**CSCI.258 (63.258) - Introduction to Operating Systems Using UNIX**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Professor Krishnamoorthy, Computer Science Dept., Framingham State College**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Homework #1 – 150 Points**

**Coverage: Chapters 1, 2, 3, 4 and 5 from Das Textbook.**

**Refer Syllabus for due date**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objectives**

1. Login to UNIX system and logout
2. Learn features of the vi editor and to use online help (man pages). Gain experience using them to create a document
3. Know a list of UNIX commands with one line description for each
4. Learn redirecting standard output (stdout) to a file using the output redirection operator > instead of output going to the monitor, the default device for stdout.
5. Learn more on UNIX file system such as file attributes (permissions), ownership, links, file size, modification date, and inode number of files, and system directories and files
6. Gain experience using a set of UNIX commands covered in chapters 1, 2, 3, and 4
7. Learn to transfer files between UNIX system and Windows system and print them
8. Understand the output of various commands and answer questions

This homework has two parts. You must complete both parts and submit them together on or before the due date using the UNIX server at the university.

**The text files you submit for Part I and Part II must be present in your account on the UNIX system at FSU. Instructor will check your account for the files.**

**IMPORTANT NOTE**: UNIX commands are case sensitive. For example, the command vi is not same as VI. File names and directory names are case sensitive. The file Hw1 is not same as hw1. Type carefully.

**Part I: Read Chapter 1, 2, 3, and 4**

**How to get the answers to the questions in Part I of the homework?**

**The answers to this homework should not be handwritten or typed**. The answers must be obtained by issuing the necessary UNIX commands on the FSU UNIX server. To capture the output of a UNIX command use the **script** command as described below:

1. Using *script* command lets you record your login session in a file. To start *script* type the command *script* and to end the capture, type *exit* command. The *script* command creates a file called, “typescript” in your current working directory in your account, if you do not specify any filename with script command as argument. If you want the output to go to a specific filename of your choice, then specify the filename as the argument to script command, like *script hw1q1output*. Refer your textbook for more information on the use of script command. When you use *script* command, make sure you (1) type “Enter” key few times to get space between questions, (2) then, issue the *echo* command to display the question number, like *echo hw1q1*, (3) type “Enter” key few times to get space after the question number, and then (4) type the required UNIX command(s) for that question. An example is given below for question #1. In this example, $ is the prompt displayed by the Bourne Shell, which processes the UNIX commands you type.

$

$ echo Q1

$

$ uname

**Caution:**

Every time you type *script* command, it will erase the current content of the “typescript” file or the filename you specify in the script command. To avoid losing the answers you have obtained so far for one or more questions, you must copy the *typescript* file into another file using the *cp* command as shown below:

cp typescript hw1q1to5

where hw1q1to5 is a filename.

**Issue the cp command immediately after exit from the script to avoid losing it by issuing the script command again.**

**Copying Several Files into One File**

Assume that you have obtained 4 files named hw1q1, hw1q2, hw1q3, and hw1q4 containing the answers to homework1 questions Q1, Q2, Q3 and Q4. You can copy the content of all four files into one file, say hw1 using *cat* command as shown below.

cat hw1q1 hw1q2 hw1q3 hw1q4 > hw1

Where > is the output redirection operator that stores the output into a file instead of displaying in the monitor.

**Transferring Files between UNIX and PC**

ftp can be used to transfer file from UNIX server to PC and from PC to UNIX server.

After you have obtained all the answers in one file for Part I and creating Part II file, transfer both files to your PC using ftp.

**For the files to be stored on your PC, give the file name that ends with .txt**. For example, let us say the Part I file on UNIX is hw1part1. The file name on the PC side should be hw1part1.txt. Similarly, if hw1part2 is the file on UNIX for Part II, store that file on the PC as hw1part2.txt. The .txt extension will help in opening the file in Windows Notepad by double clicking on it; Windows operating system will automatically use Notepad text editor to open the file ending with .txt.

Refer to the ftp document provided in the assignment section for details on how to use ftp and do file transfer.

You should **“not edit the output file”** obtained from the UNIX serverusing Word or any editor on the PC.

**Backspace Key in UNIX:** While working with Windows, the *backspace* key on your keyboard erases the previous character you have typed. In the UNIX server, **to backspace**, you must press the ‘Ctrl’ key and the ‘h’ key at the same time. It is said as, “control h” and written as Ctrl-h.

**Erasing Entire Line:** When you type “Control u” (Ctrl u), the entire UNIX command line you have typed will be discarded (ignored). You will see the text disappearing.

**Format of UNIX Command**: UNIX command has 3 parts as shown below:

**Command-name** [**options**] [**arguments**]

First part (Command-name) is the name of the UNIX command. Part 1 is required. Part 2 is zero or more options. Some commands, like pwd, do not have an option. Most commands have multiple options. Each option is preceded by the minus symbol (-) followed by one letter, like -a. Multiple options can be combined together with one minus sign followed by the letters of the options, like -ai. Whether to use the option or not depends on the task you are trying accomplish. It is not required to use options. To indicate that you do not have to use the option(s), it is shown in square bracket [] as [options]. Part 3 is arguments. A command may have zero or more arguments, similar to options. Specifying argument(s) depends on how you use the command. To indicate that it is not required to use arguments, it is shown between square brackets [] as [arguments]. Command name, each option and each argument is separated by one or more spaces. The following examples show the UNIX command, ***ls*** (NOTE: letter l & s) with various parts:

ls

ls –l (NOTE” the option is letter 1 and not digit 1; they look similar.)

ls –l –i

ls –li

ls –l /home

**Homework Submission Method**

**Email the text file for Part I and Part II in one email on or before the due date.**

Do not submit using Blackboard digital drop box.

**IMPORTANT Note: Exit from script before using vi editor and enable it after exiting from the vi editor. To repeat, you should not have script command enabled while using the vi editor. If you submit your homework with script enabled while using vi editor, you will have lots of junk characters in the output file. Your homework will not be corrected as well as you will lose points for using vi with script enabled.**

**Caution:**

1. **You will NOT get any points if you do not use the UNIX system to get answers to the questions as explained above.**
2. Go through the output of each command and understand the output. This will help you learn and help you in your quizzes.

Note:

Instructions for (1) login to UNIX system, and (2) transferring files between Windows PC and UNIX system using ftp are given in the assignment section. You can also contact Help Desk for assistance, in case of any problem.

To do your homework you can use FSU lab PCs in the Hemenway Hall building, ground floor room HH G17 in addition to your PC. Help Desk is located in HH G17. It is open seven days a week. Contact Help Desk for hours.

**On-line Help in UNIX – *man* command**

UNIX operating system provides on-line help (manual) for all the UNIX commands the system supports. If you want to find all the details about a command, use the “*man*” command. It is a short form for manual. For example, to find the details about “*more*” command, type “*man more*” after the shell prompt $ as shown below:

$ man more

Complete Part I work described below using the script method to capture output as explained earlier.

Find the appropriate command(s) for each question, type the command(s), and capture the output of the command(s) for each question below into your homework output file.

**Remember that the purpose of the homework is not just type the commands; it is to learn what the commands do. Knowing is learning. It will also help to get more points in your quizzes.**

**Part I: Reading, learning and use of certain UNIX commands**

**You have to use the UNIX server at FSU to get output for the commands.**

**IMPORTANT NOTE: You must study about the commands needed for each question and know how to use them before you begin answer the question.**

**CAUTION: Part I of the homework will take time (several days) to complete since you have to read the chapters in the textbook, find the correct command and its option(s) and argument(s) for each sub-question in a question and then type it to get the answer. Hence, start working on it from day one and devote sufficient days and time to complete the work on time.**

**A directory is not a text file. Therefore, do not display the content of a directory using cat command. If you do, you will see junk output.**

**IMPORTANT NOTE: All your files for all your homework must be in your directory (and or subdirectories) until the end of the semester for the instructor to view them at any time. Absence of homework file(s) will result in credit not given to the homework you submit.**

**Q1. Hint: List of commands to read and use: uname, hostname, stty, date, echo, who, whoami, “who am i”, and mv**

Use appropriate command(s) and options to show the answers to the following questions.

1. Display the name of the UNIX operating system
2. Display the version (also called revision) of the operating system
3. Display the name of the computer (host) using two different commands
4. Using one command, display the key settings for (a) backspace previous character and (b) to discard entire current line. From the output, using echo command answer the question: what is the key setting to delete/discard entire current command you have typed so far in the line. Your answer should have descriptive text like: echo “Key setting to discard current command line is …”
5. Display the current date and time
6. Display the list of users currently logged into the system
7. Look at the output of who command; then, using echo command answer your terminal ID and your login time
8. Change the working directory to root directory
9. Display the pathname of the working directory
10. Change your working directory to your login directory without using any option or argument in the command, that is, type just the name of the command (one word).
11. Copy one of your files into a file called new. This is one way of creating a new file.
12. Using long option, display the attributes of the new file
13. Using echo commands, answer the following for the new file. Your answer should have descriptive text like: “The character representing file type is: -“ “The owner permission is ”
    1. The character representing the file type
    2. The permissions for the user (owner). NOTE: It is 3 characters
    3. The permissions for the group. NOTE: It is 3 characters
    4. The permissions for others. NOTE: It is 3 characters
    5. The size of the file in bytes
14. Rename (move) the new file to another name

**Q2. Hint: List of commands to read and use: cd, pwd, mkdir, rm, cat, echo, ls, ln, and echo, and .. in the pathname**

**Read the description at the end of Part I that shows the relationship between directory content, inode table and file content before doing this question.**

1. Change your working directory to your login directory
2. Create the file called *share* using cat command under your login directory. Enter one or two lines of text in it.
3. Display the content of the *share* file
4. Display the number of hard links to the *share* file ONLY using the long option
5. Using echo command, answer how many hard links (hard link count) the *share* file has.
6. Display the inode number for the *share* file using appropriate option.
7. Create the subdirectory, *linkdir* under your login directory
8. Change working directory to *linkdir.* Stay in this directory until the rest of this question.
9. Create a hard link called *share2* to the *share* file. Remember that the share file is the parent directory and NOT in the current working directory. Hence, you need to specify correct relative or absolute pathname for share file in your hard link command.
10. Display the inode number of *share2* file
11. Using echo command, answer whether the inode number for *share* and *share2* files is same or different. Look to the earlier command to see the inode number of *share* file.
12. Using echo command, answer what is the inode number of *share2* file
13. Display the number of hard links to the *share2* file
14. Using echo command, answer the number of hard link count to *share2* file.
15. Create a soft (symbolic) link called *softShare* to the *share* file. Caution: If you specify wrong pathname in the softlink, then you will not be able to display the content of the file; instead the system will display an error message.
16. Display the inode number of *softShare* file
17. Using echo command answer whether the inode number of *softShare* is same as *share* or not. From the inode number answer whether share and software are same file or two different files.
18. Display the attributes of the *softShare* file using long option in the command. Notice that the number of hard links and the file permission on *softShare* is different from the *share* file.
19. Display the content of share2 file
20. Display the content of *softShare* file
21. Using echo command, answer whether *share2* and *softSahre* have the same content as that of share file or not

**Q3. Hint: List of commands to read and use: use of wild card characters (\*, ? and set []), mkdir, rmdir, touch, cd, pwd, ls, .. and .**

* 1. Change working directory to your login directory
  2. Create a subdirectory called ‘wild’ for this question under your login directory
  3. Change working directory to the wild directory created earlier
  4. Display the pathname of the working directory
  5. Use touch command(s) to create the files: intro, notea, noteb, notex, ref1, ref2, ref3, section1, section2, section3, section4x, section4y, section4z

**Using one or more of the wild card charaters \*, ? and range (also called set) like [abx] and [1-3] in the file pathname,** issue appropriate command(s) to do the following. Note: You must show the use of the above three wild card characters individually in one or more commands for part (6) to 11) below.

1. list all files that begin with *section* using wild card character \*
2. List the files section1, section2, and section3 using wild card character ?
3. List the intro files only using a wild card character
4. List the files section1, section3, ref1 and ref3 using set/range
5. Change the permission to all files that begin with *section* to rwxrw-rw-.
6. Display the file attributes to show that file permissions changed
7. Change working directory to the parent directory using relative path
8. Display pathname of working directory
9. Issue command to delete the wild directory
10. Using echo command answer whether you are able to delete a non-empty directory
11. Change working directory to the wild directory
12. Delete all the files using wild card characters **without using –r option**
13. Change working directory to parent directory
14. Delete the wild directory

**Q4. Hint: List of commands to read and use: cd, pwd, ls, echo, cat, od, quoting character and output redirection operator >**

1. Change your working directory to your login directory.
2. A new file can be created in 4 different ways: (i) vi, (ii) touch, (iii) cp, and (iv) cat command. You have used touch and cp commands before. In this question, you will use cat command. In part II, you will use the vi text editor.
3. Create a short file using cat command and output redirection operator. The content of the file must be typed from the keyboard, which is the default standard input (stdin) device. End the input to the file using Ctrl d keys at the same time.
4. Copy the command you have created to a new file.
5. Display the content of the file you have created using one *od* command with appropriate option(s) for each question below:
   1. Display in character form
   2. Display in hexadecimal form
   3. Display in octal form
   4. Display in both character and octal form
6. Using echo command, answer how the end of line (also called new line) character is displayed. **NOTE: Shell gives special meaning to certain characters like \*, ?, \ and others. To make shell not to give special meaning to that character, you have to put a \ in front of it. This is called quoting character. Therefore, you have to quote the character \ by putting a \ in front of it like \\ in order for the echo command to display it. Read more about it quoting character in your textbook.** The end of line character is the representation of the “enter” key. The ASCII character code for the “enter” key is not a displayable single character like the letter ‘a’ for example. Hence, the system displays it differently.
7. Using echo command, answer: (i) the ASCII number code for new line (enter key) and (ii) for any one letter, display the letter and its ASCII number code. This can be done by comparing the output of the character form and the hexadecimal form above. Also, search Internet for ASCII table and give the ASCII code for capital Z and lowercase z. You have to do this search on your laptop by running browser and going to site like google.com

**Directory, File Content, File Attributes, and Hard and Soft Links**

Directory joe inode# inode table

1999

d, rwx------, link count, Content

Address

Type, perm, link count Content

Address

L, rw-r—r--, 2, 150, 10, Content

Address

-, rw-r—r--, 2, 150, 10, Content

Address

. 1999

Hw1q1output 2000

softShare 3000

share 4200

share2 4200

2000

3000

4200

This file is a short file created using cat command for the hard link and soft link questions in hw1.

../share

Figure 1: Relationship between directory, inode table and file content on disk

The directory, inode table, and file content are all on disk storage. The relationship between them is shown in the Figure 1 above.

A file system is a set of files residing one part of disk called partition, or on an entire disk, or on multiple disks depending on the size of the file system. Each file system has an inode table. The inode table is an array of certain size; this is similar to the array in a programming language like Java. Each entry in the inode table has the ***attributes*** of a file such as file type, file permission, owner, group owner, modification date, number of hard links, file size and disk address where the content of that file is stored and more. Each array index to the inode table is called an inode number. Hence, each file has an ***inode number*** which is the index like array index to the **inode table** where its ***file attributes*** are stored.

For example, the inode number of ***share*** file is 4200 and it has file type as – representing ordinary file, permission of rw-r—r--, number of hard link as 2, user identifier (uid) of 150, group identifier (gid) of 10 and so on and at the end the disk address where the content of the file is stored.

***Share2*** is another name for the share file content. Notice that the node number for share and share2 file is the same, which is 4200. In other words, it is like one person having two names like an official name and an alias (also known as name).

Directory (example ***joe***) is a special file with type directory which is displayed in the ls –l command as character ‘d’. It contains (content) is names of files and names of subdirectories in that directory and their inode numbers. When you create a directory using mkdir command, the system automatically creates two entries in that directory: (1) the inode number for the directory itself represented as . (dot) in the name field, and (2) inode number of the directory’s parent directory represented as .. (double dot) in the name field. In figure 1 only the . entry is shown.

**Part II: Using *vi* editor and *man* command – Read vi chapter 5**

**IMPORTANT NOTE: To do Part II, open two login sessions to the UNIX server, that is, login to the UNIX server twice. In one login window, issue man command to find the one line description for each command and in the second window use vi editor and create the Part II file.**

**CAUTION: Part II of the homework will take time (several days) to complete since you have to find the one line description for more than 100 commands and type each description in the vi editor.**

**CAUTION: You must use the vi editor on the FSU UNIX server. Using any other editor is not acceptable. Using any system other than the UNIX/Linux system is not acceptable. You cannot create this file by any other way other than using the vi editor on the UNIX server since the purpose of this assignment is to learn to use the vi editor.**

**IMPORTANT NOTE: Currently, the vi editor on the HP-UX server is generating “no disk space” error and hence it is not usable to create a file. The vim editor is similar to the vi editor. You can download the Windows version of the vim editor to your laptop (PC). Then, use the vim editor to create the file. After creating the file, upload it to the UNIX server using ftp. Search the Internet for the link to the vim editor. The following link may work:** [**http://www.vim.org/download.php**](http://www.vim.org/download.php)

For Part II, a document is given in the assignment section. Create a UNIX file with the content given in Homework 1 - Part II document **using the vi text editor**. As you create the file, fill the “**description part**” for each command with one line description. Few examples for the one line description are given in the Homework 1 – Part II document for illustration.

Online manual pages for each command are available in the UNIX system. This can be accessed by using the *man* command. For example, to access the manual page for *cp* command, type the *man* command as shown below:

man cp

This will display the first page of the manual. On top of this page, you will see the one line description of the command. Type this one line description in the Part II file you are creating using the vi editor.

**NOTE:** There may not be man pages for couple of commands. For these commands, search the Internet for the description.

After creating the file on UNIX system, transfer the file to your PC using *ftp* command. Instruction for using ftp is given in the assignment section.

**CAUTION:**

1. All your homework files must be present on the UNIX server until the final grade is submitted at the end of the semester.
2. Do not edit the output files on the UNIX server or after transferring on your PC.