

**Тема: “Знайомство з базовими командами CLI-режиму в Linux”****Мета роботи:**

1. Знайомство з базовими командами CLI-режиму в Linux.
2. Знайомство з базовими текстовими командами в термінальному режимі роботи в різних ОС.

**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:****Command Line Skills**The Shell

Once a user has entered a command the terminal then accepts what the user has typed and passes to a shell.

The CLI provides more precise control, greater speed and the ability to automate tasks more easily through scripting.

The shell is the command line interpreter that translates commands entered by a user into actions to be performed by the operating system.

The Linux environment allows the use of many different shells.

The most commonly used shell for Linux distributions is called the Bash shell.

The Bash shell also has many popular features, a few of which are listed below:

- Command line history
- Inline editing
- Scripting: The ability to place commands in a file and then interpret (effectively use Bash to execute the contents of) the file, resulting in all of the commands being executed.
- Aliases: The ability to create short nicknames for longer commands.
- Variables: Used to store information for the Bash shell and for the user.

When a terminal application is run, and a shell appears, displaying an important part of the interface — the prompt.

Typically the prompt contains information about the user and the system. Below is a common prompt structure:

```
sysadmin@localhost:~$
```

The prompt shown contains the following information:

- Username (sysadmin)
- System name (localhost)
- Current Directory (~)

The ~ symbol is used as a shorthand for the user's home directory. Typically the home directory for the user is under the /home directory and named after the user account name; for example, /home/sysadmin.

Commands

A command is a software program that when executed on the CLI, performs an action on the computer.

To execute a command, the first step is to type the name of the command.

If you type ls and hit Enter. The result should resemble the example below:

```
sysadmin@localhost:~$ ls
```

```
Desktop Documents Downloads Music Pictures Public Templates Videos
```

Some commands require additional input to run correctly. This additional input comes in two forms: options and arguments. Options are used to modify the core behavior of a command while arguments are used to provide additional information (such as a filename or a username). Each option and argument is normally separated by a space, although options can often be combined.

The typical format for a command is as follows:

```
command [options] [arguments]
```

Keep in mind that Linux is case-sensitive. Commands, options, arguments, variables, and file names must be entered exactly as shown.

### Arguments

An argument can be used to specify something for the command to act upon.

If the `ls` command is given the name of a directory as an argument, it lists the contents of that directory.

```
sysadmin@localhost:~$ ls /etc/ppp  
ip-down.d ip-up.d
```

Some commands (such as `ls`) accept multiple arguments:

```
sysadmin@localhost:~$ ls /etc/ppp /etc/ssh  
/etc/ppp:  
ip-down.d ip-up.d  
/etc/ssh:  
moduli          ssh_host_dsa_key.pub  ssh_host_rsa_key  sshd_configssh_config  
ssh_host_ecdsa_key  ssh_host_rsa_key.pub  ssh_host_dsa_key  ssh_host_ecdsa_key.pub  ssh_import_id
```

### Options

Options can be used with commands to expand or modify the way a command behaves.

For example, using the `-l` option of the `ls` command results in a long listing, providing additional information about the files that are listed:

```
sysadmin@localhost:~$ ls -l  
total 0  
drwxr-xr-x 1 sysadmin sysadmin 0 Jan 29 2015 Desktop  
drwxr-xr-x 1 sysadmin sysadmin 0 Jan 29 2015 Documents  
Output Omitted...
```

Often the character is chosen to be mnemonic for its purpose, like choosing the letter `l` for long or `r` for reverse. Options can be used in conjunction with other options, for example:

```
sysadmin@localhost:~$ ls -l -r  
sysadmin@localhost:~$ ls -lr  
sysadmin@localhost:~$ ls -rl
```

The order of the combined options isn't important. The output of all of these examples would be the same.

Options are often single letters; however, sometimes they are words or phrases as well.

Typically, older commands use single letters while newer commands use complete words for options.

Single-letter options are preceded by a single dash `-` character, like the `-h` option.

Full-word options are preceded by two dash `--` characters like the full-word form of the `-h` option, the `--human-readable` option.

### Variables

A variable is a feature that allows the user or the shell to store data. This data can be used to provide critical system information or to change the behavior of how the Bash shell (or other commands) work. Variables are given names and stored temporarily in memory. There are two types of variables used in the Bash shell: local and environment.

Command Types

The type command can be used to determine information about command type:

type command

There are several different sources of commands within the shell of your CLI:

- Internal commands also called built-in commands, these commands are built into the shell itself;
- External commands are stored in files that are searched by the shell;
- Aliases can be used to map longer commands to shorter key sequences;
- Functions can also be built using existing commands to create new commands or override commands built-in to the shell or commands stored in files.

Quoting

Quotation marks are used throughout Linux administration and most computer programming languages to let the system know that the information contained within the quotation marks should either be ignored or treated in a way that is very different than it would normally be treated. There are three types of quotes that have special significance to the Bash shell: double quotes `"`, single quotes `'`, and back quotes ```. Each set of quotes alerts the shell not to treat the text within the quotes in the normal way.

Control statements

Control statements allow you to use multiple commands at once or run additional commands.

Control statements include:

- Semicolon (`;`)
- Double ampersand (`&&`)
- Double pipe (`||`)

**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. Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:
  - Chapter 5 - Command Line Skills
  - Chapter 6 - Getting Help
3. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:
  - Chapter 05 Exam
  - Chapter 06 Exam
4. \*Дайте визначення наступним поняттям:
  - Командний інтерпретатор
  - Оболонка
  - Команда
5. \*\*Дайте відповіді на наступні питання:
  - Яку базову інформацію надає рядок запрошення `prompt`?
  - Для чого команді потрібні параметри та аргументи?
  - Яке призначення команд `ls`, які параметри та аргументи вона може мати? Наведіть 3 приклади.
  - Яким чином можна використати історію команд, які переваги це надає?
  - Яке призначення команди `echo`?
  - Охарактеризуйте поняття змінної в оболонці Bash, які типи змінних вона підтримує?
  - Яке призначення команд `env`, `export` та `unset`?
  - Які команди для отримання довідки по командам в терміналі ви знаєте?
6. Підготувати в електронному вигляді початковий варіант звіту:
  - Титульний аркуш, тема та мета роботи
  - Словник термінів
  - Відповіді на п.4 та п.5 з завдань для попередньої підготовки

**Хід роботи:**

- Опрацюйте всі приклади команд, що представлені у лабораторній роботі курсу *NDG Linux Essentials - Lab 5: Command Line Skills* та *Lab 6: Getting Help*. Створіть таблицю для опису цих команд

| Назва команди | Її призначення та функціональність  |
|---------------|---|
| ls            | Виводить інформації про каталоги та файли. За замовчуванням без аргументів відображає інформацію для поточного каталогу   |
| ls -l         | Використанні параметру <b>-l</b> в команді <b>ls</b> дозволяє відобразити інформацію про файли, розташовані в поточному робочому каталозі, у довгому форматі, який надає більш розширену додаткову інформацію |
| ls -l /tmp    | Використання аргументу <b>/tmp</b> в поєднанні з параметром <b>-l</b> в команді <b>ls</b> дозволяє відобразити детальну інформацію про файли в каталозі <b>/tmp</b> .   |
|               |   |
|               |   |
|               |   |

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

- Робота в в терміналі (закріплення практичних навичок) **обов'язково представити свої скріншоти:**
  - Робота зі змінними (Variables) та псевдонімами (Aliases) в терміналі:
    - Створіть змінні, що будуть містити Ваші імена та прізвища \$var\_name1, \$var\_name2, \$var\_name3
    - За допомогою команди echo виведіть імена студентів вашої команди
    - Створіть псевдоніми mycal1, mycal2, mycal3 для команди cal для автоматичного виведення календарю вашого року народження
  - \*Робота з функціями (Functions) в терміналі:
    - Створіть функцію students\_report, що порядково буде виводити спочатку імена студентів Вашої команди, а потім роки їх народження
  - \*Робота з лапками (Quoting) в терміналі. Виведіть в командному рядку наступні речення:
    - “We create such variables as \$var\_name1, \$var\_name2, \$var\_name3, which stored our names Name1, Name2, Name3” (у реченні спочатку виводимо назви змінних, а потім їх вміст)
    - “We create such Aliases as mycal1, mycal2, mycal3, which can show our calendars: Calendar1, Calendar2, Calendar3” (у реченні спочатку виводимо назву команди-псевдонімів, потім вивід цих команд).
  - \*\*Робота з інструкціями керування (Control Statements) в терміналі:
    - Чи можна завдання 2.1 та 2.2 ходу роботи виконати через інструкції керування без написання окремої функції, як це буде виглядати?
  - Робота з командами довідки (Man Pages) в терміналі:
    - На прикладі команди uname продемонструйте як отримати довідку. На основі отриманої додаткової інформації наведіть 5 різних варіантів виводу результату інформації по даній команді з використанням 5 різних параметрів (Options)

**Контрольні запитання:**

- Які типи команд існують в оболонці Bash?
- Що таке змінні оточення? Які вони бувають. Як їх можна переглянути в терміналі?
- \*Опишіть змінну \$PS1. Як в терміналі переглянути її вміст?
- \*Як можна змінити значення змінної \$PS1? Що при цьому відбудеться в рядку запрошенні в bash (рядок запрошення перед початком кожної команди). Як змінити значення цієї змінної не на поточний сеанс, а за замовчуванням?
- \*Для чого використовують лапки в оболонці Bash? .
- \*\*Для чого використовують інструкції керування, які їх види Ви знаєте?
- \*\*В чому різниця якщо в кінці рядку запрошення bash стоїть символ \$ чи #? Наприклад на екрані ми бачимо наступні записи

```
[centos@localhost Desktop]$ █
```

```
[root@localhost Desktop]#
```

- \*\*Яке призначення команд whereis та locate? Яка між ними відмінність?

### Оформлення звіту:

1. Титульний аркуш
2. Тема та мета роботи
3. Завдання попередньої підготовки
4. Основні позиції ходу роботи
5. Відповіді на контрольні запитання
6. Висновки за результатами роботи (**обов'язково!!!**)

### Система оцінювання лабораторної роботи:

Виконано завдання базового рівня складності - **3 бали**

Виконано завдання базового та середнього рівня складності - **4 бали**

Виконано завдання всіх рівнів складності (в тому числі й підвищеного) - **5 балів**

Завдання середнього рівня складності позначені в завданнях (\*)

Завдання підвищеного рівня складності позначені в завданнях (\*\*)

**Примітка:** за виконання робіт в командах та оформлення звітів з використанням системи контролю версій (git) та англійської мови може бути нараховано **додатковий 1 бал**.