**Тема: “Захист системи та користувачів у Linux. Створення користувачів та груп”**

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство з базовими діями при створенні нових користувачів та нових груп користувачів.

**Матеріальне забезпечення занять**

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows (Windows 7).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

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

**Короткі теоретичні відомості:**

**System and User Security**

Administrative Accounts

There are many different ways to execute a command that requires administrative or root privileges. Logging in to the system as the root user allows you to execute commands as the administrator. This access is potentially dangerous because you may forget that you are logged in as root and might run a command that could cause problems on the system. As a result, it is not recommended to log in as the root user directly.

Because using the root account is potentially dangerous, you should only execute commands as root if administrative privileges are needed. If the root account is disabled, as it is on the Ubuntu distribution, then administrative commands can be executed using the *sudo* command. If the root account is enabled, then a regular user can execute the *su* command to switch accounts to the root account.

When you log in to the system directly as root to execute commands, then everything about your session runs as the root user. If using the graphical environment, this is especially dangerous as the graphical login process is comprised of many different executables (programs that run during login). Each program that runs as the root user represents a greater threat than a process run as a standard user, as those programs would be allowed to do nearly anything, whereas standard user programs are very restricted in what they can do.

The other potential danger with logging into the system as root is that a person that does this may forget to log out to do their non-administrative work, allowing programs such as browsers and email clients to be run as the root user without restrictions on what they could do. The fact that several distributions of Linux, notably Ubuntu, do not allow users to log in as the root user should be enough indication that this is not the preferred way to perform administrative tasks.

User Accounts

There are several text files in the */etc* directory that contain the account data of the users and groups defined on the system. For example, to see if a specific user account has been defined on the system, then the place to check is the */etc/passwd* file.

The /etc/passwd file defines some of the account information for user accounts. Each line contains information pertaining to a single user. The data is separated into fields by colon characters.

System Accounts

Users log into the system using regular user accounts. Typically, these accounts have UID values of greater than 500 (on some systems 1,000). The root user has special access to the system. This access is provided to the account with a UID of 0.

There are additional accounts that are not designed for users to log into. These accounts, typically from UID 1 to UID 499, are called system accounts, and they are designed to provide accounts for services that are running on the system.

System accounts have some fields in the /etc/passwd and /etc/shadow files that are different than other accounts. For example, system accounts rarely have home directories as they typically are not used to create or store files.

Group Accounts

Your level of access to a system is not determined solely by your user account. Each user can be a member of one or more groups, which can also affect the level of access to the system.

Traditionally, UNIX systems limited users to belonging to no more than a total of sixteen groups, but the recent Linux kernels support users with over sixty-five thousand group memberships.

The /etc/passwd file defines the primary group membership for a user. Supplemental group membership (or secondary group membership) and the groups themselves are defined in *the /etc/group* file.

**Viewing User Information**

The id command is used to print user and group information for a specified user.

*id [options] username*

When switching between different user accounts, it can be confusing as to which account is currently logged in. When executed without an argument, the id command outputs information about the current user, allowing you to confirm your identity on the system.

The *who* command displays a list of users who are currently logged into the system, where they are logged in from, and when they logged in. Through the use of options, this command is also able to display information such as the current runlevel (a functional state of the computer) and the time that the system was booted.

The *last* command reads the entire login history from the /var/log/wtmp file and displays all logins and reboot records by default. An interesting detail of the reboot records is that it displays the version of the Linux kernel that was booted instead of the login location. The /var/log/wtmp file keeps a log of all users who have logged in and out the system.

**Introduction**

During the installation process, most installers create a normal user and either give this user the permission to execute administrative commands with sudo or require the root user account password be configured as part of the installation process. Most Linux systems are configured to allow for one unprivileged (non-root) user to log in, as well as have the ability to effectively execute commands as the root user, either directly or indirectly.

If the computer is to be used by only one person, then having only one regular user account might be sufficient. However, if a computer needs to be shared by multiple people, then it is desirable to have a separate account for each person who uses it. There are several advantages to individuals having their own separate accounts:

Accounts can be used to grant selective access to files or services. For example, the user of each account has a separate home directory that is generally not accessible to the other users.

The sudo command can be configured to grant the ability to execute select administrative commands. If users are required to use the sudo command to perform administrative commands, then the system logs when users perform these commands.

Each account can have group memberships and rights associated with it allowing for greater management flexibility.

On some distributions, creating a new user account also automatically creates a group account for the user, called a User Private Group (UPG). On these systems, the group and username would be the same, and the only member of this new group would be the new user.

For distributions that do not create a UPG, new users are typically given the users group as their primary group. The administrator can manually create group accounts that are private for the user, but it's more common for the administrator to create groups for multiple users that need to collaborate. User accounts can be modified at any time to add or remove them from group account memberships, but users must belong to at least one group for use as their primary group.

Before you begin creating users, you should plan how to use groups. Users can be created with memberships in groups that already exist, or existing users can be modified to have memberships in existing groups.

If you already have planned which users and groups you want, it is more efficient to create your groups first and create your users with their group memberships. Otherwise, if you create your users first, and then your groups, you'll need to take an extra step to modify your users to make them members of your groups.

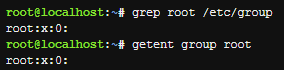
The most common reason to create a group is to provide a way for users to share files. For example, if several people who work together on the same project and need to be able to collaborate on documents stored in files for the project. In this scenario, the administrator can make these people members of a common group, change the directory ownership to the new group and set permissions on the directory that allows members of the group to access the files.

After creating or modifying a group, you can verify the changes by viewing the group configuration information in the /etc/group file with the grep command. If working with network-based authentication services, then the getent command can show you both local and network-based groups.

grep pattern filename

getent database record

For local usage, these commands show the same result, in this case for the root group:



**Creating A Group**

The groupadd command can be executed by the root user to create a new group. The command requires only the name of the group to be created. The -g option can be used to specify a group id for the new group:



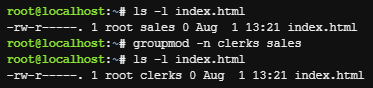
If the -g option is not provided, the groupadd command will automatically provide a GID for the new group. To accomplish this, the groupadd command looks at the /etc/group file and uses a number that is one value higher than the current highest GID number. The execution of the following commands illustrates this:



Modifying a Group

The groupmod command can be used to either change the name of a group with the -n option or change the GID for the group with the -g option.

Changing the name of the group may confuse users who were familiar with the old name and haven't been informed of the new name. However, changing the group name won't cause any problems with accessing files, since the files are owned by GIDs, not group names. For example:



Note: The file in the example above is not available within the virtual machine environment of this course.

After the previous groupmod command, the index.html file has a different group owner name. However, all users who were in the sales group are now in the clerks group, so all of those users can still access the index.html file. Again, this is because the system defines the group by the GID, not the group name.

On the other hand, if you change the GID for a group, then all files that were associated with that group will no longer be associated with that group. In fact, all files that were associated with that group will no longer be associated with any group name. Instead, these files will be owned by a GID only, as shown below:

**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 15 - System and User Security
* Chapter 16 - Creating Users and Groups

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

* Chapter 15 Exam
* Chapter 16 Exam

1. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   1. Розкрийте поняття UPG, коли їх доцільно використовувати?1

Встановлення нових програм: UPG просто дозволяє встановлювати нові програми на комп'ютері з певними залежностями, що автоматично вирішуються системою.

Оновлення програм: UPG надає можливість оновлювати встановлені програми до нових версій, якщо вони вони звісно ж стають доступними, забезпечуючи безпеку та стабільність системи.Також UPG дозволяє просто видаляти непотрібні програми

* 1. Якими командами можна створити групи користувачів? Наведіть приклади2
  2. Якими командами можна змінити налаштування груп користувачів? Наведіть приклади3

1. Підготувати в електронному вигляді початковий варіант звіту:

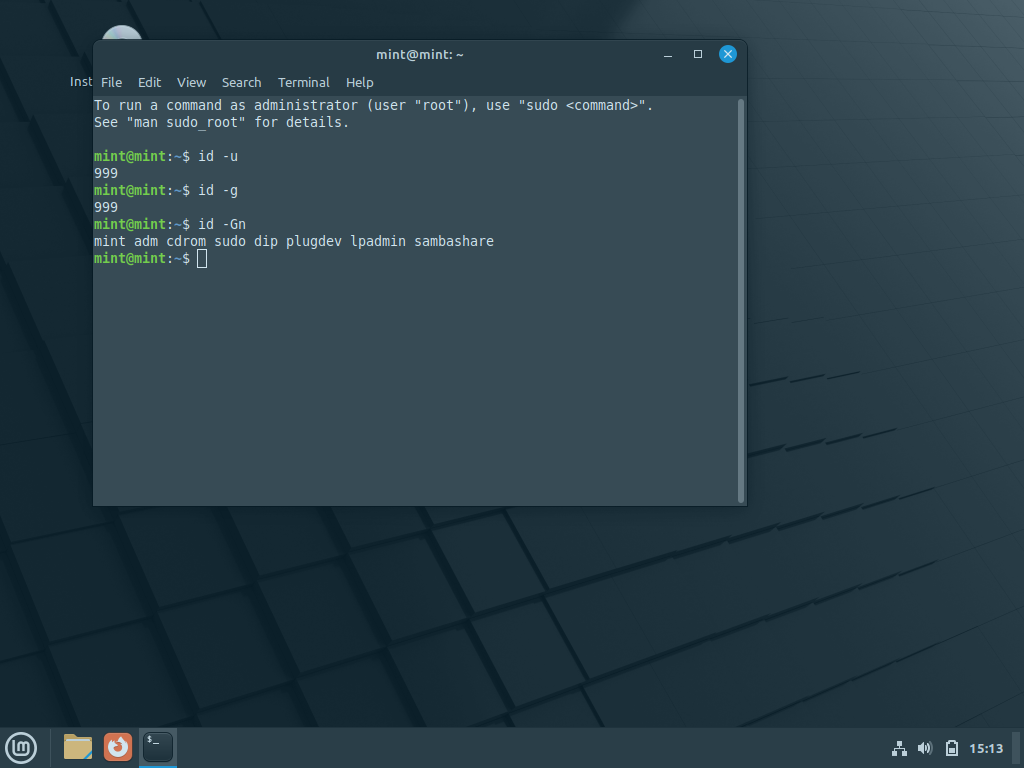
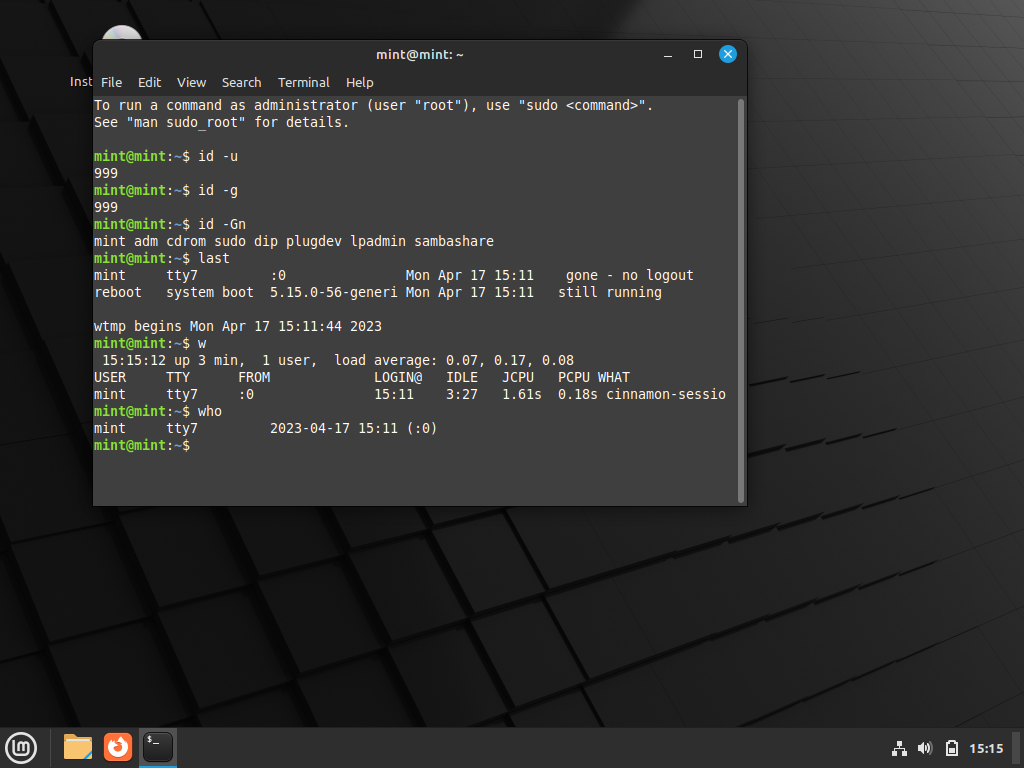
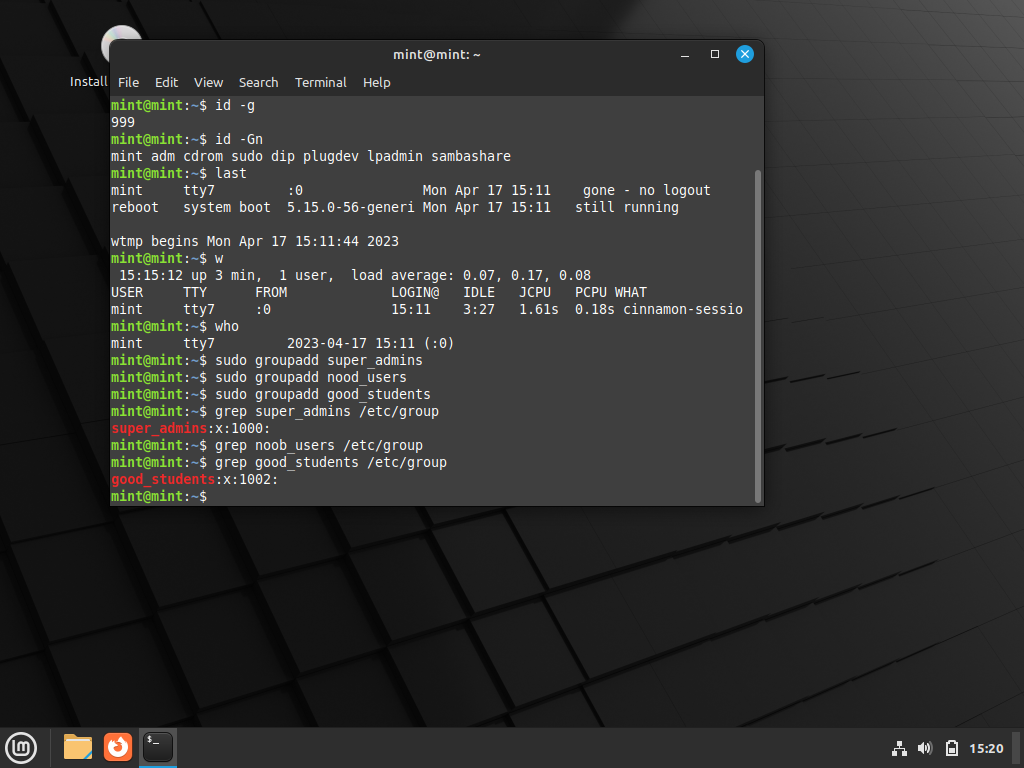
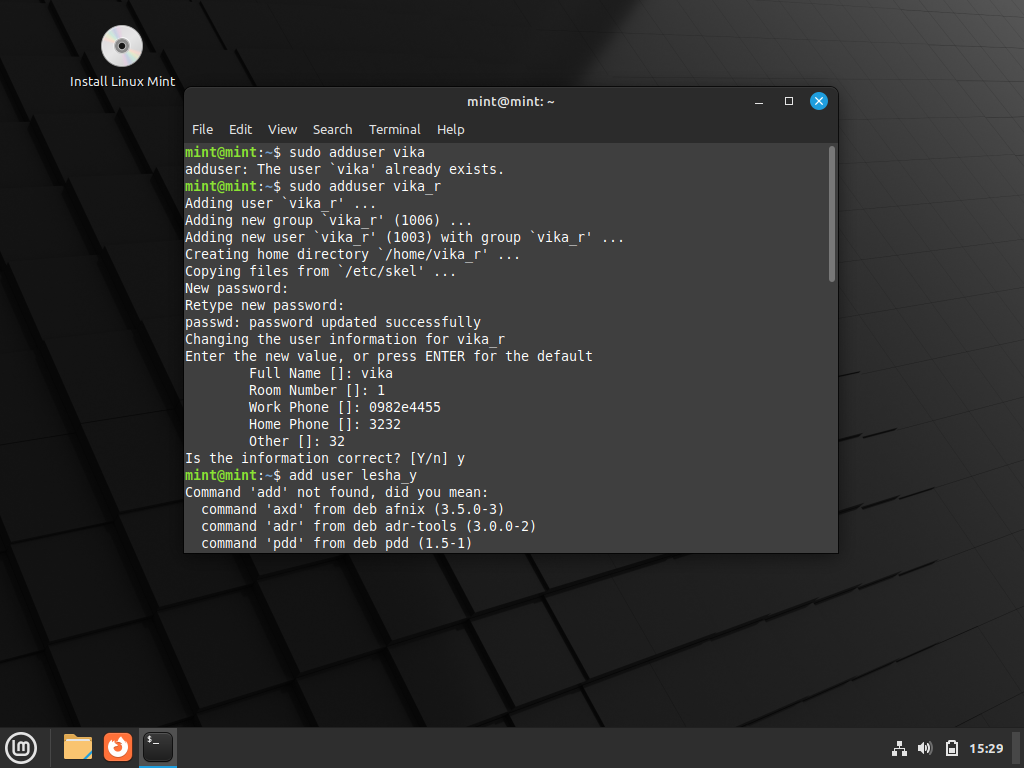
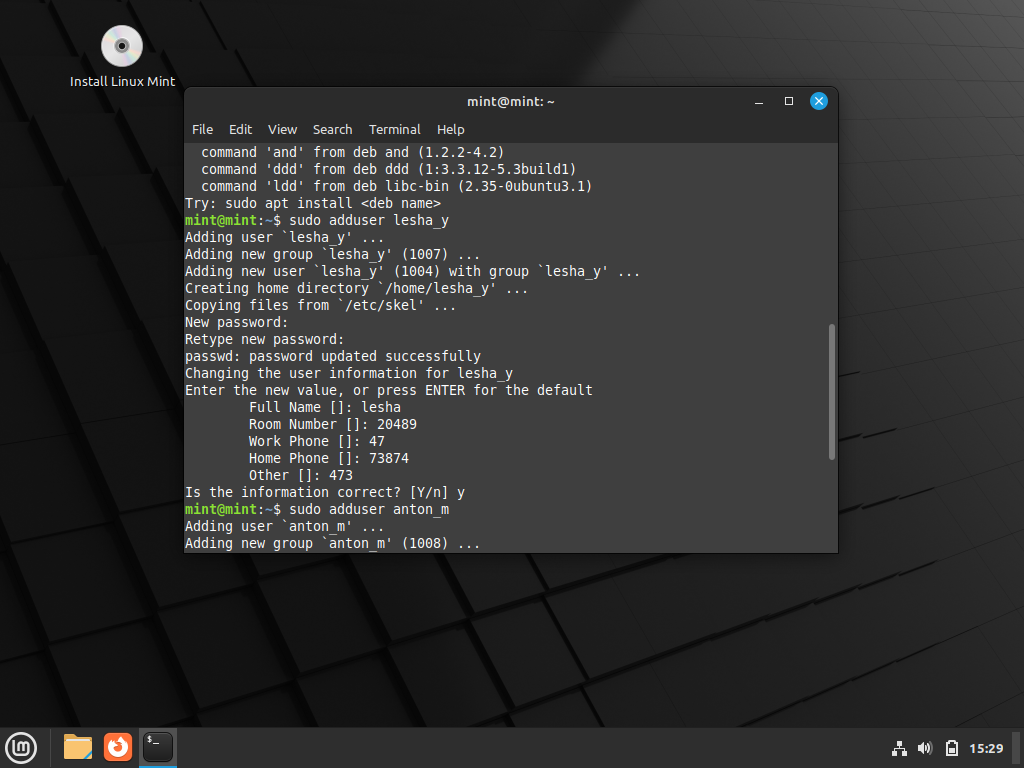
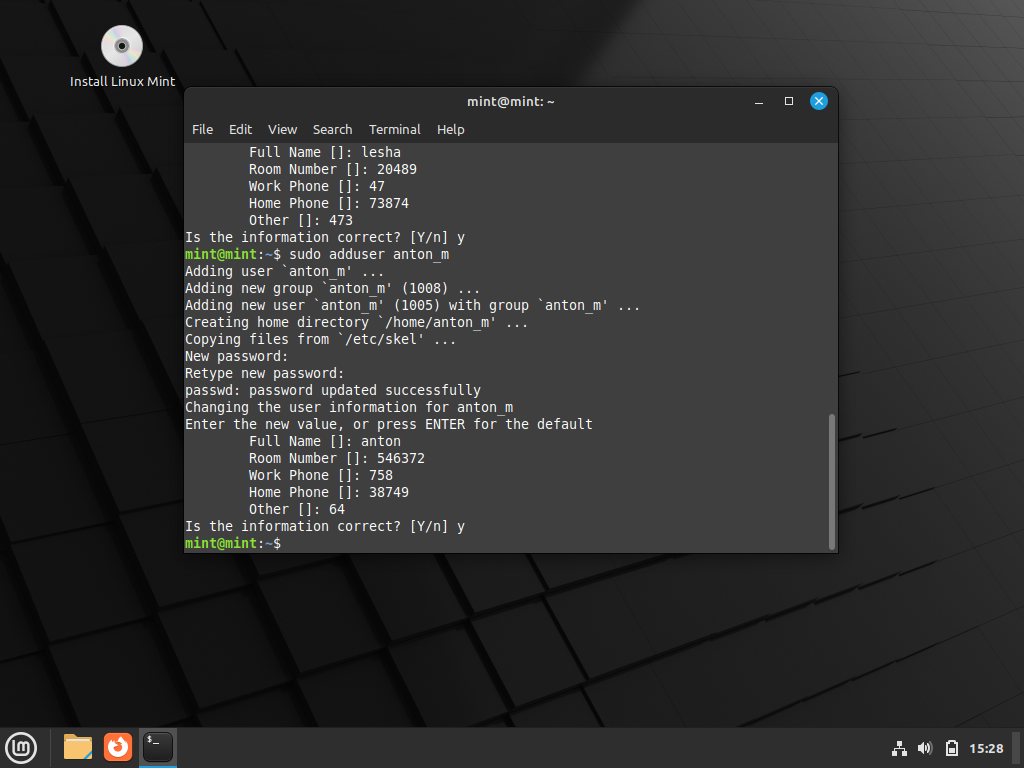
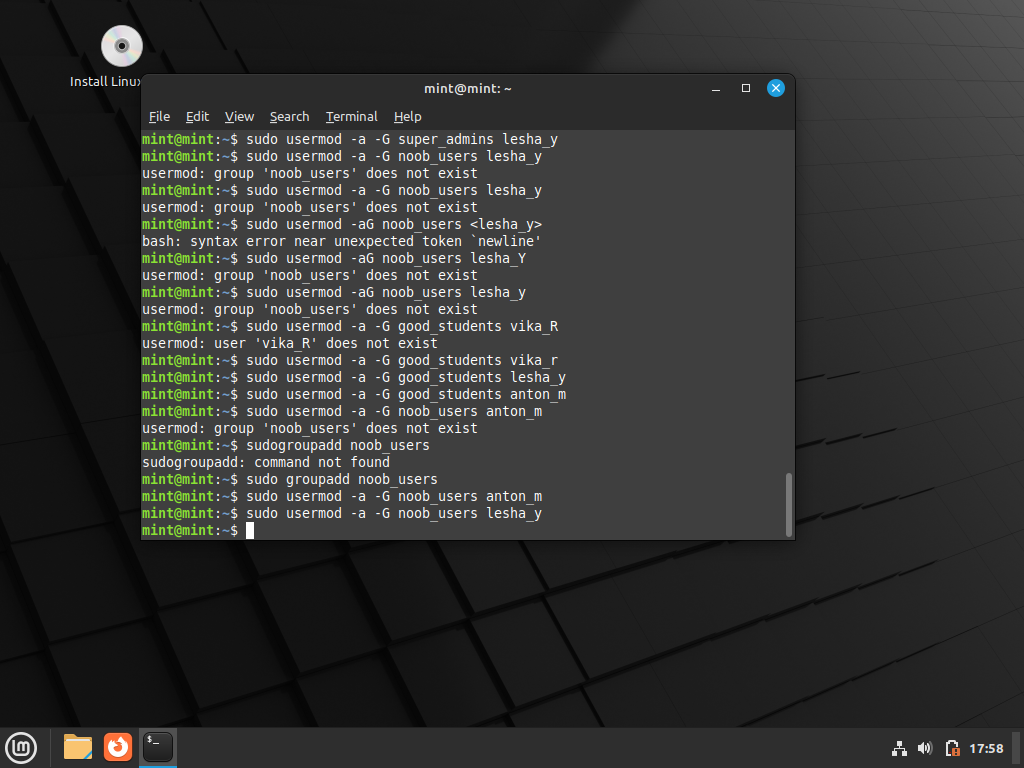
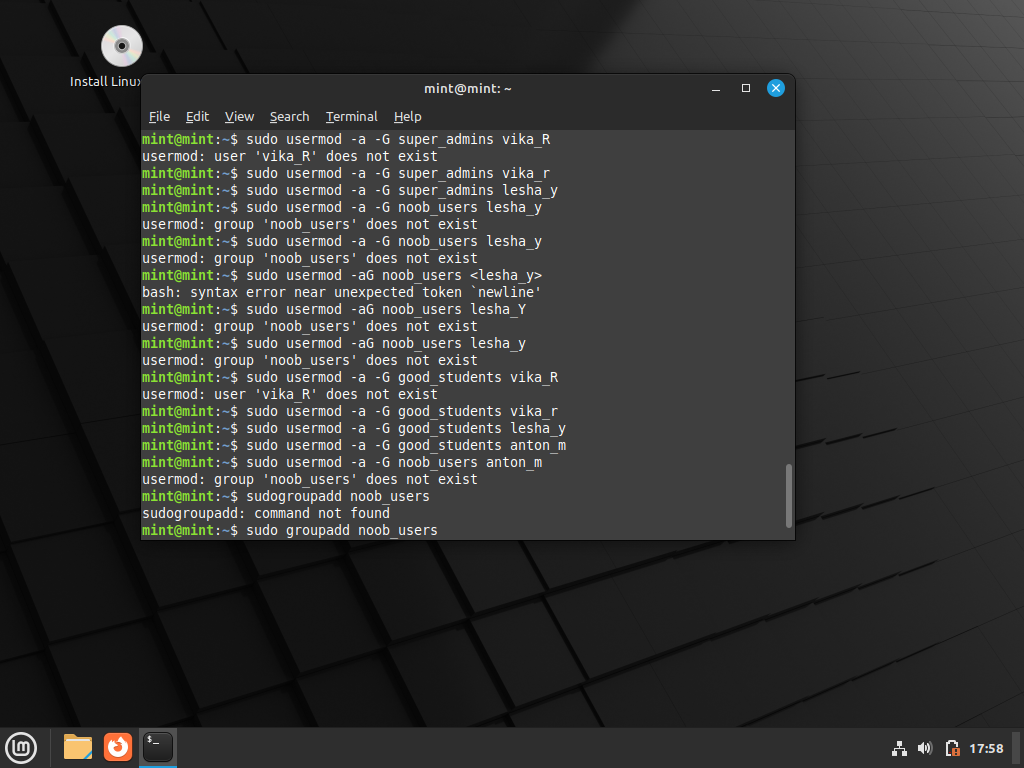
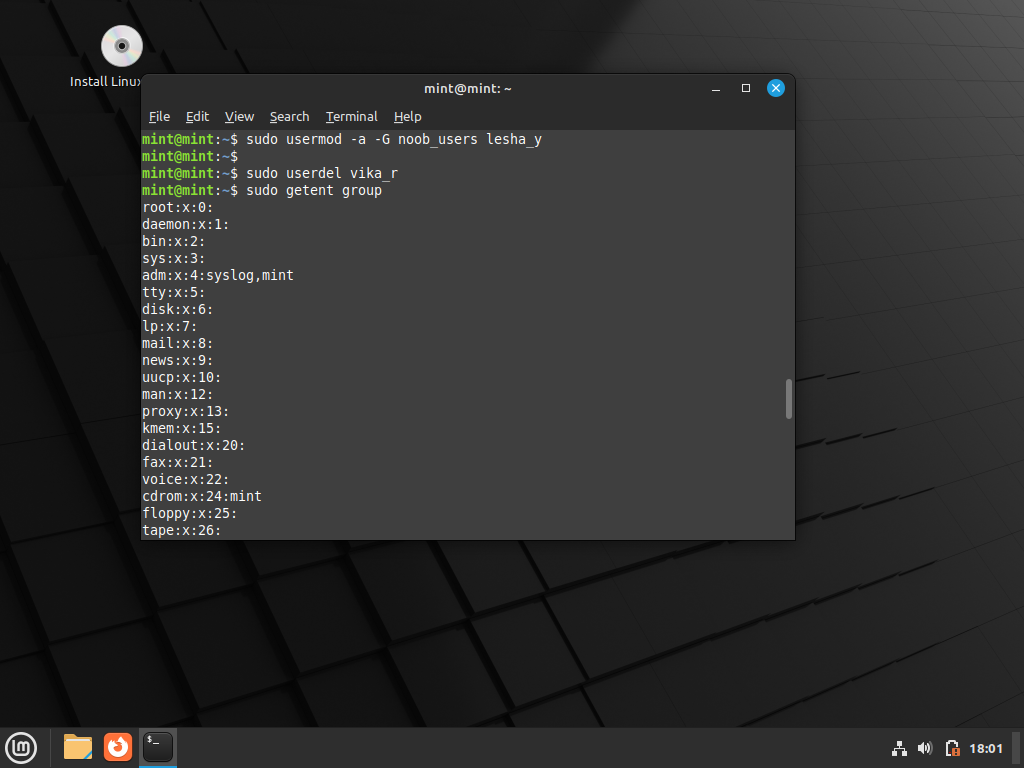
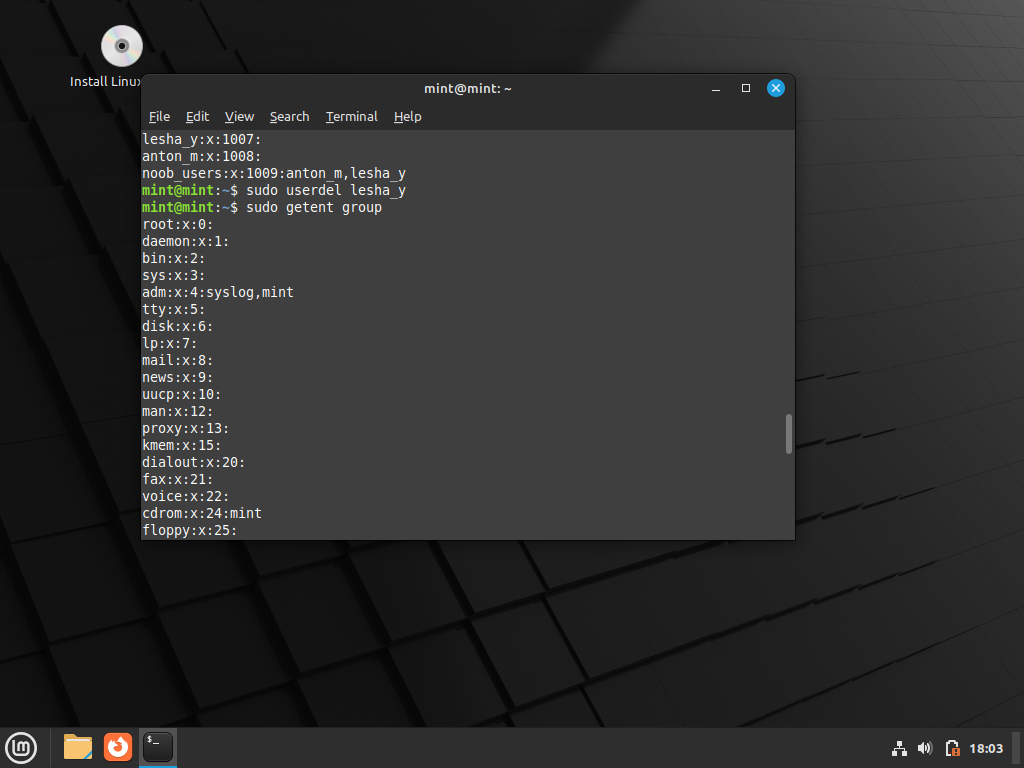
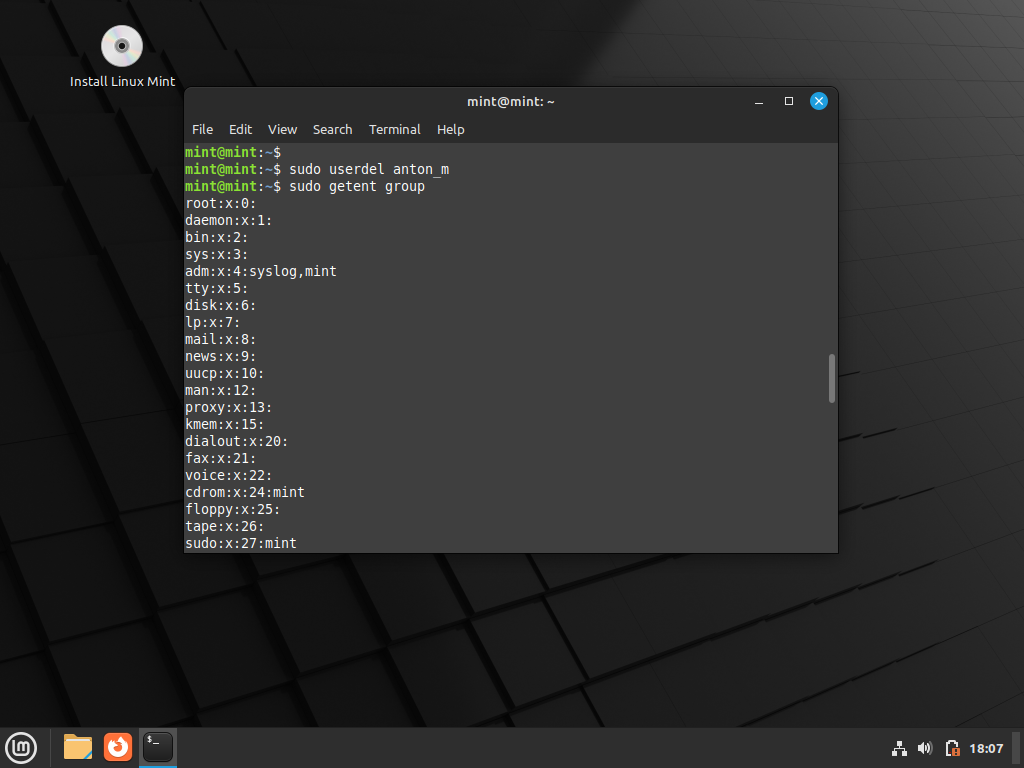
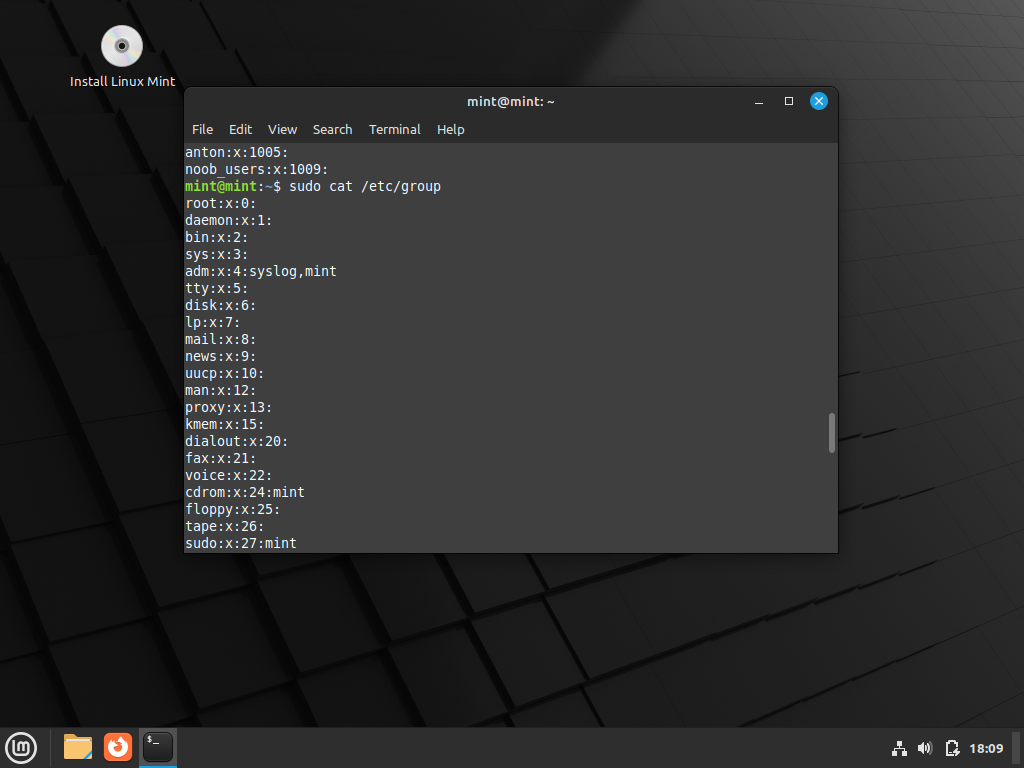
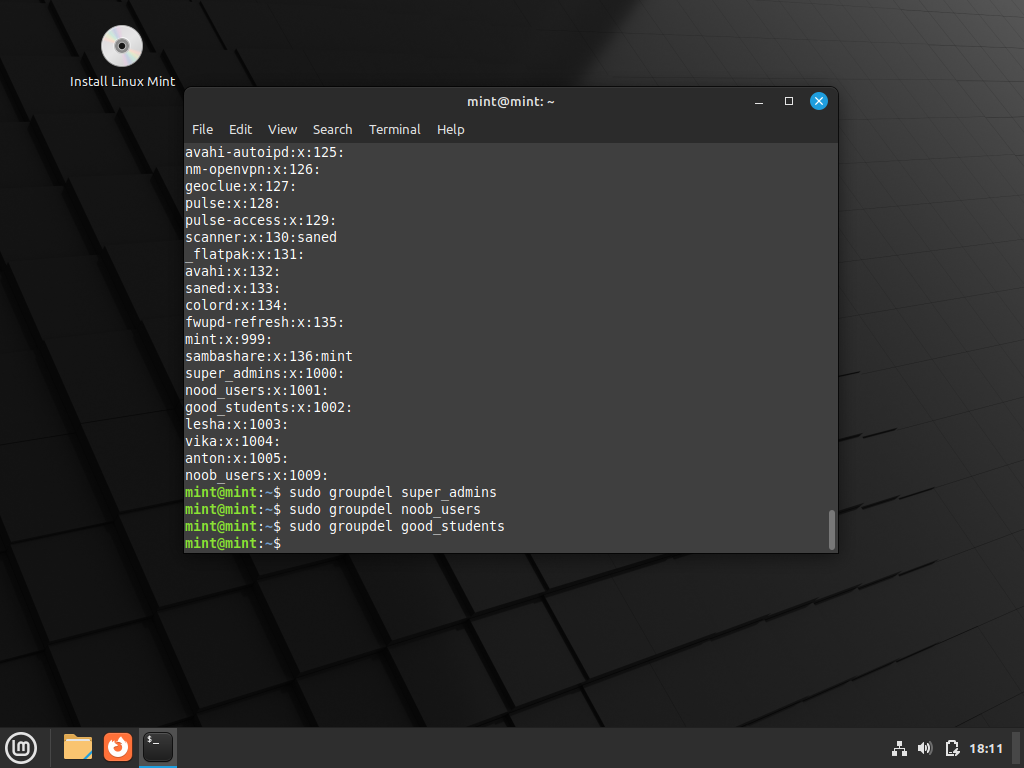
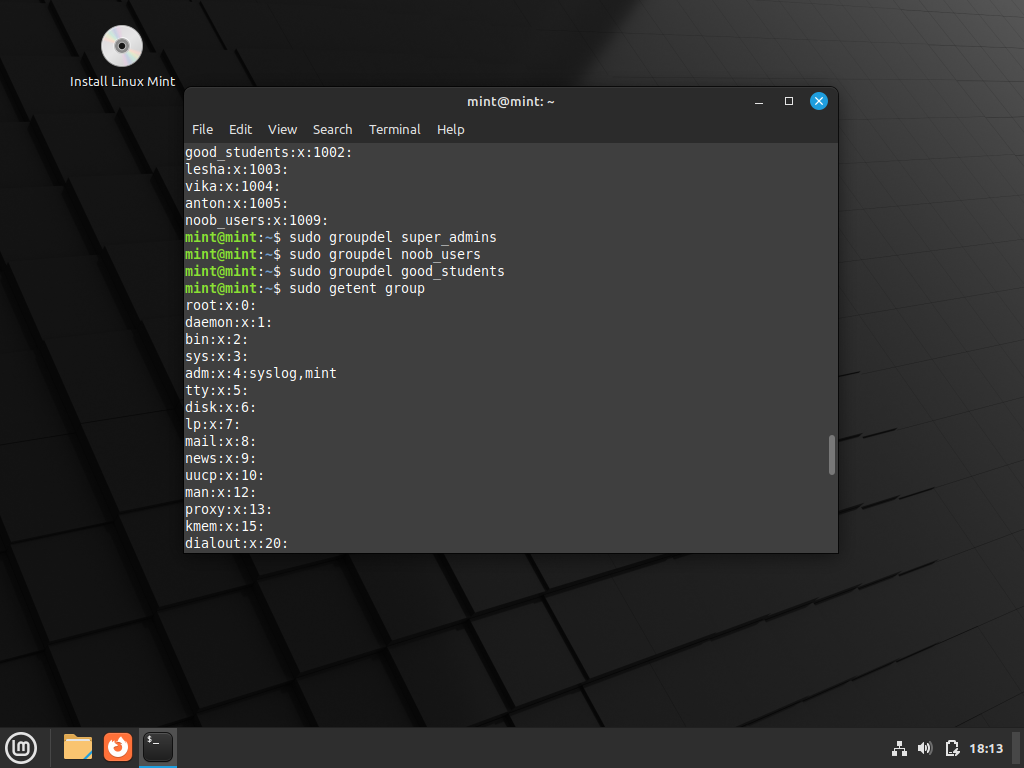
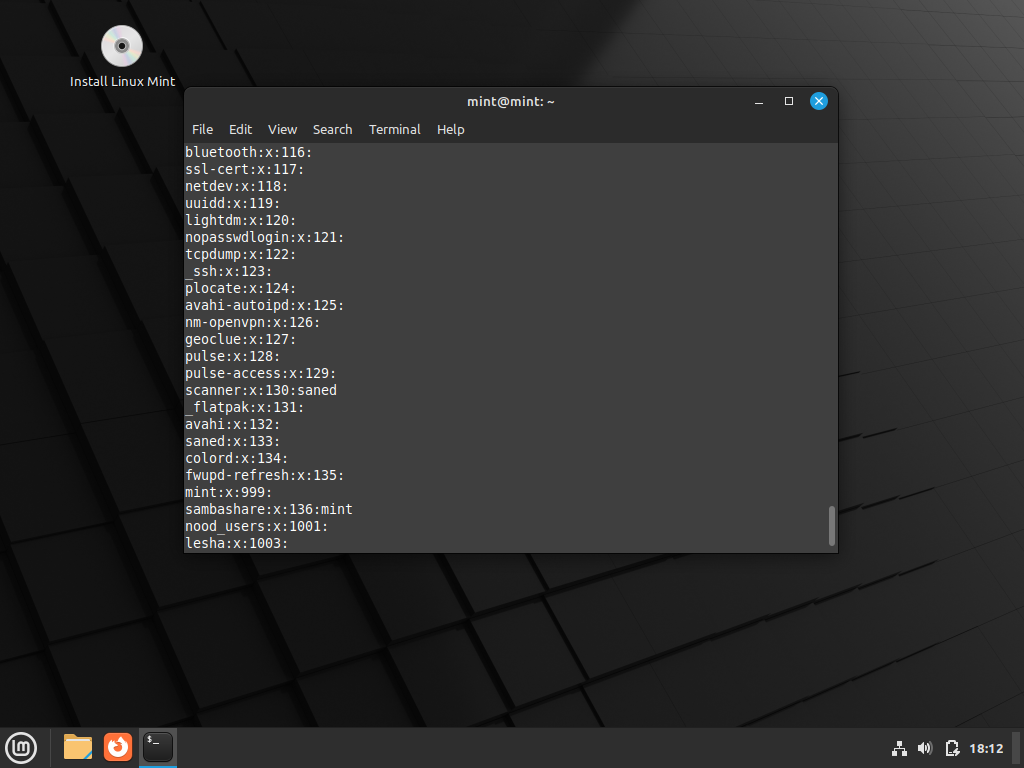
* Титульний аркуш, тема та мета роботи
* Словник термінів
* Відповіді на п.4.1 та п.4.5 з завдань для попередньої підготовки

**Хід роботи.**

* 1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
  2. Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її. Виконайте вхід в систему під користувачем: CentOS, пароль для входу: reverse ***(якщо виконуєте ЛР у 401 ауд.)*** та запустіть термінал.
  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 15: System and User Security*** та ***Lab 16: Creating Users and Groups.*** Створіть таблицю для опису цих команд\*\*\*

| Назва команди | Її призначення та функціональність |
| --- | --- |
| -l -ls | Право власності на файл можна підтвердити за допомогою параметра довгого списку -l команди ls. |
| chmod | якщо ви хочете зробити каталог більш приватним |
| stat | відображає більш детальну інформацію про файл, включаючи надання права власності на групу як за назвою групи, так і за номером GID. |
| chown | Команда може бути виконана лише користувачем root, і вона може змінити як користувача, так і групу, яка володіє файлом. |
| chgrp | Команда може використовуватися або користувачем, який є власником файлу, або користувачем root. |

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

* 1. Виконайте наступні практичні завдання у терміналі наступні дії (продемонструвати скріншоти):
* виведіть інформацію про поточного користувача різними способами (підказка використовуйте команди id та grep);
* 
* попрактикуйте в терміналі команди last, w та who. Порівняйте результати виводу кожної команди, які деталі відсутні в кожній із команд порівняно з іншими?
* 
* Звичайно! Ось порівняння команд **last**, **w** та **who** в терміналі, їх результати виводу та деталі, які можуть відсутні в кожній з команд:
* Команда **last**: Вона виводить інформацію про останні входження користувачів в систему, зчитуючи дані з файлу **/var/log/wtmp**. Команда **last** може відображати такі деталі:
* Ім'я користувача
* Термінал, з якого був здійснений вхід
* IP-адреса, з якої був здійснений вхід
* Дата та час входу та виходу
* Проте варто зазначити, що команда **last** може не відображати інформацію про поточних користувачів, оскільки вона зчитує дані з файлу журналу входжень **/var/log/wtmp**, який може бути очищений або обмежений правами доступу.
* Команда **w**: Вона виводить інформацію про поточних користувачів, які зараз залогінені в системі. Команда **w** може відображати такі деталі:
* Ім'я користувача
* Термінал, на якому користувач зараз залогінений
* Час входу користувача
* Завантаження системи та середнє навантаження
* Проте команда **w** може не відображати інформацію про минулі входження користувачів, а також не відображає IP-адреси та інші деталі, які можуть бути доступні в команді **last**.
* Команда **who**: Вона виводить коротку інформацію про поточних користувачів, які зараз залогінені в системі. Команда **who** може відображати такі деталі:
* Ім'я користувача
* Термінал, на якому користувач зараз залогінений
* Час входу користувача
* Проте команда **who** також може не відображати інформацію про минулі входження користувачів, IP-адреси
* створіть дві нові групи користувачів - super\_admins, noob\_users та good\_students, визначте їх ідентифікатори;
* 
* для кожного члену Вашої команди за допомогою терміналу створіть нового користувача (якщо працюєте самі, то просто трьох довільних користувачів), не забудьте після створення нового користувача одразу задати йому пароль;
* 
* 
* додайте нових користувачів у створені Вами нові групи таким чином, щоб у групах super\_admins та noob\_users було по 2 користувачі, один з яких є в обох групах, у групу good\_students додайте всіх трьох користувачів;
* 
* перегляньте інформацію про групи, та які користувачі до них входять, поясніть що ви бачите;
* Звідси видно, що групи "super\_admins" та "noob\_users" мають по 2 користувачі, а саме "user1" та "user2" входять до групи "super\_admins", а "user2" та "user3" входять до групи "noob\_users". Також зазначено, що користувач "user2" є спільним учасником обох груп "super\_admins" та "noob\_users".
* Група "good\_students" має всіх трьох користувачів, а саме "user1", "user2" та "user3". З цього можна зрозуміти, що всі троє користувачів входять до групи "good\_students".
* видаліть першого створеного вами користувача, перегляньте чи залишиться інформація про нього в групах, де він перебував;
* 
* видаліть другого користувача, перегляньте чи залишиться інформація про нього в групах, де він перебував;
* 
* видаліть третього користувача, перегляньте чи залишиться інформація про нього в групах, де він перебував;
* 
* перегляньте інформацію про існуючі групи користувачів;
* 
* видаліть створені Вами групи користувачів;
* 
* перегляньте інформацію про існуючі групи користувачів.
* 
* 

**Контрольні запитання**

1. Чому в конфігураційних файлах паролі не зберігається в явному вигляді?1

Захист від несанкціонованого доступу: Зберігання паролів в явному вигляді в конфігураційних файлах може бути небезпечним, оскільки це може дозволити несанкціонованій особі отримати доступ до системи та її ресурсів. Зашифровання паролів у конфігураційних файлах допомагає запобігти такому небажаному доступу.

Забезпечення цілісності: Зашифрування паролів у конфігураційних файлах також може бути застосовано для захисту від змін або випадкових помилок внаслідок людського фактору або автоматичних процесів. Це може допомогти забезпечити цілісність налаштувань системи.Також тому що відповідає стандартам безпеки

1. Чому не рекомендується виконувати повсякденні операції, використовуючи обліковий запис root?2
2. У чому відмінність механізмів отримання особливих привілеїв su і sudo?3
3. Чому домашній каталог користувача root не розміщено в каталозі /home?1

Відокремлення від звичайних користувачів: Домашній каталог користувача root розташований в окремому каталозі, щоб відокремити його від домашніх каталогів звичайних користувачів. Це забезпечує вищий рівень безпеки, оскільки користувач root має повний доступ до всієї системи, і може виконувати команди з підвищеними привілеями. Розміщення домашнього каталогу користувача root в окремому місці унеможливлює ненавмисний доступ користувачів звичайних рівнів до даних користувача root.

Сумісність зі стандартами: Розміщення домашнього каталогу користувача root в окремому каталозі /root відповідає стандартам систем Linux і Unix, таким як Filesystem Hierarchy Standard (FHS) і Linux Standard Base (LSB)

1. Для чого використовується команда getent?2
2. Яким чином можна видалити існуючі групи користувачів? Чи залишиться інформація про них десь у системі?3
3. Як можна змінити пароль користувача?1
4. Яке призначення команди chage?2
5. Які параметри команди usermod ви вважаєте найбільш використовуваними?3

**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!!!)**

**Віка - 1**

**Антон - 2**

**Льоша -3**

**Ми дізнались як додавати користувачів,як створювати групи,як видаляти користувачів,як видаляти групи,як виводити користувачів та групи на екран,переглядали яку інформацію містять групи,створювали нового користувача,зробили йому пароль і тд,** попрактикували в терміналі команди last, w та who. Порівняли результати виводу кожної команди,вивели інформацію про поточного користувача різними способами .