Homework Assignment 1

Name: **Asrar Syed**

Due by: **September 17, 2024**

Course Section: **CSC 3320-034**

1. (5 points) Distinguish between **system programs** and **system calls**.

System programs are higher-level utility programs provided by an operating system that can preform various system tasks, such as file management tools or system status and information programs. They do not run in a kernel mode but request services from the kernel. System programs are a user-level program that use system calls to preform tasks.

System calls provide the low-level interface to the operating system's kernel, which controls access to the hardware and system resources. System programs rely on system calls to function.

2. (5 points) What is the pipe mechanism in **UNIX**? Show at least one command using pipes and explain how the pipe mechanism works.

It is a form of inter-process communication that enables data to be passed from one process to another without the need for a intermediate files. Pipes is denoted by a single vertical bar ( **|** ) symbol. This mechanism allows the output of one command to to be used as the input for another command.

Example: echo "Hello World" | tr 'a-z' 'A-Z'

3. (10 points) Write your own definition of system-level programming and store your definition in a text file, named **file1.txt**. Write the reasons why C is more preferable for system-level programming and store the reasons in a text file, named **file2.txt**. Combine the two files using cat command and name the result file as **combined.txt**. Finally, use **cat** command again to append “These contents were combined from two files.” Record everything you did so far on the terminal to answer this question into a file. Paste the file’s contents as your answer.

File1: System-level programming involves writing software that interacts closely with hardware and manages system resources like memory and I/O devices. It’s used to build things like operating systems, device drivers, and embedded systems.

File2: C is popular for system-level programming because it's fast and gives direct control over hardware. It lets you manage memory and resources efficiently, which is important for things like operating systems. C is also portable, so programs written in it can work on different platforms. It doesn’t add extra layers, keeping the code lightweight and quick.

Commands:

cat file1.txt file2.txt > combined.txt

echo "These contents were combined from two files." >> combined.txt

A screenshot of a computer

Description automatically generated

4. (5 points) Assume the following two paths are absolute paths of **test1** and **test2** directories in a linux system:

**/usr/bin/test1, /tmp/test2**

Write down the relative path of the **test2** directory relative to **test1** directory.

Relative Path is ../../../tmp/test2

5. (5 points) What do the following shortcuts do?

**CTRL-D, CTRL-C, CTRL-H, CTRL-U, CTRL-Z.**

CTRL-D: Sends a end -of-file (EOF) signal.

CTRL-C: This interrupts a running program.

CTRL-H: This erases the character before cursor.

CTRL-U: This erases the entire line.

CTRL-Z: This suspends a running program.

1. (5 points) How would you resume a suspended command?

The **fg** command will resume the most recently stopped (suspended) job in the foreground.

6. (5 points) How **mv** command can be used in two ways? Please explain. What does the **-i** flag do in the **mv** command?

The **mv** command can be used to rename a directory or to move a file to another directory. The **-i** flag stands for inquire.

Move a file to another directory: mv -i <file-spec> <dir>

Rename a directory: mv -i <dir> <dir

7. (10 points) Suppose the permission of a file, named **csc3320.txt** has been set as **rwxrw-r-x**. Please explain these permissions. Write the command using absolute octal numbers to change the file permission so that the user has only **read**, the groups have only **write**, and others have only **execution** permissions.

r – Read permissions

w – Write permissions

x – Execute permissions

There are nine letters denoting the read, write, and execute permissions for a user, group and others. The first three letters are permissions for the user owner. The second three letters are permissions for the group owner. The last three letters are permissions for others.

Set the file permission of one file as r---w---x

$ chmod 421 <file-spec>

8. (10 points (bonus)) Answer question 7 differently: issue three commands to change file permissions of user, groups, and others separately. Finally show using **ls** command with appropriate flag that file permissions changed.

My **la** is shorthand for **ls -l**

A screenshot of a computer

Description automatically generated