“Київський фаховий коледж зв’язку”

Циклова комісія комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №5**

з дисципліни: «Операційні системи»

**Тема: “Знайомство з командами навігації по файловій системі та керування файлами та каталогами”**

Виконав(ла/ли) студент(ка/и)

групи КСМ23-B

Команда velikiebomji:ЛободаД.Р.,

Бережний Т.І. таЧухманенкоМ.В.

Перевірила викладач

Сушанова В.С.

Київ 2024

**Тема: “Знайомство з командами навігації по файловій системі та керування файлами та каталогами”**

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство з базовими командами навігації по файловій системі.
3. Знайомство з базовими командами для керування файлами та каталогами.

**Матеріальне забезпечення занять:**

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows та віртуальна машина Virtual Box (Oracle).

3. ОС GNU/Linux (будь-який дистрибутив).

4. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Короткі теоретичні відомості:**

**Navigating the Filesystem**

In Linux, everything is considered a file. Files are used to store data such as text, graphics, and programs. Directories are a type of file used to store other files; Windows and Mac OS X users typically refer to them as folders. In any case, directories are used to provide a hierarchical organization structure. However, this structure may be somewhat different depending on the type of system in use.

When working in a Linux operating system, it is important to know how to manipulate files and directories. Some Linux distributions have GUI-based applications that allow you to manage files, but it is advantageous to know how to perform these operations via the command line.

On a Windows system, the top level of the directory structure is called My Computer (fig. 5-1). Physical devices, such as hard drives, USB drives, network drives, show up under My Computer and are each assigned a drive letter, such as C: or D:.

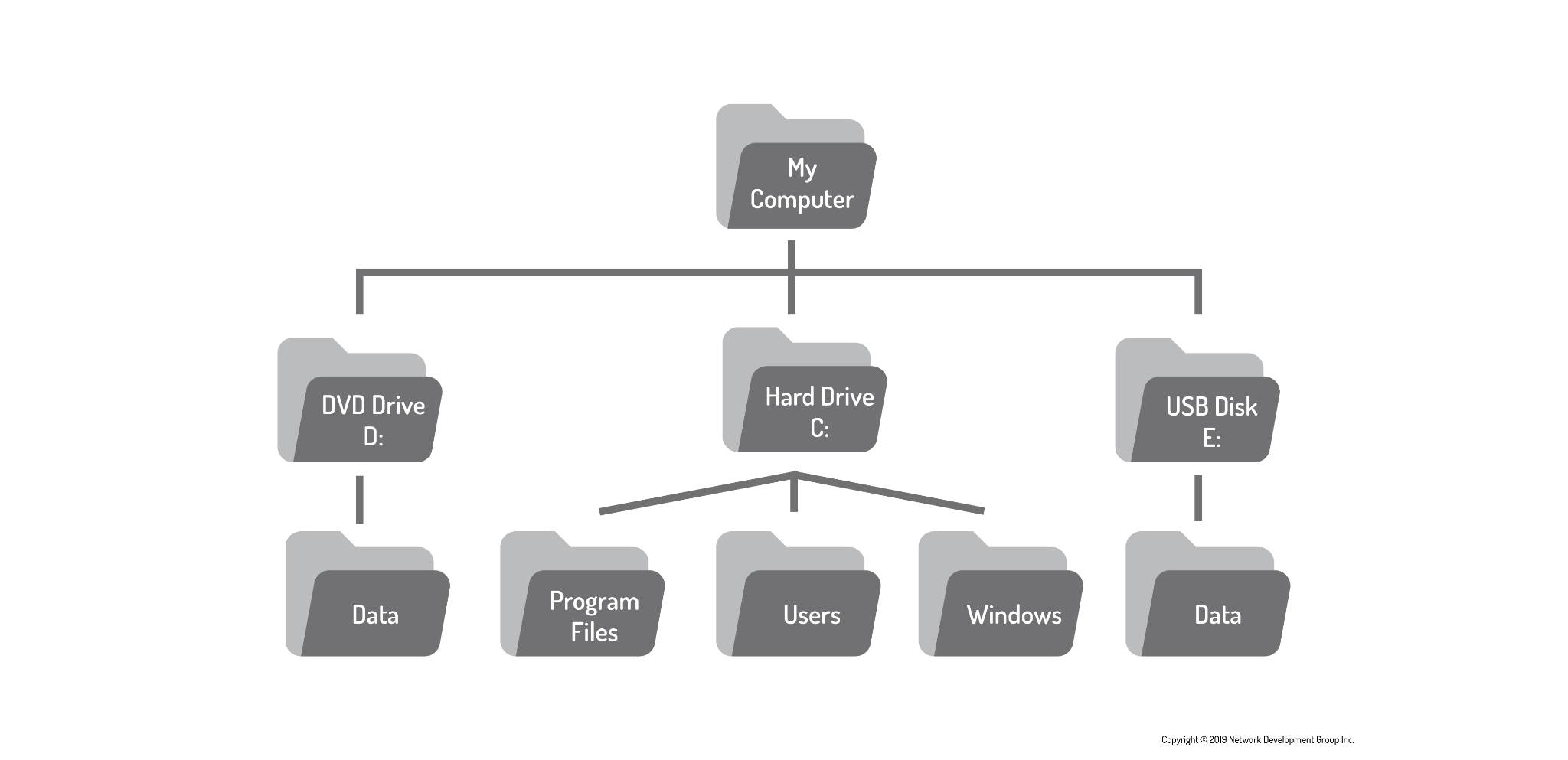


Figure 5-1. A visual representation of a Windows directory structure

Like Windows, the Linux directory structure, typically called a filesystem, also has a top level. However instead of My Computer, it is called the root directory, and it is symbolized by the slash / character (fig. 5-2). Additionally, there are no drives in Linux; each physical device is accessible under a directory, as opposed to a drive letter.

To view the contents of the root directory, use the ***ls*** command with the ***/*** character as the argument

The Linux ﬁlesystem structure originally evolved from the Unix ﬁle structure. In a Linux ﬁlesystem, common directory names are used for common functions. Table 5-1 lists some of the more common Linux virtual top-level directory names and their contents.

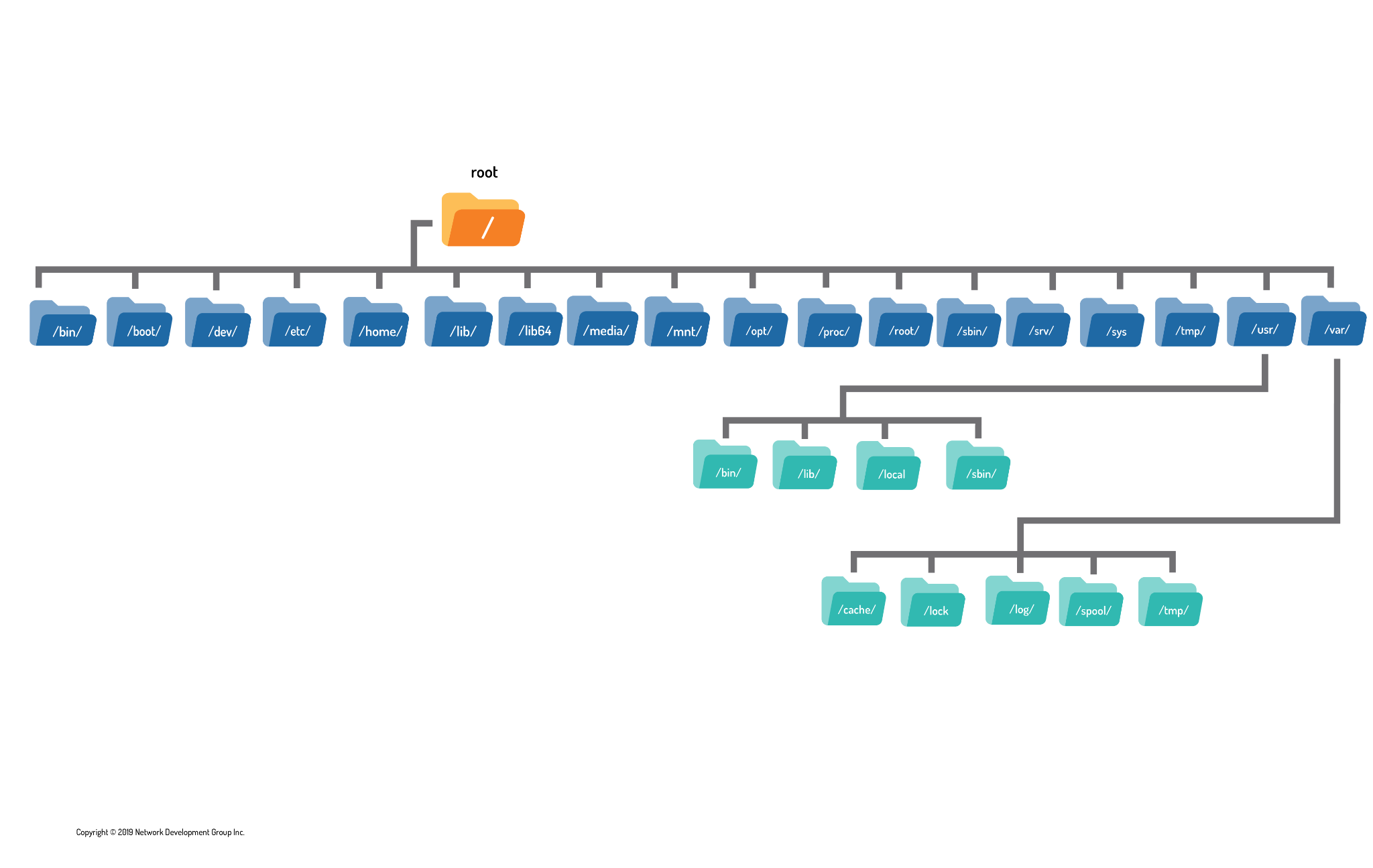


Figure 5-2. A visual representation of a typical Linux filesystem

TABLE 5-1 Common Linux Directory Names

|  |  |
| --- | --- |
| Directory | Usage |
| / | root of the virtual directory, where normally, no files are placed |
| /bin | binary directory, where many GNU user-level utilities are stored |
| /boot | boot directory, where boot files are stored |
| /dev | device directory, where Linux creates device nodes |
| /etc | system configuration files directory |
| /home | home directory, where Linux creates user directories |
| /lib | library directory, where system and application library files are stored |
| /media | media directory, a common place for mount points used for removable media |
| /mnt | mount directory, another common place for mount points used for removable media |
| /opt | optional directory, often used to store third-party software packages and data files |
| /proc | process directory, where current hardware and process information is stored |
| /root | root home directory |
| /sbin | system binary directory, where many GNU admin-level utilities are stored |
| /run | run directory, where runtime data is held during system operation |
| /srv | service directory, where local services store their files |
| /sys | system directory, where system hardware information files are stored |
| /tmp | temporary directory, where temporary work files can be created and destroyed |
| /usr | user binary directory, where the bulk of GNU user-level utilities and data files are stored |
| /var | variable directory, for files that change frequently, such as log files |

The common Linux directory names are based upon the Filesystem Hierarchy Standard (FHS). Many Linux distributions maintain compliance with FHS. Therefore, you should be able to easily ﬁnd ﬁles on any FHS-compliant Linux systems.

When you log in to your system and reach a shell CLI prompt, your session starts in your home directory. Your home directory is a unique directory assigned to your user account. When a user account is created, the system normally assigns a unique directory for the account. You can move around the virtual directory using a graphical interface. However, to move around the virtual directory from a CLI prompt, you need to learn to use the ***cd*** command.

To navigate the filesystem, use the cd (change directory) command.

cd [options] [path]

**Managing Files and Directories**

When working in a Linux Operating System, it is important to know how to manipulate files and directories. Some Linux distributions have GUI-based applications that allow you to manage files, but it is advantageous to know how to perform these operations via the command line.

Globbing

*Glob characters* are often referred to as wild cards. These are symbol characters that have special meaning to the shell.

Globs are powerful because they allow you to specify patterns that match filenames in a directory. So instead of manipulating a single file at a time, you can easily execute commands that affect many files. For instance, by using glob characters, it is possible to manipulate all files with a specific extension or with a particular filename length. Unlike commands that the shell runs, or options and arguments that the shell passes to commands, glob characters are interpreted by the shell itself before it attempts to run any command. As a result, glob characters can be used with any command.

The **asterisk \* character** is used to represent zero or more of any character in a filename. You can use the asterisk character at any place within the filename pattern.

The **question mark ? character** represents any single character. Each question mark character matches exactly one character, no more and no less.

The **bracket [] characters** are used to match a single character by representing a range of characters that are possible match characters. Brackets can also be used to a represent a range of characters.

The **exclamation point ! character** is used in conjunction with the square brackets to negate a range. For example, the pattern /etc/[!DP]\* matches any file that does not begin with a D or P.

Copying Files

The ***cp command*** is used to copy files. It requires a source and a destination. The structure of the command is as follows:

cp [source] [destination]

The source is the file to be copied. The destination is where the copy is to be located. When successful, the cp command does not have any output (no news is good news). The **-v option** causes the cp command to produce output if successful. When the destination is a directory, the resulting new file keeps the same name as the original file. To give the new file a different name, provide the new name as part of the destination. The cp command can be destructive to existing data if the destination file already exists. In the case where the destination file exists, the cp command overwrites the existing file's contents with the contents of the source file. By default, the cp command will not copy directories; any attempt to do so results in an error message. However, the recursive **-r option** allows the cp command to copy both files and directories:

cp -r [source\_directory] [destination\_directory]

Moving Files

To move a file, use the ***mv command***. The syntax for the mv command is much like the cp command:

mv [source] [destination]

When a file is moved, the file is removed from the original location and placed in a new location. Moving files can be somewhat tricky in Linux because users need specific permissions to remove files from a directory. Without the right permissions, a Permission denied error message is returned. The mv command is not just used to move a file, but also to rename a file. If the destination for the mv command is a directory, the file is moved to the directory specified. The name of the file only changes if a destination file name is also specified. If a destination directory is not specified, the file is renamed using the destination file name and remains in the source directory. Like the cp command, the mv command provides the following options:

| Option | Meaning |
| --- | --- |
| -i | Interactive: Ask if a file is to be overwritten. |
| -n | No Clobber: Do not overwrite a destination file's contents. |
| -v | Verbose: Show the resulting move. |

Creating Files

There are several ways of creating a new file, including using a program designed to edit a file (a text editor).

There is also a way to create an empty file that can be populated with data at a later time. This feature is useful for some operating systems as the very existence of a file could alter how a command or service works. It is also useful to create a file as a "placeholder" to remind you to create the file contents at a later time. To create an empty file, use the ***touch command***. Notice the size of the new file is 0 bytes. As previously mentioned, the touch command doesn't place any data within the new file

Removing Files

To delete a file, use the ***rm command***. Note that the files were deleted with no questions asked. This could cause problems when deleting multiple files by using glob characters. Because these files are deleted without question, a user could end up deleting files that were not intended to be deleted. As a precaution, users should use the **-i option** when deleting multiple files.

You can delete directories using the rm command. However, the default behavior (no options) of the rm command is to not delete directories. To delete a directory with the rm command, use the **-r recursive option.**

You can also delete a directory with the ***rmdir command***, but only if the directory is empty.

Creating Directories

To create a directory, use the ***mkdir command***

**Завдання для попередньої підготовки:**

1. \*Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань призначення команд та їх параметрів.
2. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   1. Порівняйте файлові структури Windows-подібної та Linux-подібної системи.
   2. \*Розкрийте поняття FHS. Як даний стандарт використовується в контексті файлових систем?
   3. \*\*Перерахуйте основні команди для роботи з файлами та каталогами в Linux: створення, переміщення, копіювання, видалення.
3. Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:

* Chapter 7 - Navigating the Filesystem
* Chapter 8 - Managing Files and Directories

1. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:

* Chapter 07 Exam
* Chapter 08 Exam

1. Підготувати в електронному вигляді початковий варіант звіту:

* Титульний аркуш, тема та мета роботи
* Словник термінів
* Відповіді на п.2.1-2.3 з завдань для попередньої підготовки

**Хід роботи:**

* 1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
  2. Запустіть операційну систему Linux Ubuntu. Виконайте вхід в систему та запустіть термінал ***(якщо виконуєте ЛР у 401 ауд.)***.
  3. Запустіть віртуальну машину Ubuntu\_PC ***(якщо виконуєте завдання ЛР через академію netacad)***
  4. Запустіть свою операційну систему сімейства Linux ***(якщо працюєте на власному ПК та її встановили)*** та запустіть термінал.
  5. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу ***NDG Linux Essentials - Lab 7: Navigating the Filesystem*** та ***Lab 8: Managing Files and Directories.*** Створіть таблицю для опису цих команд
  6. (Лобода)

|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| pwd | Визначає місце знаходження користувача у файловій системі, показує поточну робочу директорію (print working directory) |
| cd Documents | Команда **cd** здійснює перехід до каталогу, який у неї вказаний як аргумент. В даному випадку це каталог **Documents** |
| gzip | Стискання файлу |
| gunzip | Стискання файлу до початкової форми |
| tar | Архівувіння файлів |
| -t | Перерахувати файли в архіві |
| mkdir | Створює нову дерикторію |
| сolor | Змінює колір |
| sort | змінити порядок виведення на основі вмісту одного або кількох полів |
| head | відображення лише кількох перших або останніх рядків файлу |
| less | Передає назву файлу як агрумент |
| more | Переміщення документом за допомогою клавіщ |
| unzip | Список файлів у архівах |
| rmdir | Видалити каталог |
| rm | Видалити файл |
| mv | Переносить файл |
| cat | поверніться до домашнього каталогу користувача; |
| cp | Копіює файл або каталог (-r копіює каталог з його змістом) |
| -r |  |

**Примітка:** **Скріншоти** виконання команд в терміналі можна **не представляти**, достатньо **коротко описати команди в таблиці**.

* 1. Робота в в терміналі (закріплення практичних навичок) **обов'язково представити свої скріншоти**:
* Визначте ваш поточний робочий каталог;
* Перейдіть до кореневого каталогу та визначте Ваш поточний робочий каталог (дві команди);
* Перегляньте вміст поточного каталогу у довгому форматі (скористайтесь відповідним ключем команди ls);
* Перейдіть до каталогу /usr/share та визначте Ваш поточний робочий каталог (дві команди)
* Перегляньте вміст поточного каталогу включаючи і приховані файли (hidden files) (скористайтесь відповідним ключем команди ls);
* \*Перейдіть до каталогу /etc;
* \*Перегляньте вміст даного каталогу, але щоб виводило тільки назви файлів, що починаються з літери вашого імені;
* \*Перегляньте вміст даного каталогу, але щоб виводило тільки файли, назви яких складаються з 6 літер;
* \*\*Перегляньте вміст даного каталогу, але щоб виводило тільки файли, назви яких закінчуються на літери ваших імен, наприклад якщо ваші імена Ivan, Anna, Maks, то вибірку робиму, щоб назви файлів закінчувались на літери [i,a,m];
* \*\*Перейдіть до домашнього каталогу поточного користувача та перегляньте його вміст у рекурсивному (зворотному до алфавітного) форматі (виконати цю дію через конвеєр команд);
* В поточній директорії створити директорію з назвою вашої групи;
* Переглянути оновлений вміст домашнього каталогу поточного користувача. Скористайтесь ключем -r команди ls, яку інформацію ви отримаєте?
* Перейдіть у створену вами директорію з назвою Вашої групи та створіть у ній порожній файл *lab5*
* Створити в даній директорії 3 директорії з прізвищами студентів вашої команди *surname1, surname2, surname3* (команда mkdir мульти аргумента, тому всі три каталоги можна створити однією командою);
* Перейдіть у перший підкаталог *surname1* та створіть порожній файл з ім'ям першого студента *name1*;
* За допомогою команди *echo "Hello, my name is Name1" > name1* внесіть у цей файл дані про студента (символ *>* дозволяє вивід команди *echo* перенаправити одразу у файл *name1*;
* Перегляньте вміст файлу *name1* за допомогою команди *cat name1* (має містити щойно введену Вами інформацію)
* Зробіть копію першого файлу *name1* та перейменуйте її у файл з другим ім'ям студенту Вашої команди *name2*;
* Перегляньте вміст каталогу, обидва файли мають з'явитися;
* Перегляньте вміст другого файлу *cat name2* (він має поки що містити повну копію вмісту файлу *name1*)
* Замініть зміст файлу name2, щоб він містив відповідне ім'я другого студента за допомогою команди *echo "Hello, my name is Name2" > name2*
* Перегляньте вміст другого файлу *cat name2* (він вже має містити оновлену інформацію)
* Перемістіть файл *name2* у директорію *surname2*;
* Зробіть копію першого файлу *name1* та перейменуйте її у файл з третім ім'ям студенту Вашої команди *name3*;
* Перемістіть файл *name3* у директорію *surname3*;
* Перейдіть до директорії *surname3;*
* Перегляньте вміст третього файлу командою *cat name3* (він має містити дані про другого студента)
* Замініть зміст файлу name3, щоб він містив відповідне ім'я третього студента за допомогою команди *echo "Hello, my name is Name3" > name3*
* Перегляньте вміст файлу за допомогою *cat name3* (він вже має містити оновлену інформацію)
* Поверніться до домашнього каталогу користувача;
* \*\*Перегляньте вміст даного каталогу, але щоб виводило тільки Ваш підкаталог з назвою групи та весь його вміст (підкаталоги *surname1, surname2, surname3* тафайли *name1, name2, name3*) до того ж файли та катлоги були відкоремлені кольорами (скористайтесь відповідним ключем -R команди ls та не забудьте використати спеціальний glob-шаблон [імя каталогу])

**Примітка:** Назви підкаталогів *surname1, surname2, surname3* та файлів *name1, name2, name3* замініть на свої

1. Опишіть дії, які виконують команди для переміщення по системі каталогів: (Лобода)

* команда cd / -- перехід до кореневої дерикторії.
* команда cd /home -- перехід до дерикторії.
* команда cd ~ -- перехід до домашньої директорії поточного користувача.
* команда cd (без аргумента) --  перехід до домашньої директорії.
* команда cd .. -- перехід до батьківської дерикторії.
* команда cd ../.. -- перехід на два рівні вгору по дереву директорій.
* команда cd - -- повернення до попередньої директорії.

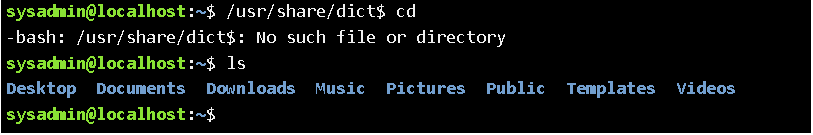
**Контрольні запитання: (Бережний)**

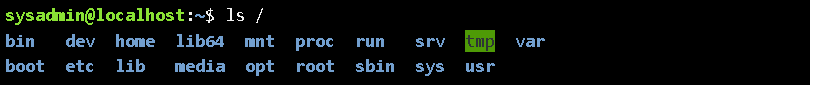
1. Як можна переглянути шлях до домашньої директорії користувача за допомогою команди echo? Існує 2 способи, наведіть обидва приклади у терміналі (відповідь є у матеріалах академії cisco на сайті netacad.com)





1. \*Чи можна переглянути вміст кореневого каталогу, перебуваючи у домашньому каталозі користувача без переходу у кореневий каталог? Продемонструйте це в командному рядку.





1. \*Яким чином в терміналі можна додати інформацію в порожній файл?

Щоб додати інформацію в порожній файл у терміналі, ми використовуємо команду:

echo "текст" > ім'я\_файлу

1. \*\*Як скопіювати та видалити існуючий каталог? Чи буде відмінність в командах, якщо каталог буде не порожній при цьому

Щоб скопіювати каталог, використовують команду: cp -r ім'я\_каталогу нове\_місце

Відмінність є, якщо працюємо в пустому каталозі, то потрібно писати -r

1. \*\*У якому з наведених нижче прикладів відбувається переміщення файлу? його перейменування? одночасно обидві дії?

* mv /work/tech/comp.png. /Desktop – переміщення файлу
* mv /work/tech/comp.png. /work/tech/my\_car.png – перейменування файлу
* mv /work/tech/comp.png. /Desktop/computer.png – переміщення та перейменування файлу