

1. Introduction

This handout is written to give users a basic understanding of the university's UNIX systems. If you find any errors in this guide or have any comments please send them to **csainz@umiami.ir.miami.edu**. This guide is also available on the web at <http://www.miami.edu/ir/pubs/manuals/>.

After reading this handout, you should be able to perform the following tasks:

- **Log on** (or **sign on**) to the system
- Execute simple commands
- Create, list, edit, and remove files
- Create and execute programs
- Obtain printouts of your files, including printouts of your program's execution
- **Log off** from the system

This typestyle will be used for things that the computer displays.

This typestyle will be used for things that you must type in.

If you have questions about anything that you have read in this document, please ask one of the User Assistants at the front desk. If you are at home, you can call the Help Line at 284-HELP.

2. Logging On To The UNIX System

Making Connections from ISF

In the ISF Computer lab, the following menu appears:

Welcome to the ISF Computer Lab

- < 1> IBM/CMS (VM/CMS)
- < 2> GABLES (OpenVMS)
- < 3> Jaguar (Unix)
- < 4> Student Integrated System
- < 5> Library
- < 6> EASY
- < 7> SCICS
- < 8> Local Prompt

Enter Choice:

If you do not see the above menu, but see the following on the screen, press **[RETURN]** and the menu should appear.

Local -011- Session 1 disconnected from xxxxx

Enter To Continue>

If you see anything else on the screen like VM/ESA or SCICS etc. Press the *[DO]* key and the *[F14]* key.

From the menu, type the number for Jaguar (UNIX) and press *[RETURN]*. You should now proceed to the section titled **Identifying Yourself**.

Making Connections via Modem

To use the UNIX system through a modem, set your communications program as follows:

Eight (8) data bits
No parity
One (1) stop bit

For up to 14400 baud the number to call to connect is:
284-6010.

For 28800 baud only the number to call to connect is:
284-6700

If you are using a low speed modem your responses will be better if you turn off both buffering and data compression on your modem.

Once you have made a connection, press *[ENTER]* and you should see the following prompt:

Local>

At the Local> prompt, type the letter **c** (for connect) and **jaguar**, the name of the UNIX system, and press the *[RETURN]* key:

Local> *c jaguar [RETURN]*

If the connection was made, the UNIX system will inform you of this with something similar to this message:

Local -010- Session 1 to JAGUAR established

You are now connected to the UNIX system.

Identifying Yourself

The system will display something similar to the following:

Digital UNIX version Vx.x

login:

At the login: prompt, type in your userid and press the *[RETURN]* key.

If you do not know what your userid is, please get the handout entitled **Userid Processor**. When you find out what your userid is, proceed with this handout.

The system will respond with another prompt:

Password:

You are required to supply a password so that the system knows that you are really who you say you are. Since this is your first time logging into the system, enter your default password. Your default password is formed by taking the last four (4) digits of your student number and the first two letters of your last name and putting them together. For example, if your name was John Public and your student number was 123-45-6789, then your default password would be 6789pu.

UNIX is case sensitive. The password 6789pu and 6789PU are not the same. Both the userid and the default password will be in lower case.

Enter the default password at the Password: prompt and press the *[RETURN]* key. Notice that what you type is not displayed on the screen. This is so no one can look over your shoulder and see your password. If your userid is valid, the system will accept the default password and log you on.

Changing Your Password

At this point, the system will ask you to change your default password. This is because your default password is too easy for someone to figure out.

If at any time you want to change your password, use the command **passwd** and follow the process illustrated here.

The system will begin the process with this prompt:

Old password:

Type in your default password (the password you just typed to get logged on) and press the *[RETURN]* key. Again, the system will not display what you are typing. The system will then respond with this prompt:

Enter new password:

You should choose a password that will be hard for others to guess. Do not use your first, middle, maiden, or last names. Do not use names in general. They are easy to remember, and also easy to guess by others. Do not pick a password that somehow describes yourself. Be obscure, but pick some series of letters and numbers that you can remember. A password is not very good if you make it so hard that even you cannot remember it.

Never write down your password. You may accidentally lose whatever you wrote your password on.

Never give out your password to anyone. It is against University policy to give out your password.

A password must begin with a letter, and it must be at least six characters long. Since UNIX is case sensitive you can mix lower and upper case letters to make it more difficult for someone to guess your password. When you have decided on a password, type it in and press the *[RETURN]* key. The system now displays this prompt:

Verify:

Since you cannot see what you are typing, the system is just making sure that you typed what you wanted to type. If you made a mistake, just press the *[RETURN]* key. The system will tell you that your password was not changed and you must begin the process again. Otherwise, just retype your new password and press the *[RETURN]* key.

If everything is correct the system will display the system prompt (also called the **shell**), **jaguar.ir.miami.edu>**.

If you forget your password, you will need to contact Terry Helmers at 284-3849 to have it reset.

3. File-Related Commands

In order for you to use any computer effectively, you must know how to list, create, and edit files. This section will focus on the commands that are used to accomplish these tasks.

Filenames

Before we get into the commands, you will need to know how the UNIX system names files. Filenames on UNIX may start with any alphanumeric character and can have a length of up to 100 characters. In many cases they follow the format: *filename.ext*. The *ext* (for extension) part is used to better identify the file (e.g. *.txt* for a text file and *.c* for a C source code). Examples of file names and some common extensions are listed below:

stuff.txt	(TeX T file)
view.Z	(compressed file)
program.c	(C source file)
program.p	(Pascal source file)
data.dat	(DATa file)

Unlike OpenVMS, UNIX does not automatically keep older versions of the files. Be careful when copying or deleting files since the only way to recover lost files is by requesting them from tape backup when available.

Listing Files

The **ls** command will list all the files that you have.

```
$ ls
```

or for a more detailed output

```
$ ls -l
```

If you do not have any files in your directory, the system will respond with the message:

```
total 0
```

Otherwise, the system might display something like this:

```
total 5
drwx----- 2 jpublic 512 Sep 27 1993 Mail
-rwxr-xr-x 1 jpublic 64964 Apr 21 1994 a.out
drwxr-x--- 2 jpublic 512 Apr 27 1992 bin
-rw-r--r-- 1 jpublic 12798 Apr 13 1994 giftrans.c
-rw-r--r-- 1 jpublic 146244 Apr 14 1994 gzip
```

Creating And Editing Files

Creating and editing a file is accomplished through the vi editor. To create a new file, type in this command:

```
vi filename
```

where *filename* is the name of the file you want to create. For our purposes, we will use "newfile.txt" as the filename for the new file.

Once you press [RETURN], the system will enter the editor. The editor will then display its prompt and a message telling you that newfile.txt does not already exist:

```
~
~
~
~
~
"newfile.txt" [New file]
```

The vi editor operates in two modes; the Input mode and the Command mode. In the Input mode the vi editor allows the user to type text. The Command mode allows the user to carry out commands on the file being edited.

The two most used commands to switch to the Input mode is Append, which is executed by pressing the letter “a” and Insert, which is executed by pressing the letter “i.” A description of these will follow.

To switch from the Input to the Command mode either press the escape (Esc) key or the key combination of Ctrl-[(hold down the control key, and press the left bracket key). Either one of these will change the vi editor mode to the Command mode.

Command mode commands

If a command is typed while the editor is in the Input mode the command will just be displayed as part of the document. In that case just delete the characters and switch to Command mode.

a	Switches to Input mode and allows user to start typing the document one space ahead of where the cursor was.
dd	Deletes the line the cursor is on. To delete multiple lines precede the command with a number. (e.g. 10dd)
i	Switches to Input mode and allows the user to start typing the document from where the cursor was.
u	Undoes the last command.
x	Deletes the character the cursor is on

The following commands must be preceded by the colon (:). Unlike the above commands, these will show at the bottom of the screen.

:w	Writes or saves the file
:wq	Saves the file and quits the vi editor
:q!	Quits the vi editor without saving the changes
:r <i>filename</i>	Inserts the file named “filename” into the current document. The file is inserted into the line after the cursor when the command was invoked.

4. Executing Some Simple Commands

The system will now allow you to type in commands for it to execute. Here are a few of the most common commands.

Unless specified otherwise, you must press the [RETURN] key at the end of a command. For simplicity we will omit this from this point on.

man command

This command gives detailed information on the use of any command. For example, if you wish to know what the ls command does, type:

```
$ man ls
```

cat filename

This command displays the text file on the screen. Never use this command on a non-text file, since this will probably lock up your terminal.

```
$ cat .login
```

cc filename.c

This command is used to compile C source programs into executable files. When typed just as above the executable file will have a default name of "a.out."

```
$ cc MyProgram.c -o MyExecutable
```

The -o option tells the compiler to save the executable program with a name of MyExecutable instead of the default a.out.

cd directory_name

cd /path/directory_name

This command changes the directory to the specified directory name. The directory name can be a relative name or the full path. If two periods (..) are used as the directory name (i.e. cd ..) the directory will move up one, or to the parent directory.

```
$ cd mail  
$ cd ..
```

Both commands above are relative to the user's current directory. The first command (`cd mail`) will change the directory down to a directory named "mail." The second command (`cd ..`) will change the directory up one, to where the user was originally.

control-c

The control-c command (hold down the control key and press the letter c) will interrupt any command currently executing. If, for example, the `ls` command is used and the listing of files takes too long, you can stop it by pressing the control-c key combination.

cp src_filename dest_filename

This command copies the source file `src_filename` to the destination file `dest_filename`. The source file is left unchanged. For example, if you are working on a program called `MyProgram.c` and you want to make a backup copy, you would type:

```
$ cp MyProgram.c MyProgram.c.bak
```

ls

ls -l

This command displays a list of the files in the current directory. The "-l" option provides for a more detailed listing.

lpr -Pisf filename

This command sends the file named to the printer. The options `-Pisf` tells which printer the file should be printed to. In the above case the "isf" string tells the system that the file should be sent to the ISF printer in Ungar Building, room 103A. As with the `cat` command, only text files should be printed.

```
$ lpr -Pisf MyProgram.c
```

mkdir directory_name

This command makes or creates a directory. In the example below a directory called `homework1` is created from the current directory.

```
$ mkdir homework1
```

mv old_filename new_filename

mv old_directory_name new_directory_name

The mv command moves or renames files. As shown above both files and directories can be renamed on the system. If a different path is included when renaming files, the files are moved to the new directory, as in the example below.

```
$ mv plain.txt ./mail/
```

The above command moves the file called “plain.txt” from the current directory to a directory called mail, just below the current directory. Since only the destination directory was specified the name of the file will remain the same.

pc MyProgram.p

This command compiles the PASCAL source program called “MyProgram.p,” into an executable file. When typed just as above the executable file will have a default name of “a.out.”

```
$ pc MyProgram.p -o MyExecutable
```

The -o option tells the compiler to save the executable program with a name of MyExecutable instead of the default a.out.

rm filename

This command removes or deletes the named file. Once a file is deleted it cannot be recovered, so extreme caution should be taken when using this command.

```
$ rm oldfile.txt
```

rmdir directory_name

This command removes or deletes the named directory. The directory must be empty before it can be deleted. Once a directory is deleted it cannot be recovered, so extreme caution should be taken when using this command.

```
$ rmdir homework1
```

5. Compiling And Executing Programs

More than likely, you will be using the system to create and execute programs. The creation of program files, called **source files**, use the exact same process as shown in Creating And Editing Files, in Section 3. The only difference is that you must use the correct extension. The correct extension depends on what kind of program you are writing. If you are writing a Pascal program, the extension must be **.p**. If it is a C program the extension must be **.c**.

When you have completed entering in your source file, you will need to use a compiler. A compiler is a program that checks your source file for syntax and creates what is called an object file.

Compiling a Pascal Program

To compile a Pascal source file called MyProgram.p type the following:

```
pc MyProgram.p
```

The name of the Pascal compiler is *pc*. When no other options are used and if the program is free of errors a file called *a.out* will be created. This file is the executable program.

By using the **-o** option, the compiler can be forced to save the executable program with another name. The following line tells the compiler to compile the Pascal source file named *homework.p* and create an executable called *homework*.

```
pc homework.p -o homework
```

Make sure that when using the **-o** option you do not use the same name as your source file. If you do, the compiler will replace your source file with the newly created executable.

Compiling a C Program

To compile a C source file called MyProgram.c type the following:

```
cc MyProgram.c
```

The name of the C compiler is *cc*. When no other options are used and if the program is free of errors a file called *a.out* will be created. This file is the executable program.

By using the **-o** option, the compiler can be forced to save the executable program with another name. The following line tells the compiler to compile the C source file named *MyProgram.c* and create an executable called *program*.

```
cc MyProgram.c -o program
```

Make sure that when using the **-o** option you do not use the same name as your source file. If you do, the compiler will replace your source file with the newly created executable.

Executing a Program

To execute the program, just type the name of the executable at the system's prompt. Remember if no options were used and no errors were found the default executable name is a.out.

6. Printing Files

General Printing

To make a printout of your files type:

```
lpr -Pisf filename
```

The options -Pisf tells which printer the file should be printed to. In the above case the "isf" strings tells the system that the file should be sent to the ISF printer in Ungar Building, room 103A.

Only try to print text files, all other files will produce unexpected results.

The User Assistants at the front desk of the ISF will pick up the system printouts every fifteen minutes. You will need to go to them to collect your printout. If you decide to pick up your printout later, it will be filed at the front desk alphabetically by your username for 48 hours.

Printing Your Program's Execution

To print the output of an executable program file redirection is used. With redirection you can redirect a files' input or a files' output.

The following line executes the program MyProgram and instead of reading from the keyboard, it uses the content of the file input.txt as its input.

```
MyProgram < input.txt
```

The next line redirects the output from the command "ls -l" to a file called list.txt.

```
ls -l > list.txt
```

This file can now be printed with the command:

```
lpr -Pisf list.txt.
```

7. Accessing Electronic Mail

Electronic Mail allows users to send messages to other users within the university community and around the world. To access E-mail, type **pine** at the system prompt.

8. Logging Off

When you have completed all your work, there is one final command you will need to enter:

exit

The system will log you off and the Local> prompt will return. At the Local> prompt type:

lo

to completely log off the university's computer system.

For further assistance see a User Assistant at the front desk or call 284-HELP.