\$ `cd dirspec` <Enter>

To Move up one level in the directory tree:

\$ `cd ..` <Enter>

To return to your home directory

\$ `cd` <Enter> (a "cd" by itself will return the user to their login directory)

Listing your directory contents

\$ `ls` <Enter>

To display a text file on your screen

\$ `cat filespec` <Enter>

`cat filespec -n`<Enter> to show line numbers

To use the VI editor to create or edit a file

\$ `vi filespec`<Enter>

VI Editor Reference

Get into the VI editor from the \$ prompt with the command **vi filename**. You will initially be in **COMMAND mode**. The two main modes of VI are **INSERT mode** and **COMMAND mode** (see next page). Toggle between these two modes with the **<Insert>** and **<Esc>** keys.

INSERT mode is where you type text. Get into **INSERT mode** by pressing the **<Insert>** key on a PC keyboard. Ordinarily, new text you type is inserted just before your current cursor location. To replace existing text, instead, tap **<Insert>** again to toggle between REPLACE and INSERT. The title at the bottom of the screen will show that you are in **INSERT** (or **REPLACE**) text entry. While in **INSERT mode** use control keys to change your current location in the text, delete text, or switch to **COMMAND mode**. Anything else you type will be entered as text.

Insert mode control keys (using the cursor control keys on a PC keyboard)

<u>Key</u>	<u>Function</u>	<u>Comment</u>
<UP>	Move to previous line	
<DOWN>	Move to next line	
<RIGHT>	Move right 1 character	
<LEFT>	Move left 1 character	
<End>	Move to the end of the line	
<Home>	Move to the start of the line	
<Page Down>	Move to the next screen	If already at end of file, you'll only see ~ for null lines
<Page Up>	Move to the previous screen	No effect if at head of file
<Enter>	Ends the current line	Also use to insert a blank line
<Insert>	Toggle between Inserting and Replacing text. Bottom of the screen will say REPLACE or INSERT.	In REPLACE, you can replace unwanted text without deleting it after typing new text
	Delete the current character	Can be used to delete a blank line
<Backspace>	Delete the previous character	Can be used to delete a blank line
<Ctrl-U>	Delete all characters to the left of the cursor in the current line	To remove the remaining blank line, press <Backspace> or after <Ctrl-U>
<Ctrl-V> {char}	Insert character literally, or enter decimal byte value for escape codes	
<Ctrl-Y>	Insert the character from above the cursor	Useful for copying a line you typed before
<Ctrl-E>	Insert the character from below the cursor	Useful for copying another line you typed
<Esc>	Exit from INSERT mode to COMMAND mode	You must do this before you can save your file or exit from the VI editor. The "INPUT" title at the bottom of the screen will disappear.

Command Mode of VI

COMMAND mode provides keystrokes to accomplish various tasks, or you can type a colon (:) to enter commands on a command line. In ***COMMAND mode***, the cursor movement keys work just as they would in ***INSERT mode***, but some editing keys, such as <Backspace>, do not. There are some additional positioning keystrokes you can use in ***COMMAND mode*** to move around a file more quickly. If you press the colon (:) in command mode, you can then enter a command such as w(rite) or q(uit). The command line often uses a range specifier (line numbers separated by commas) to change which lines will be affected by a command or keystroke (see next page for command line entries).

COMMAND mode keystrokes (using a PC keyboard)

Keystrokes	Function	Example
gg	Go to top of file	gg goes to first line in the file
Ngg	Go to line N of the file	10gg goes to line 10 of the file
G	Go to bottom of file	G goes to last line in the file
N<UP>	Go up N lines	6<UP> goes up 6 lines
N<DOWN>	Go down N lines	15<DOWN> goes down 15 lines
N<LEFT>	Go left N characters	10<LEFT> goes left 10 characters
N<RIGHT>	Go right N characters	5<RIGHT> goes right 5 characters
W	Go to next word	W move to next word
NW	Move right by N words	4W moves right by 4 words
b	Go to previous word	b move to previous word
Nb	Move left by N words	3b moves left by 3 words
N	Move to column N (is the "vertical bar" key, Shift-\ on many PC keyboards)	46 moves to column 46 of the current line
:	Prepare to enter a command	:w write (save the file) :w try2.txt save the file with a different name (try2.txt in this case) :wq save the file and quit VI :q! Don't save the file and quit VI :q Quit VI - will give an error if you have not saved the file.
<Insert>	Exit from <i>COMMAND mode</i> to <i>INSERT mode</i> so you can type text	

Command Entry mode of VI

Commands you can enter after pressing **:** in *COMMAND* mode

<u>Command</u>	<u>Function</u>	<u>Example</u>
<code>:r filespec</code>	Read another file and copy it into the current file. Useful for inserting "Boilerplate" like a block with your name and other info.	:r info.dat <i>Puts the contents of info.dat into your file just before your current location.</i>
<code>:[X,Y] m Z</code>	Move lines X thru Y to line Z Moves only one line if ,Y is omitted.	:1,5 m 20 <i>moves lines 1 2 3 4 and 5 to just after line 20</i> :3 m 10 <i>moves line 3 to just after line 10</i>
<code>:[X,Y] w filespec</code>	Write lines X thru Y to another file. Useful for creating "Boilerplate", such as a name block.	:1,10 w top.txt <i>writes the first 10 lines out to a file (top.txt in this case)</i>