

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ ТА
ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Затверджую

Директор навчально-наукового
інституту інформаційних технологій

Меся Катерина НЕСТЕРЕНКО
«28» серпня 2025р.

МЕТОДИЧНІ ВКАЗІВКИ ДО КУРСОВОЇ РОБОТИ З ДИСЦИПЛІНИ
«СЕРВЕРНІ ОПЕРАЦІЙНІ СИСТЕМИ»

для студентів спеціальності G5 Електроніка, електронні комунікації,
приладобудування та радіотехніка

Завідувач кафедри комп'ютерних наук
д.т.н., професор Вишнівський В.В.

Київ 2025

Методичні вказівки до курсової роботи з дисципліни «Серверні операційні системи» призначені для розробки курсової роботи та самостійної підготовки здобувачів першого рівня (бакалаврського) вищої освіти спеціальності G5 Електроніка, електронні комунікації, приладобудування та радіотехніка.

Розробник: Вишнівський В.В., д.т.н., професор, завідувач кафедри Комп'ютерних наук.

Методичні вказівки до курсової роботи схвалені на засіданні кафедри Комп'ютерних наук, протокол № 1 від «28» серпня 2025р.

Завідувач кафедри Комп'ютерних наук
 Віктор ВИШНІВСЬКИЙ

1. Вступ

Курсова робота з дисципліни «Серверні операційні системи» є однією з форм самостійної роботи студента, спрямованою на закріплення теоретичних знань та формування практичних навичок адміністрування серверних операційних систем.

У процесі виконання курсової роботи студент застосовує знання, отримані під час лекційного і лабораторного курсів, для розв'язання практичних задач адміністрування серверу. Робота передбачає розгортання віртуального сервера, встановлення та налаштування конкретного серверного компонента, документування процесу і подання результатів у формі пояснювальної записки.

Актуальність тематики курсової роботи зумовлена стрімким зростанням потреби у кваліфікованих фахівцях з адміністрування серверних систем. Знання, отримані в процесі виконання курсової, є базовими для подальшого професійного розвитку в напрямках системного адміністрування, DevOps, кібербезпеки та хмарних технологій.

2. Технології для застосування

Для виконання курсової роботи студент застосовує наступний перелік технологій та інструментів. Конкретний набір залежить від обраного варіанту.

2.1. Операційні системи

Залежно від варіанту, курсова робота виконується на одній із серверних операційних систем:

- Ubuntu Server версії 22.04 LTS або 24.04 LTS – рекомендовано для більшості Linux-варіантів;
- Debian версії 12 (Bookworm) – альтернатива Ubuntu;
- Rocky Linux або AlmaLinux версії 9 – для варіантів, де використовується екосистема RHEL;
- Windows Server 2022 або 2025 у редакції Standard – для варіантів на платформі Windows.

2.2. Гіпервізори

Робота виконується у віртуальному середовищі. Рекомендовані гіпервізори:

- Oracle VirtualBox – безкоштовний, кросплатформний, найпростіший для початківців;
- VMware Workstation Player – безкоштовний для персонального використання;
- Microsoft Hyper-V – вбудований у Windows Pro та Enterprise;
- KVM – штатний гіпервізор Linux, для більш досвідчених користувачів.

Рекомендовані параметри віртуальної машини для більшості варіантів: 2 vCPU, 2-4 ГБ RAM, диск 20-40 ГБ.

2.3. Засоби віддаленого доступу

Для роботи з серверною ОС через командний рядок:

- SSH-клієнти: OpenSSH (вбудований у Linux/macOS/Windows 10+), PuTTY, MobaXterm, Windows Terminal;
- RDP-клієнти: вбудований у Windows, Remmina (Linux), Microsoft Remote Desktop (macOS).

2.4. Текстові редактори

Для роботи з конфігураційними файлами:

- nano – найпростіший, для початківців;
- vim – класичний, потужний, потребує звикання;
- Visual Studio Code – з розширенням Remote-SSH, для комфортного редагування конфігів на віддаленому сервері.

2.5. Мови скриптування

Мови скриптування вибираються залежно від платформи:

- Bash – основна командна оболонка Linux, для скриптів адміністрування;
- PowerShell – основна оболонка Windows Server, повноцінна об'єктна мова.

Для оформлення пояснювальної записки:

- Microsoft Word – стандартний інструмент для оформлення записки;

- засоби створення скріншотів: вбудований у Windows (Win+Shift+S), Snipping Tool, Greenshot;
- засоби створення діаграм: Draw.io, Microsoft Visio, PlantUML.

4. Опис курсової роботи у документації (опис роботи)

Курсова робота полягає у виконанні комплексу дій із налаштування конкретного серверного компонента, детальному документуванні процесу і поданні результатів у формі пояснювальної записки з додатками.

4.1. Загальний порядок виконання

1. Обрати варіант курсової роботи з переліку або отримати варіант від викладача.
2. Ознайомитися з теоретичним матеріалом за темою обраного варіанту (лекції курсу, рекомендована література, офіційна документація).
3. Підготувати робоче середовище: встановити гіпервізор, завантажити ISO-образ обраної ОС.
4. Розгорнути віртуальну машину, встановити операційну систему з мінімальною конфігурацією. Або використати вже готову.
5. Виконати практичні завдання відповідно до варіанту, роблячи скріншоти кожного кроку.
6. Зберегти всі конфігураційні файли і скрипти, які були створені або змінені.
7. Оформити пояснювальну записку згідно з вимогами.
8. Здати роботу викладачу у визначений термін і захистити її.

4.2. Робота у віртуальному середовищі

Використання віртуальних машин є ключовою вимогою для виконання курсової роботи. Це обумовлено наступними причинами:

- безпека: можливі помилки не завдають шкоди основній системі студента;
- повторюваність: можливість почати заново у разі критичної помилки;

- знімки стану (snapshots): можна зберігати стан ВМ на різних етапах виконання;

- реалістичність: віртуальна машина максимально наближена до реального сервера.

Рекомендується робити знімок стану ВМ після кожного значного етапу – це дозволить швидко повернутися до попереднього стану у разі помилки.

4.3. Скріншоти і документування процесу

Кожен значний крок виконання роботи повинен бути задокументований скріншотом. Скріншот повинен:

- мати достатню роздільну здатність для читання тексту (мінімум 800x600);
- містити тільки релевантну інформацію – обрізати зайве;
- мати підпис із номером рисунка і коротким описом того, що зображено.

4.4. Загальні вимоги до оформлення

Основні технічні вимоги до оформлення пояснювальної записки:

- шрифт основного тексту: Times New Roman, 14 пт;
- міжрядковий інтервал: 1,5;
- вирівнювання: за шириною;
- абзацний відступ: 1,25 см;
- поля: ліве – 30 мм, праве – 15 мм, верхнє і нижнє – по 20 мм;
- нумерація сторінок: у верхньому правому куті, починаючи з розділу «Вступ»;
- рисунки та таблиці нумеруються послідовно, обов'язково мають назву;
- кодові фрагменти оформлюються моноширним шрифтом (Courier New або Consolas, 12 пт).

5. Варіанти курсової роботи

Нижче наведено перелік варіантів курсової роботи.

Варіант 1. Створення та налаштування користувачів у Linux

Мета: Опанувати базові операції адміністрування облікових записів у серверній ОС на прикладі Ubuntu Server.

Завдання:

1. Створити 5 користувачів з різними наборами прав (адміністратор, звичайні користувачі, сервісний акаунт).
2. Створити 2 групи (наприклад, `developers` та `operators`) і додати до них відповідних користувачів.
3. Налаштувати доступ через `sudo` для одного з користувачів.
4. Створити сервісний акаунт без можливості входу в систему (`nologin`).
5. Задokumentувати вміст файлів `/etc/passwd`, `/etc/group`, `/etc/shadow` (без реальних хешів).

Вимоги до звіту:

- скріншоти первинного списку користувачів;
- скріншоти фінішного списку користувачів;
- скріншоти групи `developers` та `operators`.

Варіант 2. Установка і базове налаштування веб-сервера Nginx

Мета: Опанувати процес встановлення веб-сервера та первинного налаштування віртуального хоста.

Завдання:

1. Встановити Nginx на Ubuntu Server через пакетний менеджер `apt`.
2. Перевірити роботу веб-сервера через браузер (сторінка `Welcome`).
3. Створити віртуальний хост для сайту з довільним іменем (наприклад, `my-site.local`).
4. Розмістити просту HTML-сторінку з інформацією про сервер.
5. Налаштувати логування (`access.log`, `error.log`).

Вимоги до звіту:

- скріншоти процесу встановлення Nginx;
- повний текст файлу конфігурації віртуального хоста з поясненнями;
- скріншот відкритої сторінки в браузері;
- приклад вмісту access.log після відвідування сайту.

Варіант 3. Установка і базове налаштування веб-сервера Apache

Мета: Опанувати процес встановлення веб-сервера Apache та створення VirtualHost.

Завдання:

1. Встановити Apache HTTP Server на Ubuntu Server.
2. Активувати необхідні модулі (rewrite, headers).
3. Створити VirtualHost для сайту з довільним іменем.
4. Розмістити просту HTML-сторінку та перевірити її відкриття.
5. Налаштувати базову перенаправлення .htaccess.

Вимоги до звіту:

- скріншоти встановлення Apache;
- повний текст файлу конфігурації VirtualHost;
- скріншот команди a2ensite та її результату;
- скріншот відкритої сторінки в браузері.

Варіант 4. Налаштування SSH-сервера з ключовою автентифікацією

Мета: Налаштувати безпечний віддалений доступ до серверу з використанням SSH-ключів замість паролів.

Завдання:

1. Встановити OpenSSH-server на Ubuntu Server.
2. Згенерувати пару SSH-ключів на клієнтській машині.
3. Скопіювати публічний ключ на сервер через ssh-copy-id.

4. Змінити порт SSH з 22 на нестандартний (наприклад, 2222).
5. Вимкнути можливість входу за паролем у конфігурації `sshd_config`.
6. Заборонити прямий вхід від імені `root`.

Вимоги до звіту:

- скріншоти процесу генерації ключів;
- фрагмент файлу `sshd_config` з внесеними змінами;
- скріншот успішного входу через SSH-ключ;
- пояснення переваг ключової автентифікації над паролями.

Варіант 5. Налаштування файрволу UFW у Linux

Мета: Опанувати базове налаштування файрволу для захисту сервера через мережу.

Завдання:

1. Встановити (за потреби) та увімкнути UFW на Ubuntu Server.
2. Дозволити вхідні з'єднання за SSH (порт 22) до увімкнення файрволу.
3. Налаштувати правила для HTTP (80) та HTTPS (443).
4. Заборонити з'єднання за одним із портів (наприклад, Telnet 23).
5. Дозволити з'єднання за SSH тільки з певної підмережі.
6. Перевірити правила через `ufw status verbose`.

Вимоги до звіту:

- скріншот кожної команди зі створення правил;
- результат `ufw status` до і після налаштування;
- пояснення чому SSH треба дозволити ПЕРЕД увімкненням UFW;
- тест з іншої машини для перевірки заборонених з'єднань.

Варіант 6. Планування задач через cron

Мета: Опанувати механізм планування регулярних задач у Linux.

Завдання:

1. Ознайомитися з синтаксисом cron-виразів.
2. Створити задачу, яка щохвилини записує дату та час у файл.
3. Створити задачу для щоденного архівування вмісту директорії /var/log.
4. Створити задачу, що щотижня видаляє старі архіви (старші 30 днів).
5. Налаштувати одноразову задачу через at.

Вимоги до звіту:

- скріншоти команди crontab -e з поясненням синтаксису;
- повний текст bash-скриптів, що викликаються з cron;
- скріншот вмісту логів після спрацювання задач;
- таблиця з поясненням cron-синтаксису.

Варіант 7. Установка і базове налаштування MySQL

Мета: Опанувати процес встановлення СКБД та базові операції з базами даних.

Завдання:

1. Встановити MySQL Server на Ubuntu Server.
2. Виконати початкове налаштування безпеки через mysql_secure_installation.
3. Створити базу даних з довільною назвою.
4. Створити користувача бази даних з обмеженими правами (тільки на створену БД).
5. Створити таблицю з кількома полями і додати тестові записи.
6. Виконати кілька SELECT-запитів.

Вимоги до звіту:

- скріншоти процесу встановлення та початкового налаштування
- команди створення БД, користувача, таблиці – з поясненнями
- скріншот результату SELECT-запитів
- пояснення різниці між правами root і створеного користувача

Варіант 8. Робота з файловою системою та розділами диска

Мета: Опанувати операції з розмітки, форматування та монтування дисків.

Завдання:

1. Додати до віртуальної машини другий віртуальний диск (наприклад, 5 ГБ).
2. Створити на новому диску розділ через `fdisk` або `parted`.
3. Відформатувати розділ у `ext4`.
4. Створити точку монтування (наприклад, `/mnt/data`).
5. Виконати тимчасове монтування через `mount`.
6. Налаштувати автоматмонтування через `/etc/fstab`.
7. Перевірити коректність налаштування перезавантаженням системи.

Вимоги до звіту:

- скріншоти команд `lsblk`, `fdisk/parted`;
- пояснення обраної файлової системи;
- фрагмент `/etc/fstab` з внесеним рядком;
- скріншот команди `df -h` після монтування.

Варіант 9. Написання bash-скрипта для адміністрування

Мета: Опанувати основи написання скриптів для автоматизації рутинних задач.

Завдання:

1. Написати скрипт, який приймає параметр (наприклад, ім'я директорії).
2. Реалізувати перевірку наявності параметра і виведення інструкції.
3. Реалізувати перевірку існування директорії.
4. Створити `tar.gz`-архів вказаної директорії з іменем, що містить поточну дату.
5. Записати результат у лог-файл.
6. Зробити скрипт виконуваним і протестувати з різними параметрами.

Вимоги до звіту:

- повний текст скрипта з коментарями;
- пояснення кожної команди;
- скріншоти тестування скрипта з різними вхідними даними;

- приклад вмісту створеного лог-файлу.

Варіант 10. Установка та базове використання Docker

Мета: Опанувати початкові навички роботи з контейнерами.

Завдання:

1. Встановити Docker Engine на Ubuntu Server.
2. Запустити перший контейнер (наприклад, hello-world).
3. Запустити контейнер з веб-сервером Nginx з прокидання порту.
4. Ознайомитися з командами `docker ps`, `docker logs`, `docker exec`.
5. Створити власний Dockerfile для простого веб-сайту.
6. Побудувати образ і запустити контейнер із власного образу.

Вимоги до звіту:

- скріншоти встановлення Docker;
- приклад запусчених контейнерів (`docker ps`);
- текст створеного Dockerfile з поясненнями;
- скріншот відкритої в браузері сторінки з контейнеру.

Варіант 11. Створення користувачів у Windows Server через PowerShell

Мета: Опанувати керування локальними обліковими записами у Windows Server через командний рядок.

Завдання:

1. Встановити Windows Server у режимі Server Core або з графічним інтерфейсом.
2. Через PowerShell створити 5 локальних користувачів з різними атрибутами.
3. Створити локальну групу і додати до неї частину користувачів.
4. Налаштувати політику складності паролів.
5. Написати PowerShell-скрипт для масового створення користувачів зі списку у CSV-файлі.

Вимоги до звіту:

- скріншоти кожної команди PowerShell;
- повний текст скрипта масового створення;
- приклад CSV-файлу з даними користувачів;
- скріншот вмісту `lusrmgr.msc` після виконання скрипта.

Варіант 12. Установка і базове налаштування IIS у Windows Server

Мета: Опанувати процес встановлення веб-сервера IIS та розгортання простого сайту.

Завдання:

1. Встановити Windows Server (2022 або 2025).
2. Через Server Manager (або PowerShell) встановити роль Web Server (IIS).
3. Перевірити роботу IIS через браузер (сторінка за замовчуванням).
4. Створити новий сайт з довільним іменем через IIS Manager.
5. Розмістити просту HTML-сторінку в директорії сайту.
6. Налаштувати сайт на прослуховування порту 8080.

Вимоги до звіту:

- скріншоти процесу встановлення ролі IIS;
- скріншоти IIS Manager з налаштуваннями сайту;
- приклад PowerShell-команди для встановлення ролі;
- скріншот сайту, відкритого в браузері.

6. Порядок захисту роботи

Для захисту курсової роботи, її опис разом із скриншотами, має бути заздалегідь (за 3 дні до захисту) наданий викладачу дисципліни «СЕРВЕРНІ ОПЕРАЦІЙНІ СИСТЕМИ».

Метою курсової роботи є закріплення теоретичних знань з дисципліни «Серверні операційні системи» шляхом практичного розгортання та налаштування конкретного серверного компонента на віртуальному середовищі, з подальшим документуванням процесу і результатів.

Список рекомендованих джерел

Загальні джерела, які корисні для більшості варіантів. Для конкретного варіанту студент самостійно доповнює список специфічними ресурсами.

1. Ubuntu Server Documentation - <https://documentation.ubuntu.com/server/> .
2. The Linux Command Line, William Shotts -
<https://www.kea.nu/files/textbooks/humblesec/thelinuxcommandline.pdf>
3. Microsoft Learn – Windows Server - <https://learn.microsoft.com/en-us/windows-server/> .
4. PowerShell Documentation – <https://learn.microsoft.com/en-us/powershell/> .
5. Arch Linux Wiki – <https://wiki.archlinux.org/> (корисно для всіх варіацій ОС Linux).
6. Nginx Documentation – <https://nginx.org/en/docs/> .
7. Apache HTTP Server Documentation – <https://httpd.apache.org/docs/> .
8. Docker Documentation – <https://docs.docker.com/>.
9. MySQL Reference Manual – <https://dev.mysql.com/doc/> .
10. DigitalOcean Community Tutorials –
<https://www.digitalocean.com/community/tutorials> .

ДОДАТКИ

Додаток А

Приклад оформлення титульного листа курсової роботи

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Курсова робота

з дисципліни "Серверні операційні системи"

на тему: «Створення та налаштування користувачів у Linux»

Виконав:
здобувач вищої освіти
група ІТД-31

Максим КОРОЛЮК

Перевірив:
науковий ступінь, вчене звання

Сергій ВАСИЛЕНКО
к.т.н., доцент

Київ 2026

Приклад Bash-скрипта

```
#!/bin/bash
# create_user.sh – скрипт створення нового користувача
# Використання: ./create_user.sh <ім'я_користувача>

# Перевірка наявності параметра
if [ -z "$1" ]; then
    echo "Помилка: не вказано ім'я користувача"
    echo "Використання: $0 <ім'я_користувача>"
    exit 1
fi

USERNAME=$1

# Перевірка прав root
if [ "$EUID" -ne 0 ]; then
    echo "Помилка: скрипт треба запускати з правами root"
    exit 1
fi

# Створення користувача
useradd -m -s /bin/bash "$USERNAME"
```

Приклад конфігураційного файлу

```
server {  
    listen 80;  
    listen [::]:80;  
  
    server_name example.local;  
    root /var/www/example;  
    index index.html index.htm;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
  
    error_log /var/log/nginx/example_error.log;  
    access_log /var/log/nginx/example_access.log;  
}
```

Оформлення виводу команд

```
$ systemctl status nginx
```

```
● nginx.service - A high performance web server
```

```
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled)
```

```
   Active: active (running) since Wed 2026-09-23 10:15:32 EEST
```

```
     Docs: man:nginx(8)
```

```
   Process: 1234 ExecStartPre=/usr/sbin/nginx -t (code=exited)
```

```
 Main PID: 1235 (nginx)
```

```
    Tasks: 3 (limit: 4644)
```

```
   Memory: 4.2M
```