

VIETNAM – KOREA UNIVERSITY OF

INFORMATION AND COMMUNICATION TECHNOLOGY

**DEPARTMENT OF COMPUTER SCIENCE**

**REPORT**

**PROGRAMMING SYSTEM**

**TOPIC: WRITE OR MODIFY FUNCTIONS RELATED TO FILE MANAGEMENT - PROGRAMMING THE EXECUTION PROCESS**

|  |  |
| --- | --- |
| Group member (Group 7) | : Phạm Thanh Trường  Phan Văn Bằng  Phan Văn Lai  Hồ Thắng Trình |
| Instructor | : TS. Đặng Quang Hiển |

**Da Nang, December 23, 2022**

VIETNAM – KOREA UNIVERSITY OF

INFORMATION AND COMMUNICATION TECHNOLOGY

**DEPARTMENT OF COMPUTER SCIENCE**

**REPORT**

**PROGRAMMING SYSTEM**

**TOPIC: WRITE OR MODIFY FUNCTIONS RELATED TO FILE MANAGEMENT. PROGRAMMING THE EXECUTION PROCESS**

|  |  |
| --- | --- |
| Group member (Group 7) | : Phạm Thanh Trường  Phan Văn Bằng  Phan Văn Lai  Hồ Thắng Trình |
| Instructor | : TS. Đặng Quang Hiển |

**Da Nang, December 23, 2022**

# **OPENING**

System programming is an essential and important foundation in any computer’s application development, and always evolving to accommodate changes in the computer hardware. System programs provide an environment where programs can be developed and executed. In the simplest sense, system programs also provide a bridge between the user interface and system calls.

The main aim of system programming is to design of system software and to provide basic for judgment in the design of software. Understanding how to properly program systems is a critical skill for developers. Therefore, our group has decided to choose the topic: "Write or modify functions related to file management - programming the execution process". To understand programming principles and techniques for system programming, basic components, and functions of operating systems.

**THANKS**

We would like to thank the Department of Computer Science, Vietnam - Korea University of Information and Communication Technology for creating conditions for us to prepare this Report. We would like to express our sincere thanks to our teacher Mr. Dang Quang Hien has enthusiastically guided and advised us during the implementation of the project. In the process of implementing the thesis, we have learned a lot of useful knowledge and valuable experience as the foundation for the process of working and researching later. I would also like to thank the teachers in the Department who have dedicatedly taught and equipped me with important knowledge in the past time. Sincere thanks to all of you, especially the members of the group, who united, supported, and helped us during our time at the school and during the implementation of this thesis. Although we have tried to complete the application within the scope and ability, it is inevitable that there will be shortcomings. We look forward to receiving your understanding and suggestions. We will try harder to improve and develop our topic. Our team sincerely thanks you!

**COMMENT**

(Of Instructor)

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

Da Nang ..……………………………...

Instructor

TS. Đặng Quang Hiển

**TABLE OF CONTENTS**

[OPENING 1](#_Toc122649738)

[LIST OF PICTURES 10](#_Toc122649739)

[PART I: INTRODUCE 11](#_Toc122649740)

[1.1. Topic name 11](#_Toc122649741)

[1.2. Group Member 11](#_Toc122649742)

[PART II: THEORY 12](#_Toc122649743)

[2.1. Linux File Ownership 12](#_Toc122649744)

[***2.1.1. User* 12**](#_Toc122649745)

[***2.1.2. Group* 12**](#_Toc122649746)

[***2.1.3. Other* 12**](#_Toc122649747)

[2.2. Linux File Permissions 12](#_Toc122649748)

[***2.2.1. Changing file/directory permissions in Linux* 14**](#_Toc122649749)

[***2.2.2. Changing Ownership and Group in Linux* 16**](#_Toc122649750)

[PART III: RESULT 18](#_Toc122649751)

[3.1. Write or modify functions related to file management 18](#_Toc122649752)

[***3.1.1. Modify function “cat” in Linux 18***](#_Toc122649753)

[***3.1.2. Write function “ls-group7” same as “ls” command 19***](#_Toc122649754)

[***3.1.3. Write function “cd-group7” same as “cd” command 20***](#_Toc122649755)

[***3.1.4. Address of Linux system commands 21***](#_Toc122649756)

[3.2. Change Linux File Permissions 22](#_Toc122649757)

[PART IV: CONCLUSION 23](#_Toc122649758)

[5.1. Result 23](#_Toc122649759)

[5.2. Limit 23](#_Toc122649760)

[5.3. Development 23](#_Toc122649761)

[REFERENCES 24](#_Toc122649762)

# **LIST OF PICTURES**

[Figure 1. Linux file permission 13](#_Toc122649718)

[Figure 2. Change file permission 15](#_Toc122649719)

[Figure 3. Change file permission 16](#_Toc122649720)

[Figure 4. Changing Ownership 17](#_Toc122649721)

[Figure 5. Changing Group 17](#_Toc122649722)

[Figure 6. Modify function “cat” in Linux 18](#_Toc122649723)

[Figure 7. Source code Modify function “cat” in Linux 18](#_Toc122649724)

[Figure 8. Write function “ls-group7” same as “ls” command 19](#_Toc122649725)

[Figure 9. Source code Write function “ls-group7” same as “ls” command 19](#_Toc122649726)

[Figure 10. Write function “cd-group7” same as “cd” command 20](#_Toc122649727)

[Figure 11. Source code Write function “cd-group7” same as “cd” command 20](#_Toc122649728)

[Figure 12. Address of Linux system commands 21](#_Toc122649729)

[Figure 13. Address of Linux system commands in /bin 21](#_Toc122649730)

[Figure 14. Change access permission 22](#_Toc122649731)

[Figure 15. Changing Ownership and Group in Linux 22](#_Toc122649732)

# **PART I: INTRODUCE**

## 1.1. Topic name

File management - Write or modify functions related to file management. In addition, programming the execution process (reading files, changing file contents, changing permissions, …)

## 1.2. Group Member

* Phạm Thanh Trường – 20IT461
* Phan Văn Bằng - 20IT490
* Phan Văn Lai – 20IT1028
* Hồ Thắng Trình – 20IT511

# **PART II: THEORY**

Linux is a clone of UNIX, the multi-user operating system which can be accessed by many users simultaneously. Linux can also be used in mainframes and servers without any modifications. But this raises security concerns as an unsolicited or malign user can corrupt, change or remove crucial data. For effective security, Linux divides authorization into 2 levels:

* Ownership
* Permission

## 2.1. Linux File Ownership

Every file and directory on your Unix/Linux system is assigned 3 types of owner, given below.

### ***2.1.1. User***

A user is the owner of the file. By default, the person who created a file becomes its owner. Hence, a user is also sometimes called an owner.

### ***2.1.2. Group***

A user- group can contain multiple users. All users belonging to a group will have the same Linux group permissions access to the file. Suppose you have a project where a number of people require access to a file. Instead of manually assigning permissions to each user, you could add all users to a group, and assign group permission to file such that only this group members and no one else can read or modify the files.

### ***2.1.3. Other***

Any other user who has access to a file. This person has neither created the file, nor he belongs to a user group who could own the file. Practically, it means everybody else. Hence, when you set the permission for others, it is also referred as set permissions for the world.

## 2.2. Linux File Permissions

Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners discussed above.

* **Read:** This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.
* **Write:** The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory. Consider a scenario where you have to write permission on file but do not have write permission on the directory where the file is stored. You will be able to modify the file contents. But you will not be able to rename, move or remove the file from the directory.
* **Execute:** In Windows, an executable program usually has an extension “.exe” and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code(provided read & write permissions are set), but not run it.

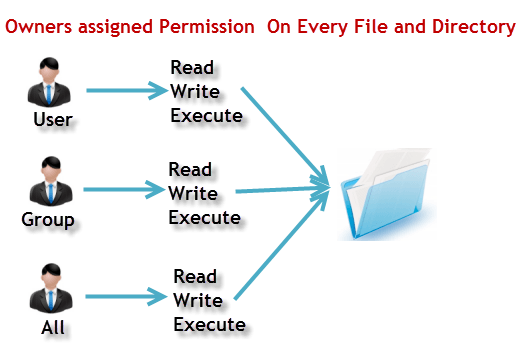


Figure 1. Linux file permission

Syntax show file permissions in Linux:



Here, we have highlighted **‘-rw-rw-r–‘**and this weird looking code is the one that tells us about the Unix permissions given to the owner, user group and the world.

Here, the first ‘**–**‘ implies that we have selected a file.p>



Else, if it were a directory, **d** would have been shown.



**r** = read permission

**w** = write permission

**x** = execute permission

**–** = no permission

### ***2.2.1. Changing file/directory permissions in Linux***

Say you do not want your colleague to see your personal images. This can be achieved by changing file permissions.

We can use the ‘**chmod’** command which stands for ‘change mode’. Using the command, we can set permissions (read, write, execute) on a file/directory for the owner, group and the world

Syntax:

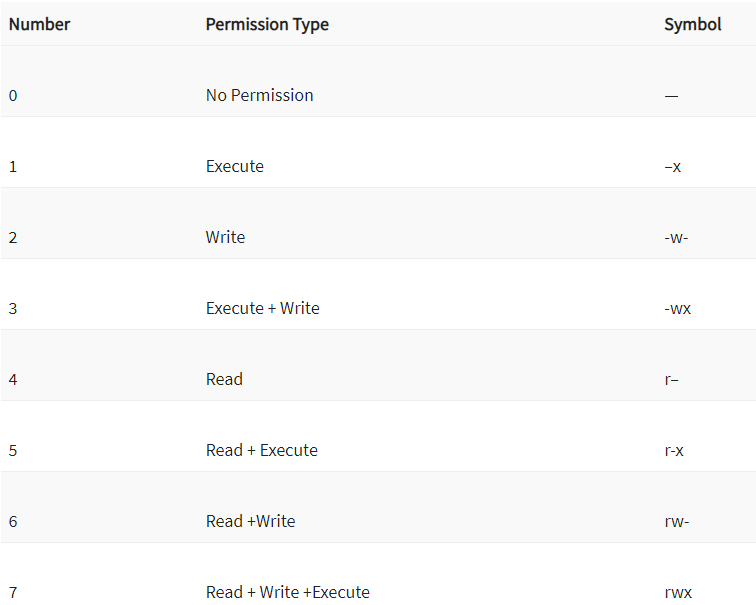


There are 2 ways to use the command:

* Absolute mode
* Symbolic mode

***2.2.1.1. Absolute (Numeric) Mode in Linux***

In this mode, file permissions are not represented as characters but a three-digit octal number. The table below gives numbers for all for permissions types.



Let’s see the chmod permissions command in action:

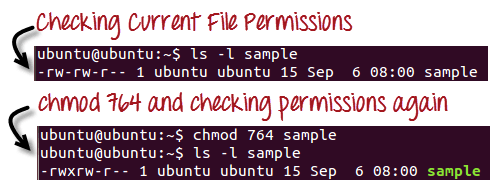
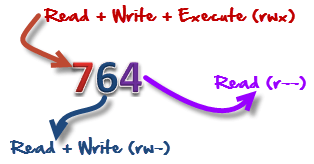


Figure 2. Change file permission

In the above-given terminal window, we have changed the permissions of the file ‘sample to ‘764’



‘764’ absolute code says the following:

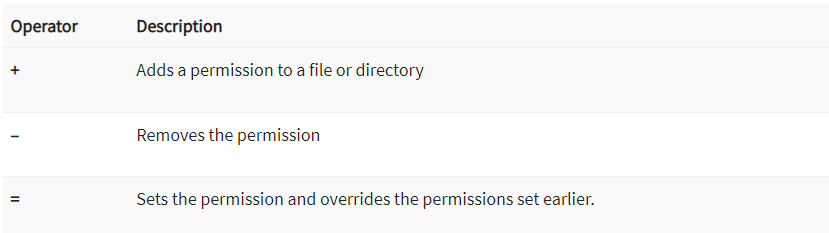
* Owner can read, write and execute
* Usergroup can read and write
* World can only read

**This is shown as ‘-rwxrw-r–**

This is how you can change user permissions in Linux on file by assigning an absolute number.

***2.2.1.2.* Symbolic Mode in Linux**

In the Absolute mode, you change permissions for all 3 owners. In the symbolic mode, you can modify permissions of a specific owner. It makes use of mathematical symbols to modify the Unix file permissions.



The various owners are represented as –



We will not be using permissions in numbers like 755 but characters like rwx. Let’s look into an example:

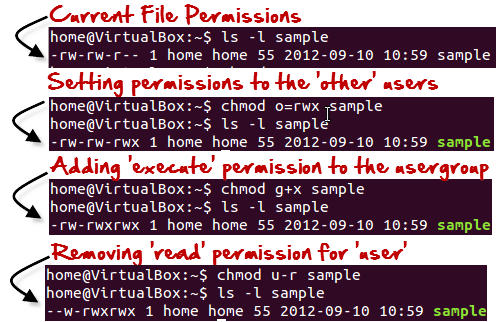


Figure 3. Change file permission

### ***2.2.2. Changing Ownership and Group in Linux***

For changing the ownership of a file/directory, you can use the following command:



In case you want to change the user as well as group for a file or directory use the command:



Let’s see this in action:

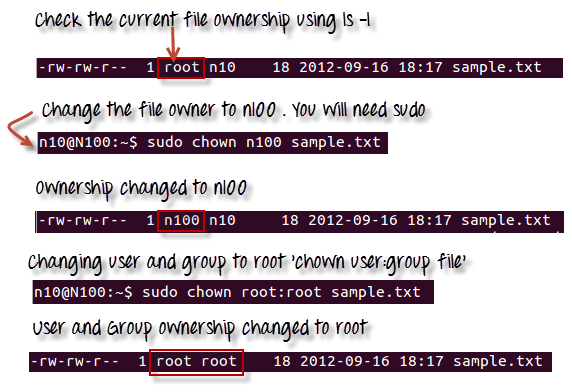


Figure 4. Changing Ownership

In case you want to change group-owner only, use the command:

chgrp group\_name filename

‘**chgrp’** stands for change group.

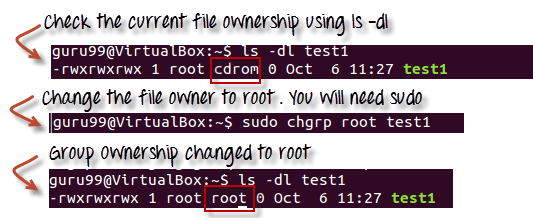


Figure 5. Changing Group

# **PART III: RESULT**

## 3.1. Write or modify functions related to file management

### ***3.1.1. Modify function “cat” in Linux***

Our team changed the "cat" command in the Linux system file. After calling the "cat" command, the terminal still displays the contents of the file, but the bottom also writes the words "-- p/s: cat program by group 7 - ltht"

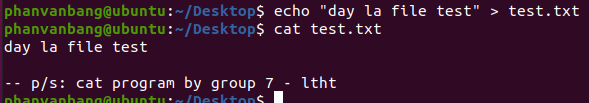
****

Figure 6. Modify function “cat” in Linux

Source code:

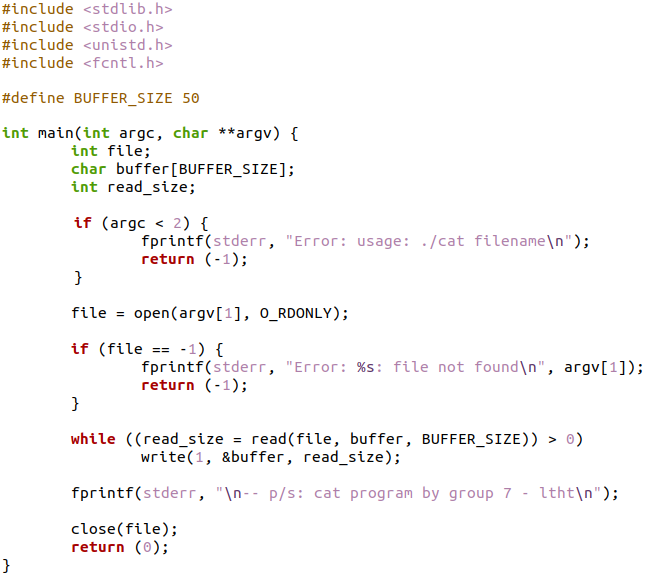


Figure 7. Source code Modify function “cat” in Linux

### ***3.1.2. Write function “ls-group7” same as “ls” command***

Our team wrote the command "ls-group7" which has the same effect as the command "ls" in the Linux operating system. After calling the command "ls-group7", the terminal still displays files and folders, but the bottom also says "-- p/s: ls program by group 7 - ltht"

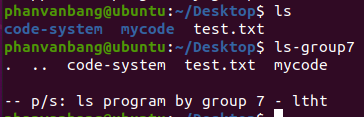


Figure 8. Write function “ls-group7” same as “ls” command

Source code:



Figure 9. Source code Write function “ls-group7” same as “ls” command

### ***3.1.3. Write function “cd-group7” same as “cd” command***

Our team wrote the command "cd-group7" which has the same effect as the command "cd" in the Linux operating system. After calling the command "cd-group7", the terminal displays the path after moving in, but the bottom also says "-- p/s: cd program by group 7 - ltht"

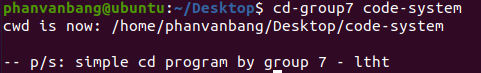


Figure 10. Write function “cd-group7” same as “cd” command

Source code:

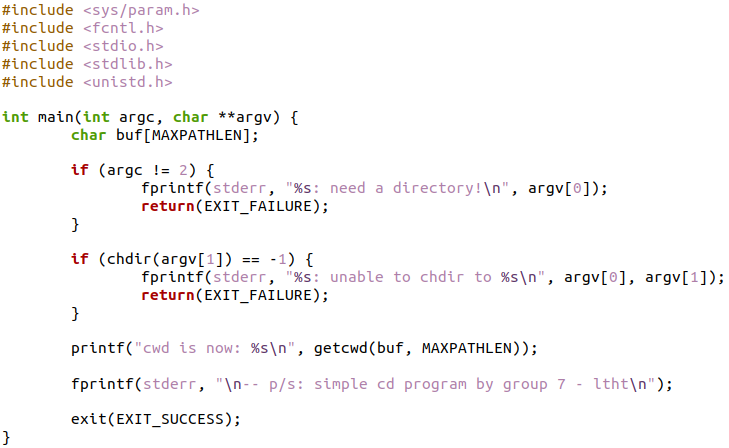


Figure 11. Source code Write function “cd-group7” same as “cd” command

### ***3.1.4. Address of Linux system commands***

The address of system commands in Linux is at /bin

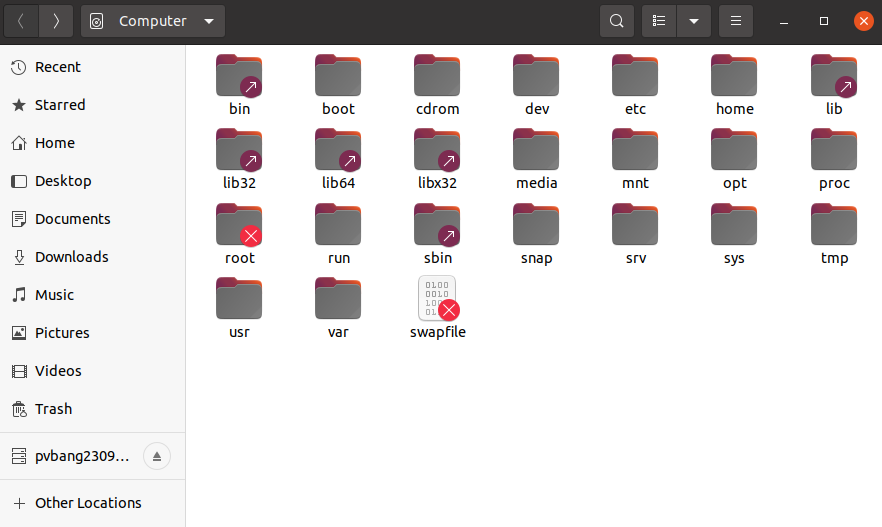


Figure 12. Address of Linux system commands

/bin folder

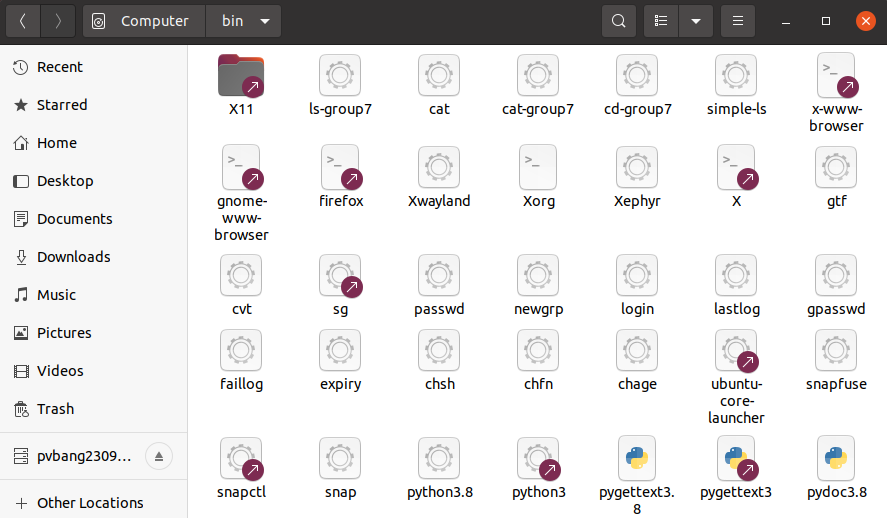


Figure 13. Address of Linux system commands in /bin

## 3.2. Change Linux File Permissions

Change access permission of the file “changing-permissions.txt” to 000. The chmod 000 is not read, not write, not execute.

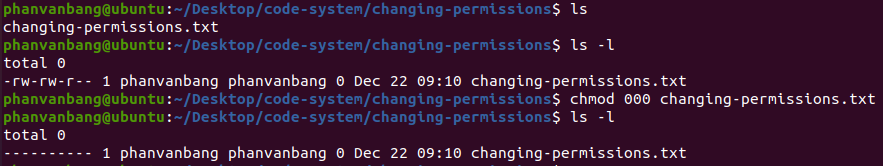


Figure 14. Change access permission

Changing Ownership and Group in Linux:

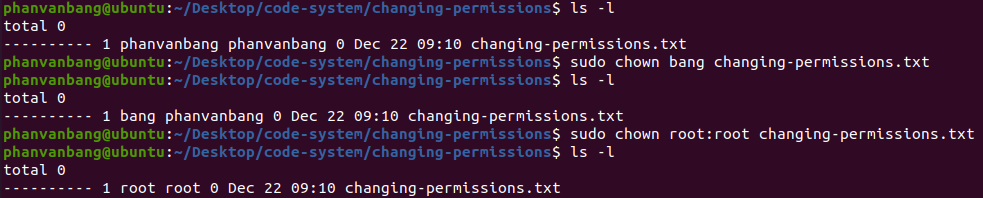


Figure 15. Changing Ownership and Group in Linux

# **PART IV: CONCLUSION**

## 5.1. Result

Understand programming principles and techniques for system programming, basic components, and functions of operating systems.

Write, compile, debug and execute C programs for system programming and properly use system interfaces provided by UNIX (or UNIX-like Operating System).

Organize and work effectively in groups, practice the ability to use English in presenting and reading documents.

## 5.2. Limit

Have not set up and build efficient networked multi-process applications.

## 5.3. Development

Set up and build efficient networked multi-process applications yet.

Learn more about programming principles and techniques for system programming, basic components, and functions of operating systems.

Develop more features for existing and new programs.

# **REFERENCES**

[1] Advanced Programming in the UNIX® Environment, W. Richard Stevens Stephen A. Rago, Safari Books, 2000

[2] The Practice of Programming, Brian W. Kernighan and Rob Pike, O’REILLY, 2014

[3] The\_C\_Programming\_Language, Edition\_Ritchie\_Kernighan, Springer, 2013

[4] *https://www.geeksforgeeks.org/c-program-to-find-and-replace-a-word-in-a-file-by-another-given-word/*

[5] *https://stevens.netmeister.org/765-APUE/*

[6] *https://www.redhat.com/sysadmin/customizing-linux-filesystem-commands*

[7] *https://cheatography.com/davechild/cheat-sheets/linux-command-line/*