

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра комп'ютерної інженерії та кібербезпеки

ЗВІТ З ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ

студентки IV курсу групи КІ-19-1
спеціальності 123 «Комп'ютерна
інженерія»

Матвієнко Мар'яни Валеріївни
(прізвище, ім'я та по-батькові)

Керівник старший викладач кафедри КІ
та КБ, Окунькова О. О.

Дата захисту: " 27 " березня 2023 р.

Національна шкала _____

Кількість балів: _____

Оцінка: ECTS _____

Члени комісії

(підпис)

О. О. Окунькова
(прізвище та ініціали)

(підпис)

В. В. Воротніков
(прізвище та ініціали)

(підпис)

Ю.М. Росінський
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до переддипломної практики складається зі вступу, трьох розділів, висновку, списку використаної літератури.

Текстова частина викладена на 33 сторінках друкованого тексту.

Пояснювальна записка має список використаних джерел, який містить 6 найменувань і займає 1 сторінку. В роботі наведено 25 рисунків. Загальний обсяг роботи – 39 сторінок.

У першому розділі визначено загальний опис компанії та проведено статистику використання різних ОС серед розробників та системних адміністраторів.

У другому розділі виконано описано підсистеми адміністрування Linux та навчено роботі з ними.

У третьому розділі описано основні модулі для роботи в мові Python для адміністрування Linux та наведено скрипти для роботи із файловою системою.

Висновки містять в собі результати виконаної роботи.

Ключові слова: ПІДСИСТЕМА АДМІНІСТРУВАННЯ, ОС LINUX, PYTHON, АВТОМАТИЗАЦІЯ ПРОЦЕСІВ, СКРИПТИ, ВІДДАЛЕНИЙ ДОСТУП ДО СИСТЕМИ.

					«Житомирська політехніка».23.123.11.000 – ПП				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Матвієнко М.В.			Звіт з переддипломної практики		Літ.	Арк.	Аркушів
Керівник		Ожунькова О.О.						4	
Реценз.							ФІКТ, гр. КІ-19-1		
Н. Контр.									
Затверд.									

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	4
ВСТУП	5
РОЗДІЛ 1. ВІДОМОСТІ ПРО МІСЦЕ ПРОХОДЖЕННЯ ПРАКТИКИ.....	6
1.1 Компанія-розробник ПЗ ТОВ «ТЕУМАХ».....	6
1.2 Використання різних ОС серед розробників та адміністраторів	7
Висновки до першого розділу	9
РОЗДІЛ 2. ОГЛЯД ПІДСИСТЕМ АДМІНІСТРУВАННЯ ОС LINUX.....	10
2.1 Підсистеми адміністрування ОС Linux.....	10
2.1.1 Systemd	11
2.1.2 Puppet.....	15
2.2 Визначення задач для автоматизації	21
Висновки до другого розділу	22
ВИСНОВКИ.....	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	33

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ТОВ	Товариство обмеженої відповідальності
ОС	Операційна система
УІАП	Управління розвитку інформаційних технологій
DHCP	Dynamic Host Configuration Protocol

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ВСТУП

Актуальність теми. Адміністрування Linux є важливою складовою роботи будь-якої компанії, яка використовує Linux у своїй інфраструктурі. Це може бути звичайний веб-сервер, хмарний сервіс, мережа комп'ютерів, кластери серверів тощо. Однак, для забезпечення ефективної роботи такої інфраструктури, потрібна знання інструментів адміністрування та автоматизації процесів.

Мова програмування Python займає лідерські позиції в розробці програмного забезпечення та адмініструванні систем. Python дозволяє ефективно і швидко розробляти скрипти для автоматизації рутинних операцій, керування ресурсами та обробки даних.

Отже, тема проєкту підсистеми адміністрування ОС Linux на основі мови програмування Python є дуже актуальною та важливою в сучасному світі, оскільки дозволяє підвищити ефективність роботи компаній, які використовують Linux в своїй інфраструктурі, та забезпечити безпеку і захист даних.

Мета проходження переддипломної практики – набуття практичних навичок розробки та адміністрування ОС Linux на основі Python.

Предмет переддипломної практики – інформаційні технології, пов'язані з розробкою програмного забезпечення та адмініструванням операційних систем Linux.

Основними завданнями переддипломної практики в ТОВ «Теумах» є:

1. Набуття навичок роботи в колективі
2. Вивчення підсистем адміністрування ОС Linux.
3. Визначення задач для автоматизації, які можуть бути вирішені з використанням Python
4. Ознайомлення з інструментами для автоматизації адміністрування ОС Linux за допомогою Python
5. Розробка простих скриптів на Python для автоматизації рутинних процесів, керування ресурсами та обробки даних.

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

РОЗДІЛ 1. ВІДОМОСТІ ПРО МІСЦЕ ПРОХОДЖЕННЯ ПРАКТИКИ

1.1 Компанія-розробник ПЗ ТОВ «ТЕУМАХ»

ТОВ «ТЕУМАХ» – це молода, проте динамічно розвиваюча компанія з інформаційних технологій, заснована в 2017 році. Незважаючи на невеликий стаж, компанія успішно працює на ринку і зарекомендувала себе як надійний та професійний партнер для своїх клієнтів.

На сьогоднішній день в команді ТОВ "ТЕУМАХ" працює понад сто спеціалістів різних напрямків, серед яких розробники, дизайнери, тестувальники, менеджери з продажу та маркетингу. Компанія має широкий спектр послуг, які включають розробку програмного забезпечення, веб-розробку, мобільний додатки, системи управління базами даних та інші.

Директор ТОВ «ТЕУМАХ» – Остапенко Олександр, який володіє значним досвідом в галузі інформаційних технологій та успішно керує компанією, сприяючи її стабільному розвитку.



Рисунок 1.1 – Логотип ТОВ «ТЕУМАХ»

Компанія орієнтована на:

- Веб-розробку – роботу над вдосконаленням існуючих проєктів, створення веб- та мобільних додатків з нуля, SAAS-платформ, багато- та односторінкових веб-сайтів, MVP, своїх CRM-систем, ERP-систем, рішень Marketplace і т.д.
- Розробку мобільних додатків (створення функціональних і зручних нативних, кросплатформених мобільних додатків для iOS і Android).
- UI/UX дизайн – дизайн веб- та мобільних додатків, дизайн комплексних проєктів та платформ, дизайн прикладної дошки, дизайн виробу, брендинг.

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

В своїй роботі розробники використовують наступні технології:

1. Frontend: React, Redux, Next, Vue, Vuex, Nuxt;
2. Backend: Node.js, Next.js, PHP (переважно з Symfony), Python;
3. Бази даних: PostgreSQL, MySQL, MongoDB, DynamoDB, Firebase;
4. Mobile: React Native, NativeScript-Vue, Dart, Flutter.

1.2 Використання різних ОС серед розробників та адміністраторів

ТОВ «ТЕУМАХ» розташовується в одному із сучасних бізнес-центрів м. Житомир, де працює велика кількість ІТ-компаній, серед яких Infopulse, Uvidest та ін. Компанія в колективі має двох системних адміністраторів, які працюють у відділі Helpdesk.

В інших компаніях кількість варіюється в залежності від її розміру.

В малих компаніях працює лише один системний адміністратор, який відповідає за всю інфраструктуру компанії, або жодного. В середніх та великих компаніях є команди з декількох системних адміністраторів з різними спеціалізаціями, такими як мережеві адміністратори, адміністратори баз даних та інші.

Крім того, у великих ІТ-компаніях є окремі відділи з системного адміністрування, наприклад, відділ мережевого адміністрування, відділ баз даних тощо, які складаються з більшої кількості адміністраторів.

Системні адміністратори використовують різні операційні системи (ОС) в залежності від потреб компанії та її інфраструктури.

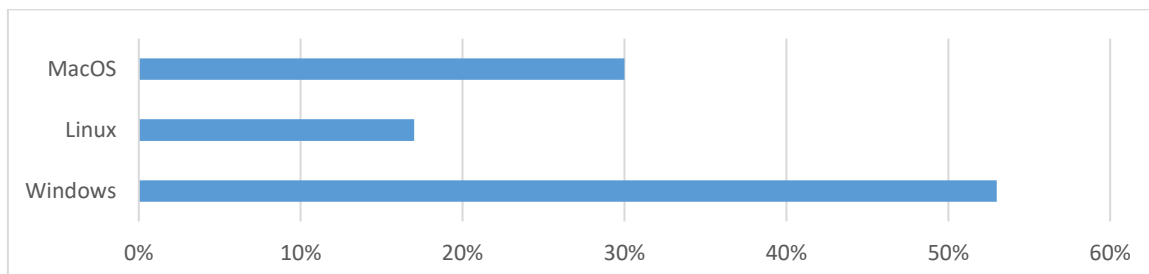
Windows лишається найпопулярнішою операційною системою серед українських ІТ-спеціалістів: нею користуються 52% розробників. Зростає популярність MacOS, відсоток використання наразі становить 43%. Частка користувачів Linux, навпаки, падає, і наразі становить лише 17%.

Windows найчастіше віддають перевагу Game Developers (88%), спеціалісти з ERP/CRM (86%), аналітики та підтримки (по 75%). Найактивніші користувачі MacOS – дизайнери (53% використовують цю ОС) та РМ (51%),

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

Linux – сисадміни (43%) та DevOps (42%), спеціалісти з Data Science (35%) та безпеки (30%).

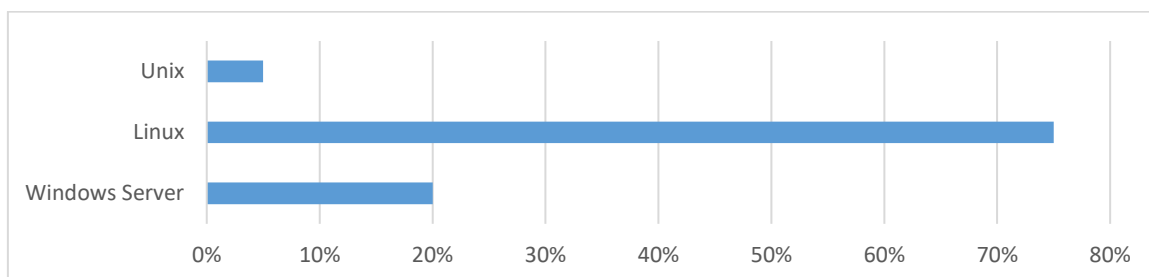
Інтерни та джуни частіше обирають Windows (понад 60% користуються цією ОС), в той час як спеціалісти рівня Senior та вище – MacOS (використовують понад 50% з них).



Діаграма 1.1 – Статистика використання різних ОС розробниками

Серед системних адміністраторів ситуація відрізняється – популярністю користуються такі операційні системи:

1. Linux – більше 75%;
2. Windows Server – більше 20%;
3. Unix – менше 5%.



Діаграма 1.2 – Статистика використання різних ОС системними адміністраторами

Актуальність Linux серед системних адміністраторів пов'язана з кількома факторами:

1. Відкритий вихідний код – Linux має відкритий вихідний код, що дозволяє системним адміністраторам вносити зміни у систему для вирішення специфічних проблем.

2. Безпека – Linux відома своєю безпекою і міцністю, що робить його популярним в бізнес-середовищах, де потрібна надійна захист від кібератак.
3. Надійність – Linux має високу надійність і стабільність, що дозволяє зменшити час простою системи і підвищити продуктивність.
4. Наявність безлічі інструментів – Linux має безліч інструментів для автоматизації та управління системою, що дозволяє системним адміністраторам ефективніше виконувати свої обов'язки.
5. Низькі витрати – Linux є безкоштовною операційною системою, що знижує витрати на ліцензію.
6. Висока популярність – Linux є однією з найпопулярніших операційних систем серед користувачів та розробників програмного забезпечення, що робить його логічним вибором для системних адміністраторів.

Однак, зважаючи на те, що Linux є більш технічною ОС, його використання вимагає більш глибоких знань та навичок від системних адміністраторів. Тому актуальність Linux в першу чергу пов'язана з наявністю кваліфікованих фахівців, які зможуть ефективно адмініструвати цю ОС.

Загалом, актуальність Linux серед системних адміністраторів пов'язана з його безкоштовністю, високою стабільністю та безпекою, а також з наявністю кваліфікованих фахівців, які зможуть ефективно адмініструвати цю ОС.

Висновки до першого розділу

В даному розділі було наведено відомості про місце проходження практики та проведено аналіз використання розробниками та системними адміністраторами цієї та інших компаній різних операційних систем.

В результаті сформовано та представлено статистику використання різних ОС та обґрунтовано популярність найчастіше використовуваних.

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

РОЗДІЛ 2. ОГЛЯД ПІДСИСТЕМ АДМІНІСТРУВАННЯ ОС LINUX

2.1 Підсистеми адміністрування ОС Linux

ОС Linux є однією з найпопулярніших операційних систем у світі, особливо в ІТ-середовищі. Для ефективного адміністрування ОС Linux інженери зазвичай використовують підсистеми адміністрування, які забезпечують автоматизацію багатьох процесів та спрощують роботу системного адміністратора.

Основні задачі, які можуть вирішуватися за допомогою підсистем адміністрування ОС Linux:

1. Управління пакетами: встановлення, оновлення та видалення програм, забезпечення безпеки та стабільності системи.
2. Моніторинг: збір та аналіз системної інформації, визначення проблем та потенційних загроз для системи.
3. Налаштування: настройка системи та її компонентів, забезпечення належного функціонування системи.
4. Автоматизація: створення скриптів та програм для автоматизації рутинних процесів, забезпечення швидкості та ефективності роботи.
5. Забезпечення безпеки: захист системи від вірусів, зломів та інших загроз, забезпечення безпеки даних.

Для розв'язання цих задач можна використовувати різні підсистеми адміністрування ОС Linux, наприклад:

- Systemd – стандартна система ініціалізації в Linux, що забезпечує автоматичне запускання сервісів та демонів під час завантаження системи, моніторинг процесів та контроль їх стану.
- Puppet – система конфігураційного управління, що дозволяє автоматизувати процеси налаштування та управління системою.

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

- Ansible – інструмент для автоматизації конфігурації та управління системами, забезпечує простоту використання та можливість працювати з багатьма ОС.
- Chef – система управління конфігураціями, що дозволяє налаштовувати та управляти великою кількістю серверів з однієї точки керування.
- SaltStack – інструмент для автоматизації управління та моніторингу великих мереж інфраструктури, що дозволяє керувати більшою кількістю серверів.

Кожна з цих підсистем має свої переваги та недоліки. В процесі проходження переддипломної практики було розглянуто найпопулярніші підсистеми адміністрування ОС Linux.

2.1.1 Systemd

Systemd – це стандартна система ініціалізації в більшості сучасних дистрибутивів ОС Linux. Це програмне забезпечення відповідає за запуск та управління процесами під час завантаження системи. Systemd був розроблений з метою заміни класичної системи ініціалізації SysVinit, яка була використовувана раніше в більшості дистрибутивів ОС Linux.

У Systemd існує централізований менеджер процесів, який відповідає за запуск, зупинку та моніторинг процесів в системі. Крім того, Systemd надає широкі можливості для автоматизації та налаштування процесів запуску та управління різноманітними сервісами та демонами в системі. Systemd також може служити системою журналювання подій, що дозволяє вести детальний моніторинг та аналізувати стан системи.

На рисунку 2.1 зображені компоненти, що входять до Systemd. Це спрощене представлення, яке показує лише основну архітектуру.

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

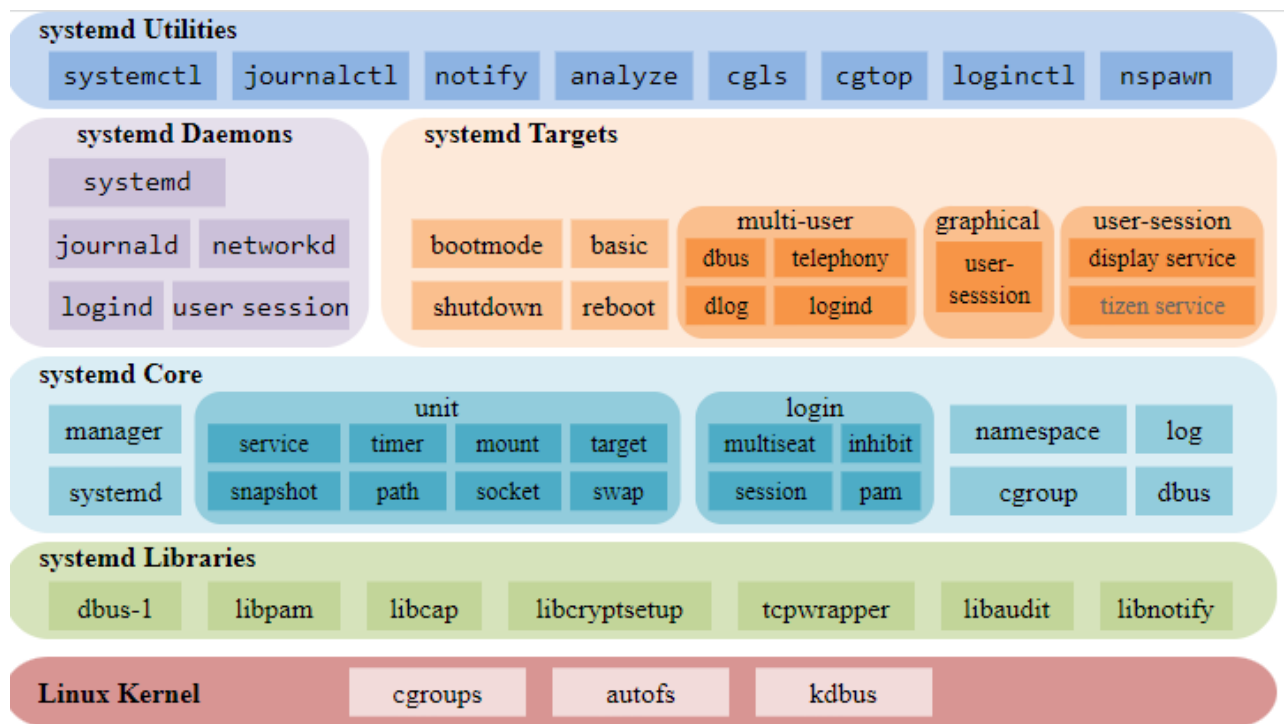


Рисунок 2.1 – Компоненти Systemd

Systemd має кілька переваг, серед яких:

1. Швидкість завантаження – systemd запускає процеси паралельно, що дозволяє зменшити час завантаження ОС.
2. Висока надійність – systemd має вбудовані засоби контролю над відповідністю процесів встановленим параметрам, відновлення процесів у разі помилок та попередження витoku пам'яті.
3. Простота налаштування – systemd забезпечує зручний інтерфейс для конфігурації та керування системою.
4. Широкий функціонал – systemd має багато вбудованих інструментів для автоматизації різних задач, таких як моніторинг процесів, логування, мережеві налаштування тощо.

Але, як і будь-яка технологія, systemd має й деякі недоліки:

1. Складність – systemd має складну структуру та вимагає від користувачів певного рівня знань та досвіду для ефективного використання.

2. Залежність від systemd – деякі користувачі вважають, що залежність від systemd обмежує можливості вибору інших інструментів та використання альтернативних ОС.
3. Вимоги до обладнання – для ефективної роботи systemd потребує обладнання з достатньою кількістю ядер процесора та достатньою кількістю оперативної пам'яті.
4. Некоректна робота з деякими програмами – у деяких випадках, systemd може працювати нестабільно з деякими програмами, які не дотримуються стандартів або мають специфічні вимоги до середовища виконання.

Systemd є стандартною системою ініціалізації в більшості сучасних дистрибутивів Linux, тому зазвичай вона вже встановлена в системі. Щоб перевірити її наявність, необхідно виконати наступну команду:

systemctl

Ця команда повинна показати перелік доступних служб і станів їх виконання.

```
student@student-VirtualBox:~$ systemctl
UNIT
proc-sys-fs-binfmt_misc.automount
dev-fuse.device
sys-devices-pci0000:00-0000:00:01.1-ata2-host1-target1:0:0-1:0:0:0-block-sr0.device
sys-devices-pci0000:00-0000:00:03.0-net-enp0s3.device
sys-devices-pci0000:00-0000:00:05.0-sound-card0.device
sys-devices-pci0000:00-0000:00:0d.0-ata3-host2-target2:0:0-2:0:0:0-block-sda-sda1.d
sys-devices-pci0000:00-0000:00:0d.0-ata3-host2-target2:0:0-2:0:0:0-block-sda-sda2.d
sys-devices-pci0000:00-0000:00:0d.0-ata3-host2-target2:0:0-2:0:0:0-block-sda-sda5.d
sys-devices-pci0000:00-0000:00:0d.0-ata3-host2-target2:0:0-2:0:0:0-block-sda.device
sys-devices-platform-serial8250-tty-ttyS0.device
sys-devices-platform-serial8250-tty-ttyS1.device
sys-devices-platform-serial8250-tty-ttyS10.device
sys-devices-platform-serial8250-tty-ttyS11.device
sys-devices-platform-serial8250-tty-ttyS12.device
sys-devices-platform-serial8250-tty-ttyS13.device
sys-devices-platform-serial8250-tty-ttyS14.device
```

Рисунок 2.2 – Перелік доступних служб і станів їх виконання

Як бачимо, Systemd наявна в системі. В разі ж відсутності її необхідно встановити за допомогою менеджера пакетів дистрибутиву Linux та перезавантажити систему. Для цього необхідно виконати наступні команди:

sudo apt-get install systemd

`sudo reboot`

Далі виконаємо кілька команд для моніторингу:

```
student@student-VirtualBox:~$ systemd-analyze blame
35.902s plymouth-quit-wait.service
25.050s vboxadd.service
 5.957s snapd.service
 3.627s dev-sda5.device
 3.017s networkd-dispatcher.service
 2.930s man-db.service
 2.695s udisks2.service
 2.515s accounts-daemon.service
 1.966s NetworkManager-wait-online.service
 1.695s logrotate.service
 1.629s NetworkManager.service
 1.565s avahi-daemon.service
 1.474s polkit.service
 1.196s systemd-resolved.service
```

Рисунок 2.3 – Перегляд часу, який потрібен для запуску програми чи служби

```
student@student-VirtualBox:~$ systemctl status ntp.service
● ntp.service - Network Time Service
   Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-03-27 03:37:00 EEST; 44s ago
     Docs: man:ntpd(8)
    Main PID: 9414 (ntpd)
      Tasks: 2 (limit: 1111)
     Memory: 1.3M
    CGroup: /system.slice/ntp.service
            └─9414 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 126:133

бep 27 03:37:04 student-VirtualBox ntpd[9414]: Soliciting pool server 91.236.251.14
бep 27 03:37:04 student-VirtualBox ntpd[9414]: Soliciting pool server 213.231.5.55
бep 27 03:37:04 student-VirtualBox ntpd[9414]: Soliciting pool server 162.159.200.123
бep 27 03:37:05 student-VirtualBox ntpd[9414]: Soliciting pool server 94.158.46.150
бep 27 03:37:05 student-VirtualBox ntpd[9414]: Soliciting pool server 82.193.104.168
бep 27 03:37:05 student-VirtualBox ntpd[9414]: Soliciting pool server 91.236.251.13
бep 27 03:37:05 student-VirtualBox ntpd[9414]: Soliciting pool server 91.189.94.4
бep 27 03:37:06 student-VirtualBox ntpd[9414]: Soliciting pool server 185.125.190.57
бep 27 03:37:06 student-VirtualBox ntpd[9414]: Soliciting pool server 79.142.192.4
бep 27 03:37:06 student-VirtualBox ntpd[9414]: Soliciting pool server 91.236.251.12
```

Рисунок 2.4 – Перевірка статусу служби

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

```

student@student-VirtualBox:~$ systemctl stop ntp.service
student@student-VirtualBox:~$ systemctl status ntp.service
● ntp.service - Network Time Service
   Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Mon 2023-03-27 03:39:28 EEST; 2s ago
     Docs: man:ntpd(8)
    Main PID: 9414 (code=exited, status=0/SUCCESS)

бep 27 03:37:05 student-VirtualBox ntpd[9414]: Soliciting pool server 94.158.46.150
бep 27 03:37:05 student-VirtualBox ntpd[9414]: Soliciting pool server 82.193.104.168
бep 27 03:37:05 student-VirtualBox ntpd[9414]: Soliciting pool server 91.236.251.13
бep 27 03:37:05 student-VirtualBox ntpd[9414]: Soliciting pool server 91.189.94.4
бep 27 03:37:06 student-VirtualBox ntpd[9414]: Soliciting pool server 185.125.190.57
бep 27 03:37:06 student-VirtualBox ntpd[9414]: Soliciting pool server 79.142.192.4
бep 27 03:37:06 student-VirtualBox ntpd[9414]: Soliciting pool server 91.236.251.12
бep 27 03:39:28 student-VirtualBox systemd[1]: Stopping Network Time Service...
бep 27 03:39:28 student-VirtualBox systemd[1]: ntp.service: Succeeded.
бep 27 03:39:28 student-VirtualBox systemd[1]: Stopped Network Time Service.
student@student-VirtualBox:~$ systemctl restart ntp.service
student@student-VirtualBox:~$ systemctl status ntp.service
● ntp.service - Network Time Service
   Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-03-27 03:39:54 EEST; 1s ago
     Docs: man:ntpd(8)
  Process: 10772 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited, status=0/

```

Рисунок 2.5 – Зупинка роботи служби та перезапуск

Щоб перевірити, чи присутній сервіс в автозагрузці, можна використати команду `systemctl is-enabled [назва_сервісу]`.

```

student@student-VirtualBox:~$ systemctl is-enabled ntp.service
enabled

```

Рисунок 2.6 – Перевірка наявності сервісу в автозагрузці

2.1.2 Puppet

Puppet – це система управління конфігурацією, що дозволяє автоматизувати процес налаштування та управління системою. За допомогою Puppet можна налаштувати різноманітні параметри на комп'ютерах, серверах та інших пристроях, що працюють під управлінням ОС Linux та інших операційних систем. Puppet дозволяє встановлювати програми, налаштовувати мережеві інтерфейси, файрволи, сертифікати, робити резервні копії та багато іншого.

Переваги використання Puppet для автоматизації адміністрування:

1. Ефективність: Puppet дозволяє ефективно керувати налаштуваннями на багатьох серверах.
2. Простота: Puppet забезпечує зручний і простий інтерфейс для зміни налаштувань сервера.

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Флексібільність: Puppet може працювати з різними ОС і різними конфігураціями.

4. Масштабованість: Puppet дозволяє легко налаштувати налаштування для великої кількості серверів.

Недоліки використання Puppet для автоматизації адміністрування:

1. Складність: Puppet може бути складним для налаштування і вимагає певного рівня знань.

2. Залежність: Puppet залежить від роботи самої системи ініціалізації.

3. Навчання: використання Puppet може вимагати додаткового часу для навчання та оволодіння.

4. Витрати: Puppet може вимагати додаткових витрат на обладнання та програмне забезпечення для налагодження та роботи.

Основна ідея роботи з Puppet полягає в тому, щоб зберігати конфігурацію серверів в файлових конфігураціях, які можуть бути контрольовані, редаговані та збережені за допомогою системи контролю версій (наприклад, Git).

Розглянемо приклад роботи з Puppet на прикладі встановлення та налаштування веб-сервера Apache на сервері з операційною системою Ubuntu 20.04.

Для початку встановлюємо дві віртуальні машини Ubuntu 20.04 у VirtualBox та налаштовуємо їх:

```
student@puppet-Agent:~$ sudo ip addr add 192.191.0.2/255.255.255.0 broadcast 192.191.0.255 dev enp0s3
student@puppet-Agent:~$ sudo ip route add default via 192.191.0.1
student@puppet-Agent:~$ ip addr show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:2c:91:7d brd ff:ff:ff:ff:ff:ff
    inet 192.191.0.2/24 brd 192.191.0.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe2c:917d/64 scope link
        valid_lft forever preferred_lft forever
```

Рисунок 2.7 – Налаштування IP-адресації на першому сервері

```

student@student-VirtualBox:~$ sudo ip addr add 192.191.0.1/255.255.255.0 broadcast 192.191.0.255 dev enp0s3
student@student-VirtualBox:~$ sudo ip route add default via 192.191.0.2
student@student-VirtualBox:~$ ip addr show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:2c:91:7d brd ff:ff:ff:ff:ff:ff
    inet 192.191.0.1/24 brd 192.191.0.255 scope global enp0s3
        valid_lft forever preferred_lft forever
student@student-VirtualBox:~$ ping 192.191.0.2
ЛУНА-ІМПУЛЬС 192.191.0.2 (192.191.0.2) 56(84) байтів даних.
64 байтів з 192.191.0.2: icmp_seq=1 ttl=64 час=1.04 мс
64 байтів з 192.191.0.2: icmp_seq=2 ttl=64 час=0.694 мс
64 байтів з 192.191.0.2: icmp_seq=3 ttl=64 час=0.702 мс
^C
--- Статистика луна-імпульсів 192.191.0.2 ---
Передано 3 пакетів, отримано 3, втрачено 0% пакетів, час 2008 мс
rtt мін/сер/макс/мdev = 0.694/0.810/1.036/0.159 мс

```

Рисунок 2.8 – Налаштування IP-адресації на другому сервері та перевірка

Встановлюємо Puppet Master та Puppet Agent на сервері Ubuntu 20.04:

```
sudo apt-get update
```

```
sudo apt-get install puppetmaster puppet
```

```

Отр:70 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata
a [516 B]
Отримано 16,6 MB за 4сВ (3 705 kB/s)

Зчитування переліків пакунків... Виконано
student@student-VirtualBox:~$
student@student-VirtualBox:~$ sudo apt-get install puppetmaster puppet
Зчитування переліків пакунків... Виконано
Побудова дерева залежностей
Зчитування інформації про стан... Виконано
Буде встановлено такі додаткові пакунки:
  augeas-lenses debconf-utils facter fonts-lato hiera javascript-common libaugeas0
  libboost-log1.71.0 libboost-program-options1.71.0 libboost-regex1.71.0
  libcpp-hocon0.1.7 libcurl4 libfacter3.11.0 libjs-jquery libleatherman1.4.2

```

Рисунок 2.7 – Встановлення Puppet Master та Puppet Agent

Далі встановлюємо пакет Puppet-Agent на другому сервері. Для цього додаємо репозиторій Puppet до списку репозиторіїв системи:

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

```

student@student-VirtualBox:~$ wget https://apt.puppetlabs.com/puppet7-release-focal.deb
--2023-03-27 05:04:32-- https://apt.puppetlabs.com/puppet7-release-focal.deb
Визначення імені apt.puppetlabs.com (apt.puppetlabs.com)... 52.84.195.121, 52.84.195.67, 52.84.195.84, ...
Встановлення з'єднання з apt.puppetlabs.com (apt.puppetlabs.com)|52.84.195.121|:443... з'єднано.
HTTP-запит надіслано, очікуємо на відповідь... 200 OK
Довжина: 11720 (11K) [application/x-debian-package]
Зберігаємо до «puppet7-release-focal.deb»

puppet7-release-f 100%[=====] 11,45K --.-KB/s за 0s

2023-03-27 05:04:32 (173 MB/s) - «puppet7-release-focal.deb» збережено [11720/11720]

student@student-VirtualBox:~$ sudo dpkg -i puppet7-release-focal.deb
Вибір ранише не обраного пакунку puppet7-release.
dpkg: розглядується вилучення puppet6-release, замість якого буде встановлено puppet7-release ...
dpkg: так, пакунок puppet6-release буде вилучено, щоб можна було встановити puppet7-release

```

Рисунок 2.8 – Додавання репозиторію Puppet до списку репозиторіїв системи

Оновлюємо список пакетів за допомогою команди *sudo apt update* та встановлюємо пакет puppet-agent:

```

student@student-VirtualBox:~$ sudo apt install puppet-agent
Зчитування переліків пакунків... Виконано
Побудова дерева залежностей
Зчитування інформації про стан... Виконано
Наступні пакунки були встановлені автоматично і більше не потрібні:
augeas-lenses fonts-lato javascript-common libaugeas0 libjs-jquery
libruby2.7 linux-headers-5.4.0-47 linux-headers-5.4.0-47-generic
linux-image-5.4.0-47-generic linux-modules-5.4.0-47-generic
linux-modules-extra-5.4.0-47-generic rake ruby ruby-augeas
ruby-deep-merge ruby-minitest ruby-net-telnet ruby-power-assert

```

Рисунок 2.9 – Встановлення пакету

Прописуємо сервер:

```

GNU nano 4.8 /etc/puppetlabs/puppet/puppet.conf
# This file can be used to override the default puppet settings
# See the following links for more details on what settings
# - https://puppet.com/docs/puppet/latest/config_important_settings.html
# - https://puppet.com/docs/puppet/latest/config_about_settings.html
# - https://puppet.com/docs/puppet/latest/config_file_main.html
# - https://puppet.com/docs/puppet/latest/configuration.html
[main]
certname = puppetAgent
server = student-VirtualBox

```

Рисунок 2.10 – Прописування головного сервера

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Для зручності перейменуємо другий сервер на puppet-Agent:

```
student@student-VirtualBox:~$ sudo hostnamectl set-hostname puppet-Agent
student@student-VirtualBox:~$ sudo reboot
```

Рисунок 2.11 – Перейменовування сервера

Запускаємо puppet на головному сервері та додаємо сервіс до списку сервісів для автозавантажень:

```
student@student-VirtualBox:~$ sudo systemctl start puppet
student@student-VirtualBox:~$ sudo systemctl enable puppet
```

Рисунок 2.12 – Запуск puppet

Тепер запускаємо puppet на другому сервері:

```
student@puppet-Agent:~$ sudo systemctl start puppet
student@puppet-Agent:~$ sudo systemctl status puppet
● puppet.service - Puppet agent
   Loaded: loaded (/lib/systemd/system/puppet.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-03-27 05:56:41 EEST; 1s ago
     Docs: man:puppet-agent(8)
    Main PID: 2878 (puppet)
      Tasks: 1 (limit: 1111)
     Memory: 61.0M
    CGroup: /system.slice/puppet.service
            └─2878 /opt/puppetlabs/puppet/bin/ruby /opt/puppetlabs/puppet/bin/puppet agent --no-daemonize --no-color --no-show-progress --no-scheduler --no-trace --no-use-scheduler --no-version-check --verbose

ber 27 05:56:41 puppet-Agent systemd[1]: Started Puppet agent.
lines 1-11/11 (END)
```

Рисунок 2.13 – Запуск puppet на puppet-Agent

Конфігуруємо Puppet Master, створюючи файл /etc/puppet/manifests/site.pp з таким змістом:

```
GNU nano 4.8 /etc/puppet/manifests/site.pp
node 'server.example.com' {
    include apache
}

class apache {
    package {'apache2':
        ensure => 'installed',
    }
    service {'apache2':
        ensure => 'running',
        enable => true,
    }
}
```

Рисунок 2.14 – Конфігурування Puppet Master

Тут використовується Puppet для налаштування веб-сервера Apache на сервері з іменем server.example.com. Створюється новий клас apache, який містить два налаштування: встановлення пакету Apache2 та налаштування його сервісу таким чином, щоб він завжди був запущеним та автоматично запускався під час завантаження системи.

Створюємо асоціації на кожному із серверів:

```
GNU nano 4.8 /etc
127.0.0.1    localhost
127.0.1.1    student-VirtualBox
192.191.0.2  puppet-Agent
```

Рисунок 2.15 – Створення хосту на головному сервері

На другому серверві прописується хост для puppet-master (192.191.0.1).

Копіюємо файл site.pp з Puppet Master на Puppet Agent за допомогою наступної команди:

```
sudo scp /etc/puppet/manifests/site.pp student@puppet-Agent:/etc/puppet/manifests/site.pp
```

Після виконання команди вивід може бути різним, залежно від того, чи успішно відбулася передача файлу на віддалений сервер. Так як все пройшло успішно, вивід порожній.

Після виконання команди *sudo puppet agent --test* Puppet агент почне зв'язок з Puppet сервером і спробує завантажити конфігурацію. Вивід команди буде залежати від того, яку конфігурацію було застосовано і які модулі використовуються. Очікуваний результат – успішне завершення з виведенням повідомлень про те, як Puppet агент оновлює конфігурацію на сервері.

В даному випадку Puppet зчитає конфігурацію, яка включає налаштування для Apache, та перевірить, чи всі необхідні пакети встановлені та сервіс запущений. Вивід був наступний:

Info: Retrieving pluginfacts

Info: Retrieving plugin

Info: Caching catalog for student-VirtualBox

Info: Applying configuration version 'timestamp'

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

Notice: /Stage[main]/Apache/Package[apache2]/ensure: created

Notice: /Stage[main]/Apache/Service[apache2]/ensure: ensure changed 'stopped' to 'running'

Notice: Applied catalog in 0.10 seconds

При цьому Puppet спочатку виконує попередні завдання, такі як зчитування даних про плагіни та кешування каталогу для швидкого доступу до нього. Далі виводиться повідомлення про те, що застосовується конфігураційна версія, яка містить налаштування для Apache. На останок, Puppet повідомляє, які зміни були застосовані до системи, які сервіси були запущені або зупинені.

2.2 Визначення задач для автоматизації

Python – відмінна мова програмування для автоматизації завдань системного адміністрування в системах Linux. Існують кілька причин, чому вона так широко використовується:

1. Простота використання: Python має простий і зрозумілий синтаксис, що дозволяє швидко створювати скрипти для автоматизації рутинних задач.
2. Широкі можливості: Python має велику кількість розширень та бібліотек, що дозволяє легко реалізовувати різноманітні задачі.
3. Крос-платформенність: Python може працювати на різних операційних системах, що дозволяє розробляти універсальні скрипти для автоматизації.
4. Відкритий код: Python є вільним програмним забезпеченням, що дозволяє розробникам створювати власні рішення для автоматизації без додаткових витрат.
5. Підтримка комунітету: Python має велику спільноту розробників та користувачів, що активно розробляє нові бібліотеки та розв'язує проблеми використання мови.

Python може бути використаний для автоматизації багатьох задач, пов'язаних з адмініструванням підсистеми Linux. Деякі з цих задач можуть включати:

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

1. Налаштування сервера: Python може бути використаний для автоматизації налаштування сервера, включаючи інсталяцію необхідного програмного забезпечення, конфігурування мережевих налаштувань і створення потрібних користувачів та груп.
2. Моніторинг системи: Python може бути використаний для створення моніторингових скриптів, які дозволяють відстежувати різноманітні метрики системи, такі як використання процесора, пам'яті та дискового простору, та сповіщати адміністратора про будь-які відхилення від норми.
3. Автоматизація резервного копіювання: Python може бути використаний для автоматизації процесу резервного копіювання, включаючи створення копій файлів та баз даних та їх збереження на зовнішніх пристроях або в хмарному сховищі.
4. Забезпечення безпеки: Python може бути використаний для автоматизації процесу забезпечення безпеки системи, включаючи встановлення і налаштування захисту від зловмисників, які намагаються зламати систему або отримати незаконний доступ до неї.
5. Автоматизація задач системного адміністрування: Python може бути використаний для автоматизації рутинних задач системного адміністрування, таких як оновлення програмного забезпечення, налаштування баз даних, моніторинг стану сервера тощо.

Висновки до другого розділу

В даному розділі було розглянуто найпопулярніші підсистеми адміністрування ОС Linux та показано роботу із ними. Наведені основні команди для роботи із Systemd та проведено моніторинг програм та сервісів. За допомогою підсистеми Puppet встановлено та налаштовано веб-сервер Apache на сервері Ubuntu 20.04.

Визначено, які задачі можуть бути автоматизовані за допомогою мови програмування Python.

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

РОЗДІЛ 3. ВИКОРИСТАННЯ МОВИ ПРОГРАМУВАННЯ PYTHON В АДМІНІСТРУВАННІ ОС LINUX

3.1 Основні модулі Python для адміністрування ОС Linux

Для адміністрування Linux за допомогою Python зазвичай використовують різноманітні модулі, які забезпечують доступ до системних ресурсів, можливість взаємодії з іншими програмами, мережеві операції та багато іншого.

Модулі на Python для адміністрування Linux – це збірники попередньо написаних функцій і класів, які надають можливості для автоматизації адміністративних завдань у Linux-середовищі. Ці модулі містять різні функції і методи, які допомагають працювати з файловою системою, мережевими налаштуваннями, процесами та іншими системними ресурсами.

Такі модулі можуть включати у себе вбудовані бібліотеки Python, такі як `os`, `sys`, `shutil`, `subprocess`, або сторонні модулі, такі як `paramiko` для виконання команд на віддалених машинах по SSH, `psutil` для моніторингу процесів та системних ресурсів або `netifaces` для роботи з мережевими інтерфейсами.

Завдяки наявності готових модулів Python для роботи з різними системними ресурсами, розробники можуть значно ефективніше та швидше створювати автоматизовані скрипти для адміністрування Linux-серверів та виконувати різноманітні завдання, такі як моніторинг системи, збір та аналіз лог-файлів, копіювання та переміщення файлів, налаштування мережевих інтерфейсів тощо.

Модулі ОС (`os`) та `Subprocess` (`subprocess`) – це два з основних модулів Python, які використовують для адміністрування Linux.

Модуль `os` дозволяє взаємодіяти з операційною системою, виконувати команди `shell`, доступатися до змінних середовища та ін. Цей модуль включає багато функцій для роботи з файловою системою, дозволяє зчитувати та записувати файли, отримувати інформацію про процеси, мережеві інтерфейси та інше. Наприклад, з його допомогою можна виконувати наступні дії:

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

- отримати список файлів та директорій в поточній директорії;
- створити нову директорію;
- отримати інформацію про систему (таку як ім'я операційної системи, кількість процесорів тощо).

Основні команди модуля os включають:

- `os.getcwd()`: повертає поточну робочу директорію.
- `os.chdir(path)`: змінює поточну робочу директорію на вказану.
- `os.listdir(path)`: повертає список файлів та директорій в заданій директорії.
- `os.mkdir(path)`: створює нову директорію з вказаним ім'ям.
- `os.rmdir(path)`: видаляє директорію з вказаним ім'ям (директорія повинна бути порожньою).
- `os.remove(path)`: видаляє файл з вказаним ім'ям.
- `os.rename(src, dst)`: перейменовує файл або директорію з `src` на `dst`.
- `os.system(command)`: виконує команду в командному рядку.
- `os.getenv(varname)`: повертає значення змінної середовища з заданим ім'ям.
- `os.getpid()`: повертає ідентифікатор поточного процесу.
- `os.kill(pid, signal)`: надсилає сигнал до процесу з вказаним ідентифікатором.

Це лише деякі з основних команд, які можна виконувати з модулем `os`. Модуль також містить багато інших корисних функцій для взаємодії з операційною системою.

Модуль `Subprocess` дозволяє запускати зовнішні процеси з Python-скрипту та зчитувати та записувати дані в їхній вхід та вихід. Цей модуль надає більш гнучкий та безпечний спосіб виконання зовнішніх команд. Він дозволяє створювати підпроцеси, перенаправляти вхід та вихід, запускати команди в фоновому режимі та інше.

Наприклад, з його допомогою можна виконувати наступні дії:

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

- запустити команду в shell та зчитати її вихід;
- запустити багато команд та підключитися до їх вхідного та вихідного потоків;
- запустити команду в фоновому режимі та отримати її PID (ідентифікатор процесу).

Модуль `socket` є одним з основних модулів Python для роботи з мережевими операціями. Він надає інтерфейс для створення сокетів, що дозволяє здійснювати з'єднання з віддаленими серверами, відправляти та отримувати дані через мережу, використовуючи різні протоколи (наприклад, TCP або UDP).

Цей модуль є досить потужним і дозволяє розробникам створювати складні мережеві додатки, такі як веб-сервери, клієнти для мережеских протоколів, інструменти для тестування мережеских додатків тощо. За допомогою модуля `socket` можна отримати доступ до різноманітних функцій мережевої роботи і керувати ними в програмах на Python.

Деякі основні команди, що можна виконувати за допомогою цього модуля:

- `socket()`: створює новий сокет.
- `bind()`: прив'язує сокет до конкретного IP-адреси та порту.
- `listen()`: дозволяє серверу приймати вхідні з'єднання.
- `accept()`: приймає вхідне з'єднання та створює новий сокет для обміну даними.
- `connect()`: встановлює з'єднання з віддаленим сервером.
- `send()`: відправляє дані через сокет.
- `recv()`: отримує дані через сокет.
- `close()`: закриває сокет.

Ці команди дозволяють створювати мережеві додатки, які можуть взаємодіяти з іншими пристроями через мережу. Для більш детальної інформації можна переглянути документацію модуля `socket` в Python.

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Написання простих скриптів для адміністрування Linux

Для більшості дистрибутивів Linux Python вже встановлений за замовчуванням. Однак, якщо необхідно встановити новішу версію Python або перевірити, чи встановлена версія Python у системі, необхідно прописати наступну команду:

```
student@student-VirtualBox:~$ python3
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Рисунок 3.1 – Перевірка наявності Python в системі

В разі, якщо інтерпретатор Python не відкриється, необхідно встановити його за допомогою менеджера пакетів дистрибутива:

```
student@student-VirtualBox:~$ sudo apt install python3
Зчитування переліків пакунків... Виконано
Побудова дерева залежностей
Зчитування інформації про стан... Виконано
python3 вже є новою версією (3.8.2-0ubuntu2).
python3 позначений як встановлений вручну.
оновлено 0, встановлено 0 нових, 0 відмічено для видалення і 612 не оновлено.
student@student-VirtualBox:~$
```

Рисунок 3.2 – Встановлення Python

Далі необхідно створити файл із розширенням .py в текстовому редакторі Nano або Vim.

Для прикладу пропишемо наступний скрипт для створення нового каталогу:

```
import os

dir_name = "example"

try:

    os.mkdir(dir_name)
    print(f"Directory '{dir_name}' created successfully")

except FileExistsError:
    print(f"Directory '{dir_name}' already exists")
```

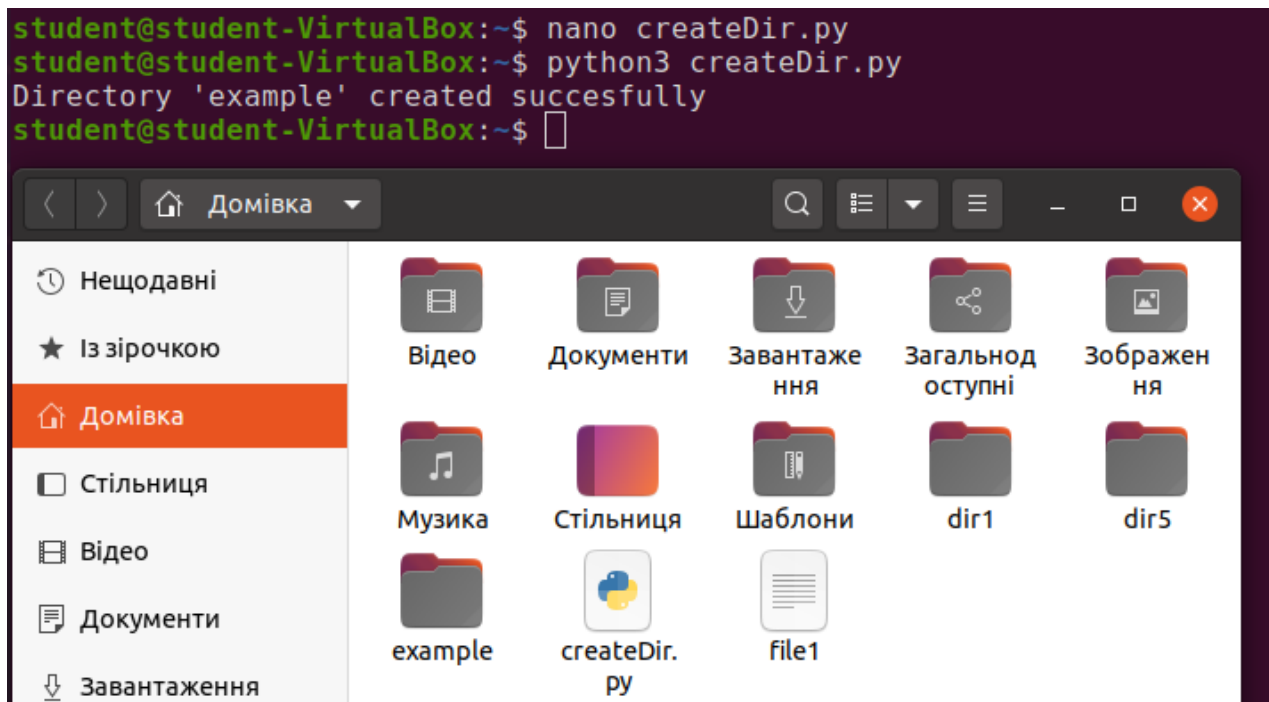


Рисунок 3.3 – Створені файл та каталог

Тепер пропишемо скрипт для видалення каталогу:

```
#Import the OS module
import os

#Name of the directory to be deleted
dir_name = "example"

try:

#Deletes the directory
    os.rmdir(dir_name)

#Prints the result, if the directory was successfully deleted
    print(f"Directory '{dir_name}' deleted successfully")

#Prints the result, if the directory doesn't exist
except FileNotFoundError:
    print(f"Directory '{dir_name}' doesn't exist")
```

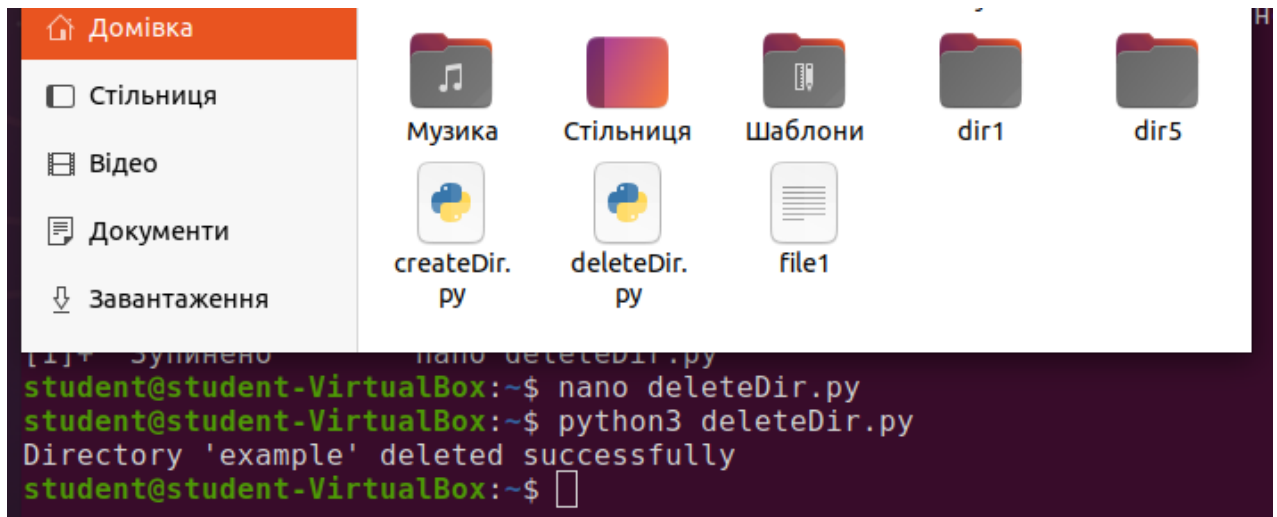


Рисунок 3.4 – Каталог видалено

Далі пропишемо скрипт для перевірки IP-адреси та імені хоста системи:

```
import socket
```

```
try:
```

```
    host = socket.gethostname()
    ip = socket.gethostbyname(host)
    print(f"IP address: {ip}")
    print(f"Hostname: {host}")
```

```
except:
```

```
    print("An error occurred")
```

```
student@student-VirtualBox:~$ nano ip.py
student@student-VirtualBox:~$ python3 ip.py
IP address: 127.0.1.1
Hostname: student-VirtualBox
student@student-VirtualBox:~$
```

Рисунок 3.5 – IP-адреса та хост системи

Ось приклад скрипту Python, який допомагає автоматично створювати користувачів на Linux з використанням модуля subprocess:

```
import subprocess
```

```
def create_user(username, password):
```

```
    try:
```

```
        subprocess.check_output(['sudo', 'useradd', '-p', password, '-m', username])
        print('User {} created successfully.'.format(username))
```

```
    except subprocess.CalledProcessError as e:
```

```
        print('Error creating user: {}'.format(e))
```

```
username = input('Enter username: ')
password = input('Enter password: ')
```

```
create_user(username, password)
```

Цей скрипт запитує ім'я користувача та пароль, а потім використовує команду `useradd` разом з параметрами для створення нового користувача в системі. Якщо процес завершується успішно, скрипт повідомляє про успіх. Якщо виникає помилка, виводиться повідомлення про помилку.

```
student@student-VirtualBox:~$ nano add.py
student@student-VirtualBox:~$ python3 add.py
Enter username: mariana
Enter password: mariana
[sudo] пароль до student:
User mariana created successfully.
```

Рисунок 3.6 – Створення користувача

Перевіримо створення користувача за допомогою команди:

```
cat /etc/passwd | cut -d: -f1
```

Ця команда використовує роздільник `:` та вибирає перше поле (ім'я користувача) з кожного рядка виведення.

```
gdm
student
systemd-coredump
vboxadd
ntp
puppet
mariana
student@student-VirtualBox
```

Рисунок 3.7 – Створені користувачі

Напишемо скрипт `python` для автоматизації системи таким чином, щоб різні типи файлів (наприклад, `mp3`, `png`, `xls` тощо) йшли в різні каталоги. Повний код наведено нижче. Цей код перевірятиме кожен тип файлу по одному, а потім створить відповідний каталог. Після цього він перемістить файл у пов'язані каталоги (наприклад, файл `.mp3` потрапить до каталогу "Audio").

Встановлюємо планову задачу для виконання цієї функції щоденно о 12:00 за допомогою модуля `schedule`. Треба запустити скрипт і він буде виконуватися щоденно о 12:00.

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import os
import shutil
import schedule
import time

watch_folder = '/home/user/Downloads'

file_types = {
    ".mp3": "Аудіо",
    ".png": "Зображення",
    ".xls": "Документи",
    ".txt": "Документи",
    ".pdf": "Документи",
    ".zip": "Документи"
}

def sort_files():
    for filename in os.listdir(downloads_path):
        file_path = os.path.join(downloads_path, filename)

        if os.path.isdir(file_path) or filename.startswith("."):
            continue

        file_ext = os.path.splitext(filename)[1]

        if file_ext in file_types:
            dest_folder = os.path.join(downloads_path, file_types[file_ext])
            os.makedirs(dest_folder, exist_ok=True)
            dest_path = os.path.join(dest_folder, filename)
            shutil.move(file_path, dest_path)

schedule.every().day.at("12:00").do(sort_files)

while True:
    schedule.run_pending()
    time.sleep(1)

```

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

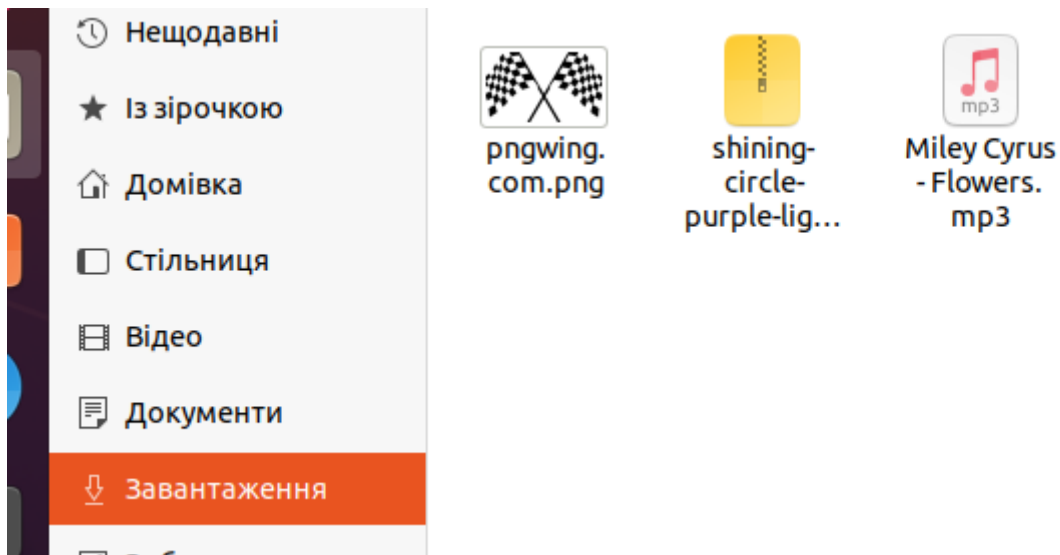


Рисунок 3.8 – Папка «Завантаження» до виконання скрипта

```
student@student-VirtualBox:~$ nano sorted.py
student@student-VirtualBox:~$ python3 sorted.py
```

Рисунок 3.10 – Виконання скрипта

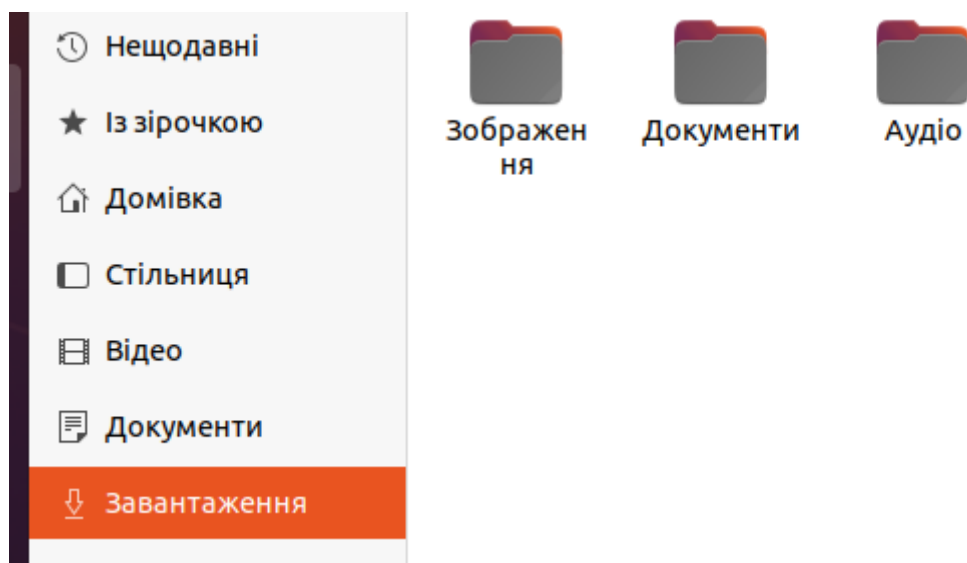


Рисунок 3.11 – Результат

Висновки до третього розділу

В результаті написання третього розділу було ознайомлено із основними модулями Python для автоматизації адміністрування ОС Linux.

Також були написані прості скрипти для роботи із файловою системою.

ВИСНОВКИ

Під час проходження переддипломної практики у ТОВ «ТЕУМАХ» було отримано багато практичних навичок.

Було розглянуто найпопулярніші підсистеми адміністрування ОС Linux та показано роботу із ними. Опрацьовано основні команди для роботи із Systemd та проведено моніторинг програм та сервісів. За допомогою підсистеми Puppet встановлено та налаштовано веб-сервер Apache на сервері Ubuntu 20.04.

Визначено, які задачі можуть бути автоматизовані за допомогою мови програмування Python.

Було ознайомлено із основними модулями Python для автоматизації адміністрування ОС Linux. Навчено писати найпростіші та складні скрипти для роботи із файловою системою ОС Linux.

		Матвієнко М.В.			«Житомирська політехніка».23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Роббінс А. Bash. Кишеньковий довідник системного адміністратора / Арнольд Роббінс., 2022. – 152 с. – (2). – (Операційні системи).
2. Немет Е. Unix та Linux. Керівництво системного адміністратора / Е. Немет, Г. Снайдер, Х. Трент., 2022. – 734 с. – (5).
3. Гіфт Н. Python і DevOps. Ключ до автоматизації Linux / Н. Гіфт, К. Берман, А. Деза., 2022. – 544 с. – (O'Reilly).
4. Гіфт Н. Python в системному адмініструванні Unix і Linux / Н. Гіфт, М. Джеремі, М. Джонс., 2018. – 512 с. – (O'Reilly).
5. Рамський Ю. С. Адміністрування комп'ютерних мереж та систем. Навч. посіб. / Юрій Савіянович Рамський.. – 196 с.
6. Колісніченко Д. LINUX. Повне керівництво по роботі і адміністрування / Д. Колісніченко., 2020. – 480 с.

		Матвієнко М.В.			«Житомирська політехніка». 23.123.11.000 – ПП	Арк.
		Окунькова О.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		33