Міністерство оборони України
Військовий інститут телекомунікацій та інформатизації
імені Героїв Крут

Кафедра
Комп'ютерних інформаційних технологій

Курсова робота

з дисципліни: «Операційні системи»
на тему: «Автоматизація встановлення та налаштування серверної компоненти (поштовий сервер) ОС *Linux* (*Debian*)»

Виконав:
Курсант 281 навчальної групи
солд. А.Гонорович
Перевірив:
Доцент кафедри №22
п/п-к Ю.Здоренко

Київ-2020

АНОТАЦІЯ

курсової роботи на тему:

«Автоматизація встановлення та налаштування серверної компоненти (поштового серверу) ОС *Linux* (*Debian*)»

Курсова робота містить: 21 сторінку, 1 рисунок та 9 джерел.

В даній курсовій роботі висвітлено приклад використання *Bash* для автоматизації адміністрування операційної системи *Debian* за допомогою якої спрощено всі особливості встановлення та налаштування програмного забезпечення серверної компоненти *Moodle* відповідно до поставленого завдання.

Результатом роботи є скрипт для роботи з операційною системою персонального комп’ютера, обробки даних згідно поставленим завданням, встановлення та налаштування відповідних компонентів *OS*.

ANNOTATION

course work on the topic:

«Automate the installation and configuration of the server component (mail server) Linux (Debian)»

Course work contains: 21 pages, 1 drawing and 9 sources.

This course paper illustrates the use of the Bash to automate administration of the Debian operating system, which simplifies all aspects of installing and configuring the Moodle server component software in accordance with the task.

The result of the work is a script to work with the operating system of the personal computer, data processing in accordance with the tasks, installation and configuration of the relevant components of the OS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface

CSS – Cascading Style Sheets

GUI – Graphical User Interface

IDE – Integrated Development Environment

OS – Operating System

RIA – Rich Internet Application

UML – Unified Modeling Language

ВВНЗ – Вищий Військовий Навчальний Заклад

ВНЗ – Вищий Навчальний Заклад

ЕОМ – Електронно Обчислювана Машина

ЗСУ – Збройні Сили України

НАТО – Організація Північноатлантичного договору

ПК – Персональний Комп’ютер

ВСТУП

Актуальність теми. Бурхливий розвиток обчислювальної техніки, необхідність в ефективних засобах розробки програмного забезпечення призвели до появи скриптових сценаріїв, орієнтованих на так зване "швидке розгортання". Відповідно всі зміни в інформаційному просторі сягають і ЗСУ, що впливає на їх подальший розвиток в даній сфері. Але кількість військовослужбовців та їх обов’язки фізично не дають здійснювати навчання всіх одночасно, але на допомогу цьому існує дистанційне навчання. За допомогою системи дистанційного навчання *Moodle,* вдається здійснювати навчання майже з усіма одразу, адже на сервері зберігаються всі потрібні дані для саморозвитку. Таким чином спростивши процедуру встановлення і налаштування такого серверу ми значно пришвидшимо розвиток інформаційних систем в ЗСУ.

Мета роботи: автоматизувати процес встановлення та налаштування серверної компоненти *Moodle* на ОС *Debian*.

Виходячи з мети роботи виникають наступні варіанти реалізації поставленого завдання:

— проаналізувати поставлене завдання;

— проаналізувати вимоги до скрипта та системи, спроектувати та розробити скрипт на Bash;

— проаналізувати особливості використання мови програмування Python на операційних системах сімейства *Unix*.

Об’єкт дослідження – процес встановлення та налаштування серверної компоненти *Moodle* на операційній системі *Debian*.

Предметом дослідження є скриптовий сценарій *Mail* на мові програмування компонента поштового серверу *Bash* для автоматизації, як система дистанційного навчання та налаштування серверної компоненти на операційній системі сімейства *OS Unix* (*Debian*).

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Постановка задачі

Ідея скрипта:

— редагування конфігурації системи для створення сприятливих умов для взаємодії та встановлення серверної компоненти бази даних;

— встановлення та налаштування актуальної версії поштового серверу на OS Debian.

Скрипт повинен допомогти:

— офіцерам ЗСУ – під час роботи з *OS* сімейства *Unix* та поштовим сервером;

— курсантам та викладачам під час автоматизації встановлення необхідного ПЗ на ПК та його базового конфігурування.

Скрипт матиме такі пункти:

— підготовка *OS* до роботи;

— встановлення необхідного поштового серверу;

— налаштування встановленого поштового серверу;

— обробка результатів та передача управління користувачу поштового серверу.

1.2 Функціональні можливості поштового серверу

Поштовий сервер — програма, що забезпечує роботу служби із сторони Інтернету. Як і всі служби Інтернету, електронна пошта заснована на взаємодії двох програм. Одна з цих програм — сервер, а інша — клієнт. Вони

взаємодіють за визначеними правилами, заданими у протоколах. Поштовий клієнт - це програма, яка встановлена на комп'ютері користувача й забезпечує взаємодію з поштовим сервером. Можливості:

— Отримує пошту, що прийшла від клієнта та доставляє її на поштовий сервер адресата.

— Приймає та накопичує пошту, що надійшла з Інтернету.

— Фільтрує пошту від *SPAM*-у згідно заданих критеріїв та блокує заражені вірусами листи.

— Переадресовує пошту, що приходить на вказані клієнтом адреси. Це може бути корисно, коли деякі працівники використовують інші адреси, або для організації «службових адрес», які не закріплені за певною поштовою скринькою.

— Підтримує поштові скриньки для працівників, що знаходяться поза межами офісу, навіть в іншій країні. Протоколи доступу - *POP*3, *IMAP*, *HTTP*.

— Обслуговує організовані списки розсилки.

Можливості управління ВПС:

— створення скриньок власноручно.

— створення аліасів.

— створення списків розсилки.

У системах протоколу *SMTP* (*Simple Mail Transfer Protocol*) реалізовано пряме доставляння повідомлення адресату. Адреса електронної пошти, наприклад, може мати вигляд: *user1@dom1.dom2.com*

Поштова система протоколу *SMTP* використовує цілий набір протоколів, кожен з яких відіграє свою роль у процесі передавання.

Протокол *SMTP* використовують для передавання повідомлень між поштовими серверами. Він працює з портом 25 та протоколом *TCP*.

Simple Mail Transfer Protocol (Простий Протокол Пересилання Пошти) — це протокол, який використовується для пересилання електронної пошти до поштового сервера або з клієнта-комп'ютера, або між поштовими серверами. В *IANA* для *SMTP* зареєстровано порт 25. Формально *SMTP* визначений в *RFC* 821 (*STD* 10) та покращений *RFC* 1123 (*STD* 3) розділ 5. Протокол який використовується зараз також відомий як *ESMTP* і визначений в *RFC* 2821. *SMTP* — порівняно простий, текстовий протокол, в якому з'єднання відбувається завжди за ініціативи відправника. *SMTP* — синхронний протокол і складається із серії команд, що посилаються клієнтом та відповідей сервера. Відправником зазвичай є поштовий клієнт кінцевого користувача або поштовий сервер. *SMTP* було розроблено як протокол транспортування і доставки, тому системи, що використовують *SMTP*, завжди повинні бути у робочому стані. Протокол часто використовується для передачі повідомлень клієнтами електронної пошти, які, проте, не мають можливості діяти як сервер.

Протоколи *POP* (*Post Office Protocol*) різних версій (*POP2* — *RFC*-937, POP3 —

RFC-1725) виконують аналогічні функції та використовують порти 109 і 110. Вони призначені для читання повідомлень з сервера для ПК. Ці протоколи працюють в *offline* режимі, тобто головні операції з поштою виконує локальний комп'ютер після її надходження з поштового сервера. Під час сеансу *POP*2 користувач послідовно одержує повідомлення про кількість наявних листів, їхній обсяг. Він може прочитати лист, а потім його знищити або ж зберегти на сервері.

Протокол *IMAP* (*RFC* 1064, *RFC* 1730) функційно наближений до *POP* і дає змогу користувачу працювати з віддаленим сервером електронної пошти. На відміну від *POP*, *IMAP* може працювати в режимі *online*. Це означає, що головні

операції над поштою виконуються безпосередньо на сервері. У цьому випадку забезпечене централізоване збереження поштової інформації та пов’язані з цим зручності оновлення, вірогідності, колективного використання, цілісності тощо. Протокол *MIME* відомий як розширення протоколу *SMTP*. Протокол *SMTP* орієнтований на передавання текстової інформації в коді *ASCII*, *MIME* ж призначено для пересилання даних двійкового формату з використанням *ASCII*-протоколів електронної пошти. Оскільки більшість *Internet*-систем електронної пошти не придатні для передавання довільного двійкового символу, то *MIME* доводиться перетворювати байт у текстовий символ.

Отже, у заголовку повідомлення *MIME* є деякі поля, які описують тип та підтип даних, а також методи кодування-декодування.

Система електронної пошти стандарту *UUCP*.

У системах протоколів *UUCP* зазначено всі проміжні хости, через які проходять повідомлення електронної пошти. Тому формат адреси, наприклад, такий: *serv1!serv2!serv3!user_dest* Повідомлення, яке не пройшло через якусь ланку ланцюжка, чекає на її доступність.

Система електронної пошти стандарту *X*.400

Система протоколів електронної пошти *X*.400 розроблена I*TU*, її

використовують у деяких випадках державні, військові та фінансові організації для гарантування надійного передавання. Порівняно з іншими системами (*SMTP*) ця система надає низку додаткових послуг, а саме:

— підтвердження доставляння та прочитання документа;

— підтримка мультимедійної інформації;

— пріоритетність повідомлень;

— захист, конфіденційність, шифрування;

— довільні формати повідомлень;

— взаємодія з іншими службами передавання повідомлень (пошта, факс, телекс).

Поштова система *X*.400 належить до систем з проміжним збереженням, вона не передає листи в режимі реального часу (як *SMTP*). У цьому вона подібна до *UUCP*. Недоліком *X*.400 є висока складність, значна вартість реалізації та адміністрування.

1.3 Аналіз існуючих програм для вирішення завдання

iRedMail – це:

1) Повноцінний поштовий сервер;
2) Працює під управлінням *Red Hat Enterprise Linux*, *CentOS*, *Debian*, *Ubuntu*, *FreeBSD*;
3) Працює як на звичній системі, так і під управлінням віртуальної машини;
4) Підтримка архітектур *i*386 та *x*86_64;
5) Використовує для встановлення і налаштування сумісні компоненти, такі як *Postfix*, *Dovecot*, *SpamAssassin* і т.д.;
6) При встановленні використовує бінарні пакети з поточного дистрибутиву;
7) Проект з відкритим вихідним кодом, який поширюється під ліцензією *GPL v*2;
8) Підтримка двох бекендів для зберігання віртуальних доменів і користувачів: *OpenLDAP* і *MySQL*;
9) Безлімітне число підтримки доменів, користувачів, поштових аліасів;
10) Підтримка двох *Web*-інтерфейсів (*RoundCube* і *SquirrelMail*).

Цей пакет дуже зручний в налаштуванні й встановленні.

1. Зручний цей пакет в тому, що:

a) Не потрібно багато знань, тобто процес встановлення і налаштування зовсім не складний;

b) Вимагає мінімум часу на розворот корпоративної пошти. Я витратив на встановлення цієї збірки буквально 20 хвилин;

c) У комплекті йде весь необхідний софт для поштового сервера (Антивірус, Антиспам, 2 *Web*-інтерфейси (за вибором)).

2. Аналогів даного пакета не існує.

Схема роботи *iRedMail*:

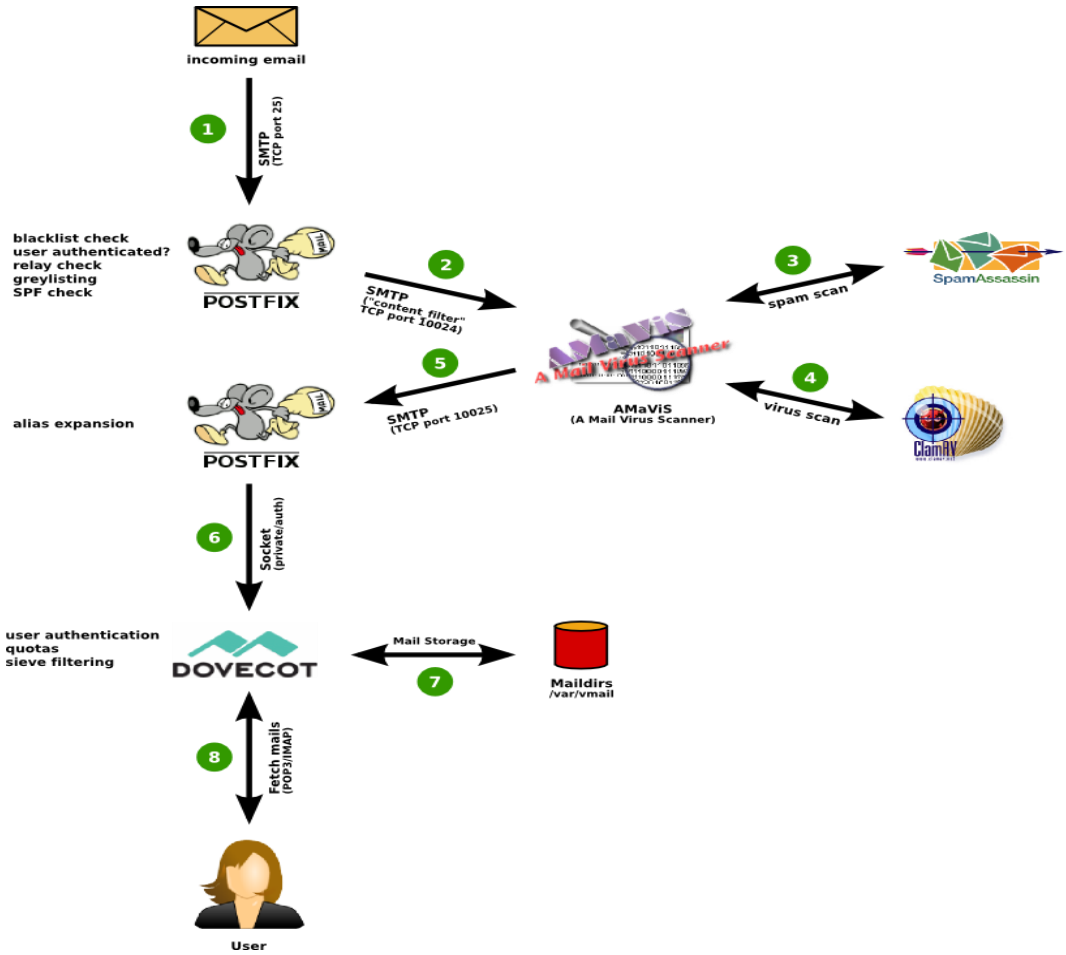

Рис 1.3.1 Схема роботи *iRedMail*

Що підтримує *iRedMail* для захисту пошти:

1. Підтримка *SPF* (*Sender Policy Framework*);
2. Підтримка *DKIM* (*DomainKeys Identified Mail*);
3. Підтримка *Greylist*;
4. Підтримка «Білих списків» (на підставі *DNS* імен та *IP* адреси);
5. Підтримка «Чорних списків» (на підставі *DNS* імен та *IP* адреси);
6. Підтримка «Чорного списку» *HELO* запитів;
7. Підтримка *HPR* (*HELO Randomization Prevention*);
8. Підтримка *Spamtrap*;
9. Інтеграція *SpamAssassin*;
10. Інтеграція *ClamAV*, автооновлення вірусних баз.

Підтримка поштових клієнтів:

Поштові клієнти з підтримкою *POP*3/*POP*3*S* і *IMAP*/*IMAPS*. Наприклад: *Mozilla Thunderbird*, *Microsoft Outlook*, *Sylpheed*.

Встановлення *iRedMail*.

В якості поштового сервера я вибрав *iRedMail*. Це збірка *Postfix+LDAP* (*MySQL*)*+SpamAssassin+ClamAV+AmaViS+Dovecot+RoundCube* (*SquirrelMail*) і т.д. У даній роботі ми будемо розглядати налаштування цього чудового сервера на базі *OpenLDAP*. Але це не є вирішенням нашого завдання, адже це готовий скрипт на розгортання поштового серверу, а нам потрібно власноруч його створювати.

Postfix – агент передачі пошти (*MTA – Mail Transfer Agent*). *Postfix* є вільним програмним забезпеченням, створювався як альтернатива *Sendmail*.

Спочатку *Postfix* був розроблений Вейтсом Венем в той час, коли він працював в Дослідницькому центрі імені Томаса Уотсона компанії *IBM*. Перші версії програми стали доступні в середині 1999 року.

Postfix відрізняється продуманою модульною архітектурою, яка дозволяє створити дуже надійну і швидку поштову систему. Так, наприклад, привілеї *root* потрібні тільки для відкриття порту (*TCP* 25 порт), а демони, які виконують основну роботу, можуть працювати звичайним користувачем в ізольованому (*chroot*) оточенні, що дуже позитивно позначається на безпеці.

Архітектура *Postfix* виконана в стилі *UNIX* – де прості програми виконують мінімальний набір функцій, але виконують їх швидко і надійно. Під час простою поштової системи непотрібні демони можуть припиняти свою роботу, вивільняючи тим самим пам’ять, а при необхідності знову запускатися *master*-демоном.

Також варто відзначити більш просту і зрозумілу конфігурацію в порівнянні з *SendMail* і меншу ресурсомісткість, особливо під час простою поштової системи.

Сумісний з *AIX, BSD, HP-UX, IRIX, GNU/Linux, Mac OS X, Solaris, Tru64 UNIX,* фактично може бути зібраний на будь-якій *Unix*-подібній операційній системі, що підтримує *POSIX* і має компілятор *C*. Є службою пересилання пошти за замовчуванням в ОС *NetBSD*.

Це наш вихід із ситуації, власноруч розгорнути поштовий сервер завдяки *Postfix*.

Висновок

У даному розділі було розглянуто основні функціональні можливості і різновиди поштових серверів.

Розглянувши всю інформацію про дане ПЗ, можна дійти висновку, що вона є одним з найкращих аналогів для вирішення поставленого завдання завдяки великої кількості переваг і майже відсутністю недоліків.

РОЗДІЛ 2
АЛГОРИТМ ТА ПОРЯДОК ВСТАНОВЛЕННЯ, КОНФІГУРАЦІЇ ПОШТОВОГО СЕРВЕРУ ТА ЙОГО ОБСЛУГОВУВАННЯ

2.1 Загальний аналіз *OS Debian*

Debian – комп'ютерна операційна система, основний дистрибутив якої складається тільки з вільного програмного забезпечення (main-секції архіву Debian). Популярний та впливовий дистрибутив GNU/Linux. Багатоцільова операційна система, що використовується: настільними комп'ютерами, лептопами, серверами, вбудованими системами. Існують проекти на основі ядер: *Debian GNU/Hurd, Debian GNU/kFreeBSD, Debian GNU/kNetBSD*.

Як доповнення, до послуги дозволяється встановлювати *deb*-пакунки, які не були включені до головного репозиторію, з причини надто обмеженої ліцензії або можливих законодавчих непорозумінь. А саме:

— додаткове (*Contrib*) – пакунки у цій частині розповсюджуються власником авторського права на умовах вільної ліцензії, але залежать від невільного ПЗ;

— невільне (*Non-Free*) – ліцензії пакетів у цій частині мають умови, що обмежують розповсюдження ПЗ.

2.2 Порядок встановлення поштового серверу *Postfix*

Інсталювання *Postfix*: *apt-get install postfix*

Процес встановлення дуже простий, під час нього вам всього лише доведеться відповісти на кілька запитань.

У першому випадку потрібно вказати загальний тип поштового налаштування. Якщо ви спочатку знаєте для яких цілей буде використовуватися ваш поштовий сервер, на питання ви зможете відповісти точно. Якщо ж кінцева конфігурація сервера поки невідома, залиште значення за замовчуванням або виберіть «Без налаштування». У будь-якому випадку, в подальшому ви зможете підігнати все під себе.

Другий раз діалогове вікно запросить *fqdn*-ім'я. Забігаючи наперед скажу, що тут найкраще буде вказати ім'я, на яке посилається *MX*-запис вашого домену (наприклад: для запису *mail.bissquit.com* найкраще вказати ім'я сервера – *mail*). План мінімум – не вказувати взагалі нічого, залишити запропоноване ім'я за замовчуванням (воно буде збігатися з ім'ям системи). Для особливо пильних і недовірливих читачів: немає, серйозно, ви можете залишити все як є, адже в основному файлі конфігурації *Postfix* ми все одно будемо вписувати необхідне нам ім'я, тобто крок із зазначенням імені при установці *Postfix* для нас в принципі зайвий. Чекаємо закінчення виконання всіх завдань по установці і плавно переходимо до невеликої початкового налаштовування нашого *MTA*.

Отже, щоб *Postfix* міг відправляти і приймати пошту, необхідно задати кілька налаштувань і в підсумку ми отримаємо мінімально працюючу конфігурацію агента пересилання. В інтернеті є безліч мінлива по налаштуванню, прочитавши які, пропадає будь-яке бажання возитися з *MTA*. На ділі ж для отримання працюючого агента потрібно зробити мінімум змін, більшість налаштувань визначені оптимальним чином за замовчуванням, при цьому зроблений непоганий упор на безпеку, чим, власне, і славиться *Postfix*.

/etc/postfix/main.cf Для початку зробимо бекап оригінального конфігураційного файлу (в даному випадку і коли б то не було рекомендую

починати правити конфіги завжди зі створення їх бекапа): *cp /etc/postfix/main.cf{,.orig}*

Відкриваємо конфіг: *nano /etc/postfix/main.cf*

Як я і говорилося вище, ми можемо визначити ім'я обслуговується поштовим сервером домена прямо в конфіги *Postfix*, керуючи параметрами *myhostname* і *mydomain*.

Задамо значення параметра *myhostname*: *myhostname = mail.mydomain.com*.

Знаючи *myhostname*, *Postfix* може отримати значення параметра *mydomain*, просто відкинувши все до першої точки. У моєму випадку залишиться просто *mydomain.com*. *Postfix* буде вважати, що саме цей домен він обслуговує. *mydomain = mydomain.com*.

Знаючи *mydomain*, *Postfix* може отримати *myhostname*, просто об'єднавши *mydomain* і ім'я хоста. Нічого страшного, якщо у вас будуть задані обидва параметри.

Тепер прийшов час визначити для яких доменів пошта буде доставлятися локальним одержувачам, а не пересилатися на третю сторону. Це як раз ті домени (в моєму прикладі всього один), які буде обслуговувати наш *MTA*. Це можна зробити за допомогою параметра *mydestination mydestination = $myhostname, $mydomain*.

$ myhostname, $ mydomain – змінні, значення яких ми визначили раніше.

Наступний параметр – *myorigin* – відповідає за додавання до адрес відправників/одержувачів імені домена. Думаю розумніше буде привласнити йому значення *$mydomain*.

myorigin = $mydomain

На цьому основна настройка завершена, можна закривати файл, зберігши зміни. Про всяк випадок перевіримо наявність необхідних адрес в параметрі *mydestination* командою *postconf mydestination*.

З налаштуванням */etc/postfix/main.cf* ми закінчили, йдемо далі.

/etc/aliases

З доставкою пошти користувачеві *root* виникнуть проблеми, а тому потрібно заздалегідь перенаправити всю вхідну для нього пошту на іншу обліковий запис. У моєму випадку це буде користувач *mailserveradmin*, під яким я і виконую всі дії.

Для цього відкриваємо файл *nano /etc/aliases.*

Додамо рядок *root*:

mailserveradmin

Для оновлення конфігурації *Postfix* без необхідності перезапуску демона введіть команду:

postfix reload

На цьому початкова настройка агента пересилання пошти *Postfix* завершена. Уже в цій конфігурації він може відправляти пошту будь-яким зовнішнім і локальним одержувачам, зрозуміло якщо у вас правильно створені всі необхідні публічні *DNS*-записи і вони встигли поширитися (зазвичай на це потрібно від пари десятків хвилин, до декількох годин або навіть діб, але це дуже рідкісний випадок.

У мінімальній конфігурації *Postfix* може відправляти і отримувати пошту по протоколу *SMTP*, використовує локальні облікові записи системи для зберігання повідомлень. Базова настройка, яку обов’язково потрібно виконати, має на меті визначення основного доменного імені поштового

сервера, яке в подальшому використовується в багатьох конфігураційних параметрах *MTA* як основа.

Базова конфігурація *Postfix* не може вважатися повноцінним поштовим сервером, адже відсутні безліч інших важливих компонентів. Проте завдання локальної відправки/прийому пошти по *SMTP* цей *MTA* виконує на відмінно.

Висновок

Проаналізувавши всю інформацію даного розділу починаючий адміністратор операційних систем зможе повністю зрозуміти порядок встановлення і налаштування поштового серверу, адже в даному розділі ми розглянули порядок встановлення поштового серверу, який в свою чергу має деякі особливості, яке не знає звичайний користувач чи адміністратор-новачок.

ВИСНОВОК

Виходячи з вищевказаної інформації можна дійти такого висновку, що скрипти для автоматизованого встановлення програмних компонент *OS* є необхідними для швидкого та правильного встановлення та налаштування необхідного ПЗ що забезпечує безпеку даних на ПК. Скрипт дозволяє виконувати цей процес непідготовленому персоналу, що значно збільшує доступність та простоту використання систем сімейства *UNIX*.

В результаті виконання курсового проекту було проаналізовано і розроблено скриптовий сценарій встановлення і налаштування серверної компоненти дистанційного навчання для серверної версії операційної системи відповідно до поставленого завдання.

Відповідно була досягнута головна мета курсової роботи, яка спрощує роботу, та швидкість розгортання поштового серверу в ЗСУ простими користувачами, які не мають спеціальних навичок в управлінні операційними системами та їх адмініструванням і здійснення базових налаштувань.

В кінцевому результаті, провівши аналіз, виходить, що скриптовий сценарій працює правильно, без помилок і повністю задовольняє умову вирішення завдання курсового проекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *UNIX and Linux System Administration Handbook / E.Nemeth, G. Snyder, T. Hein, B. Whaley*;

2. *Adelstein T. Linux System Administration / T. Adelstein, B. Lubanovic*;

3. Бэкон Д. Операционные системы / Д. Бэкон, Т. Харрис. – Питер, 2004. – 800 с. – (Издательская группа *BHV*);

4. Иртегов Д. Введение в операционные системы / Д.В. Иртегов. – Петербург, 2002. – 624 с;

5. *Postfix* – загальні відомості [Електронний ресурс] – Режим доступу до ресурсу: *https://ru.wikipedia.org/wiki/Postfix*;

6. Обзор і встановлення поштового сервера *iRedMail* [Електронний ресурс] – Режим доступу до ресурсу: *https://habr.com/ru/post/96314/*;

7. Настройка поштового сервера *iRedMail* на *CentOS* 7 [Електронний ресурс] – Режим доступу до ресурсу: *https://serveradmin.ru/ustanovka-i-nastroyka-iredmail/*;

8. Поштовий сервер *Postfix* [Електронний ресурс] – Режим доступу до ресурсу: *https://help.ubuntu.ru/wiki/postfix*;

9. Мінімальний поштовий сервер на основі *Postfix* і *Dovecot*. Частина 2: Postfix [Електронний ресурс] – Режим доступу до ресурсу: *habr.com/ru/post/258407/*.

ДОДАТОК А

Лістинг програмного коду

```
    import os
def domain():
    os.system('apt-get update')
    os.system('apt-get upgrade')
    os.system('apt-get install curl net-tools bash-completion wget lsof nano')
    os.system('echo order hosts,bind>letclhost.conf')
    os.system('echo multi on>>/etcl/host.conf')
    os.system('hostnamectl set-hostname mail.honorovich.com')
    os.system('echo 10.0.2.15 honorovich.com mail.honorovich.com>>/etc/hosts')
    input('\n Computer will reboot, start script again after reboot to install postfix.\n
Press ENTER to continue...\n')
    os.system('init 6')

    def postfix():
        os.system('apt install postfix')
        os.system('cp /etc/postfix/main.cf{,.backup}')
        os.system('cp -f main.cf /etc/postfix/main.cf')
        os.system('systemctl restart postfix')


        def dovecot():
            os.system('apt install dovecot-core dovecot-imapa')
```

```
        os.system('cp -f dovecit.conf /etc/dovecot/dovecot.conf')
        os.system('cp -f 10-auth.conf /etc/dovecot/conf.d/10-auth.conf')
        os.system('cp -f 10-mail.conf /etc/dovecot/conf.d/10-mail.conf')
        os.system('cp -f 10-master.conf /etc/dovecot/conf.d/10-master.conf')
        os.system('systemclt restart dovecot.service')
        os.system('echo export MAIL=$HOME/Maildir>>/etc/profile')
        print('Congratulations')

def webmail():
    os.system('apt install apache2.php7.0 libapache2-mod.php7.0 php7.0-curl php7.0-xml')
    os.system('cd /var/www/html')
    os.system('rm /var/www/html/index.html')
    os.system('cd /var/www/html && curl -sL https://repository.rainloop.net/installer.php\php')
    print('Congratulations')

    def adduser():
        username=input('Print user name please')
        os.system('adduser'+username)
        print('user:'+username+'@honorovich.com added.')

        def status():
            while True:
                try:
                    i=int(input('1-Install Postfix; 2-Dovecot installer; 0-exit'))
```

```
            if i==1:
                os.system('systemctl status postfix')
            elif i==2:
                    os.system('systemctl status doveecot')
            elif i==0:
                        break
            else:
                print('Please choose number again.')
        except Exception:
                print('There is no such option, please press 1,2 or 0.')


    while True:
        try:
            key=input(input('PostFix & WebMail installer.\n1 - Configure Domain. \n2-install & configure PostFix.\n3-install& configure dovecot.\n4-Install WebMail.\n5-Add account.\n6-Check status.\n0-Exit.'))


            if key==1:
                domain()
            elif key==2:
                postfix()
            elif key==3:
                dovecot()
            elif key==4:
                webmail()
            elif key==5:
```

```
            adduser()
        elif key==6:
            status()
        elif key==0:
            break
        else:
            print('\nThere is no such option.Try again.\nTip: choose number in menu.\n')

    except Exception:
        print('\nThere is no such option.Try again.\nOhh: choose number in menu.\n')
```