“Київський коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №6**

з дисципліни: «Операційні системи»

Тема: “Знайомство з базовими командами CLI-режиму в Linux для роботи з тектосвими файлами та написання найпростіших скриптових сценаріїв”

Виконали

студенти

групи КСМ-83а

Голубчиков В.О \_\_\_\_\_\_\_\_

Чегринець Б.В \_\_\_\_\_\_\_\_

Москаленко В.С \_\_\_\_\_\_\_\_

Ексузян Д.С \_\_\_\_\_\_\_\_

Перевірив викладач

Повхліб В.С. \_\_\_\_\_\_\_

Київ 2020

Робота студентів групи КСМ-83а Команда №1

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.

2. Знайомство з базовими діями при роботі з довідкою.

3. Знайомство з базовими діями при роботі з файлами та каталогами.

**Матеріальне забезпечення занять**

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows (Windows 7).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

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

**Завдання для попередньої підготовки.**

1. Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеличкий словник базових англійських термінів з питань призначення команд та їх параметрів.

|  |  |
| --- | --- |
| **Термін на англійській** | **Термін на українській** |
| Formatting features | Форматування особливості |
| Processing file | Файл обробки |
| Significant number | Значна кількість |
| Various ways | Різними способами |
| Displaying text | Відображення тексту |
| Command provides | Команда забезпечує |
| Both commands | Обидві команди |
| Redirection | Перенаправлення |
| Filter lines | Фільтрувальні лінії |
| Standard input | Стандартне введення |
| Shell for data | Оболонка для даних |
| User to redirect | Користувача для переспрямування |
| Directly searching | Безпосередній пошук |
| Particular phrase | Конкретна фраза |
| Pattern | Візерунок |

Робота студентів групи КСМ-83а Команда №1

1. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   1. Яке призначення команд cat, less, more, head and tail? Зробіть короткий опис кожної команди та виділіть їх основні параметри. Яким чином їх можна встановити.

**cat** - одна з найбільш часто використовуваних команд в Linux. Вона зчитує дані з файлів і виводить їх вміст. Це найпростіший спосіб відображення вмісту файлу в командному рядку.

**less** - програма для текстових терміналів UNIX-подібних систем, яка використовується для перегляду (але не зміни) вмісту текстових файлів на екрані. Відображає файл з можливістю прокрутки.

**more -** утиліта призначена для посторінкового перегляду файлів в терміналі Linux.

**head – команда, що дозволяє задіяти однойменну утиліту для виведення декількох перших рядків з текстового файлу або виведення іншого утиліти. Якщо ви виконуєте команду, яка виводить великий обсяг даних і при цьому зацікавлені лише в кількох перших рядках виведення, ця команда - саме те, що вам потрібно.**

**tail** – команда, що дозволяє користувачеві читати команди файлу з кінця. Також за допомогою цієї утиліти можна відслідковувати нову інформацію, оновлювану в реальному часі в конкретному файлі. Це дуже спрощує перевірку останніх записів в системі. Також команда tail в Linux дозволяє переглядати кілька файлів одночасно.

* 1. Поясніть принципи роботи командної оболонки з каналами, потоками та фільтрами.

Командна оболонка в [UNIX](https://uk.wikipedia.org/wiki/UNIX) є [інтерфейсом командного рядка](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%80%D1%8F%D0%B4%D0%BA%D0%B0) в [Unix](https://uk.wikipedia.org/wiki/Unix" \o "Unix)-подібних [операційних системах](https://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), тобто виконує команди, які подає користувач, або які читаються з [файлів](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB). Такі файли з командами оболонки називаються [сценаріями](https://uk.wikipedia.org/wiki/%D0%A1%D1%86%D0%B5%D0%BD%D0%B0%D1%80%D1%96%D0%B9" \o "Сценарій) (скриптами, програмами) оболонки. Ці сценарії не [компілюються](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%96%D0%BB%D1%8F%D1%82%D0%BE%D1%80" \o "Компілятор), а [інтерпретуються](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D1%80%D0%B5%D1%82%D0%B0%D1%82%D0%BE%D1%80" \o "Інтерпретатор) оболонкою. Це означає, що оболонка прочитує сценарій від початку до кінця, рядок за рядком, шукаючи зазначені там команди й виконуючи їх; на відміну від цього підходу, [компілятор](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%96%D0%BB%D1%8F%D1%82%D0%BE%D1%80" \o "Компілятор) перетворює цілу програму до вигляду, придатного до виконання машиною — потім файл з таким кодом можна використати в сценарії оболонки. Характерна особливість мови оболонки — багато операцій, які в традиційних мовах програмування є вбудованими, виконуються з допомогою виклику зовнішніх програм.

* 1. Яке призначення команди grep?

grep - утиліта командного рядка, яка знаходить на воді цілі терміни, що відповідають заданому регулярному виразу, і виводить їх, якщо висновок НЕ скасований спеціальними ключем.

Робота студентів групи КСМ-83а Команда №1

1. На базі розглянутого матеріалу у онлайн курсах дайте відповіді на наступні питання:
   1. Охарактеризуйте поняття скриптового сценарію у командній оболонці.
   2. Яким чином створюються та редагуються скрипти, що треба зробити щоб запустити скрипт?

**Хід роботи.**

1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
   1. Запустіть свою операційну систему сімейства Linux (якщо працюєте на власному ПК та її встановили) та запустіть термінал.
2. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу NDG Linux Essentials:

* Lab 10: Working With Text
* Lab 11: Basic Scripting
* Lab 12: Understanding Computer Hardware

1. Створіть таблицю команд вивчених у п.2 ходу роботи у наступному вигляді:

|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| vi myfile | Створює новий файл |
| dw | Видаляє слово “very” |
| u | Скасує останню операцію |
| 2dw | Видаляє два слова |
| хххх | Видаляє по одному чотири символи |
| 4u | Скасує останні 4 операції |
| 14x | Видаляє 14 символів |
| 5X | Видаляє 5 символів |
| dd | Видаляє поточний рядок |
| p | Всталяє видалені рядки під поточним рядком |
| 2u | Скасує дві останні операції |
| 2dd | Видаліть два рядки |
| 4w  D | Видаляє із поточної позиції до кінця рядка **Shift + D** . |
| J | Об’єднує два рядки, поточний і наступний, набравши велику літеру J( **Shift + J** ): |
| yw | Скопіюйє поточне слово |
| :%s/text //g | Шукайте та видаляйте слово text |
| : w | Зберігає файл. |
| vi myfile | Ще раз відкрийте myfile за допомогою vi редактора: |
| : q! | Закриває viредактор, не зберігаючи зміни: |
| /line | Шукайте слово вперед line |
| n | Шукайте наступний екземпляр слова line, натискаючи літеру n: |
| bash sample.sh | дин із способів запустити цю програму - це ввести bashдо імені файлу. |
| ls -l sample.sh  chmod a + x sample.sh  ls -l sample.sh ./sample.sh | Ви можете уникнути необхідності вводити bash перед іменем файлу, зробивши файл "виконуваним" для всіх користувачів. |
| sample.sh | Ви використовували ./перед sample.shіменем файлу, щоб вказати, що файл знаходиться в поточному каталозі. |
| echo $ PATH | Нагадаємо, що $PATH змінна використовується для пошуку команд, які ви вводите. |
| mkdir bin  mv sample.sh bin  sample.sh | Зверніть увагу, що /home/sysadmin/binце один із каталогів $PATHзмінної. |
| vi drive.sh | Скрипти, які є більш складними, можуть використовувати умовне виконання. Умовний вираз, як і **if** оператор, може використовувати результат викликаної команди **test**. |
| if [ $age -lt 16 ] | Тестовий оператор викликається автоматично, коли ви розміщуєте його аргументи у квадратних дужках, [ ]оточених пробілами. Змініть ifрядок drive.shтак, щоб він виглядав наступним чином: |
| vi num.sh | Ще одне поширене умовне твердження називається while циклом. |
| find ~ -name "\*bash\*" | Команда почне пошук в каталозі , вказаному і рекурсивно шукати всі підкаталоги. Наприклад, для пошуку файлів, що починаються з домашнього каталогу, що містять ім’я bash: |
| find / etc -name hosts 2> err.txt cat err.txt | Щоб перенаправити stderr(повідомлення про помилки) у файл, виконайте таку команду: |
| head -n 20 /proc/cpuinfo | Надає можливість відображення інформації про процесор може бути важливою при спробі визначити, чи можна використовувати у вашій системі більш розширені функції Linux. |
| free -m  free -g | Щоб дізнатися, скільки використовується оперативної пам’яті та місця підкачки, скористайтеся безкоштовною командою: |
| lspci | Щоб побачити, які пристрої підключені до шини PCI, використовуйте lspciкоманду: |
| lsusb | Спроба перерахувати підключені пристрої USB: |
| lsmod | Апаратне забезпечення функціонувало, ядро ​​Linux зазвичай завантажує драйвер або модуль. Використовуйте lsmod команду для перегляду завантажених модулів: |
| fdisk -l | Виконайте fdisk команду для переліку дискових пристроїв. В -l опції перераховані таблиці розділів для зазначених пристроїв , а потім виходить. Якщо жодних пристроїв не вказано, використовуються згадані в /proc/partitions |

1. Створіть скриптові сценарії з виводом текстових повідомлень для користувача:

* сценарій має виводити привітання до поточного користувача вказуючи поточну дату та інформацію про поточну систему;
* сценарій має виводити інформацію про апаратну конфігурацію поточної системи (використовуйте команди розглянуті в Lab 12).

**Контрольні запитання:**

1. Яким чином в командному інтерпретаторі можна перенаправляти потоки? Продемонструйте приклади, коли перенаправляється ввід / вивід / повідомлення про помилки.

**Командний інтерпретатор** — програма, яка забезпечує взаємодію користувача з операційною системою. Командний інтерпретатор звичайно підтримує командний рядок, змінні оточення, історію виконаних команд, власні конфігураційні файли.

Команди і сценарії можуть отримувати вхідні дані двома способами: з стандартного вхідного потоку (пов'язаний з клавіатурою) або з файлу. Аналогичний поділ існує і при виведенні даних: результати роботи команди або сценарію за замовчуванням направляються на екран терміналу, але ожна перенаправляти їх у файл. Якщо в процесі роботи виникають помилки. повідомлення про них може відображаються на екрані, потік помилок также можна переспрямувати у файл.

Розглянемо спочатку кілька команд, з допомогою яких можна організувати ввід/вивід.

1. Для чого використовуються команди фільтри. Наведіть декілька прикладних задач де їх використання є необхідним.

Команди фільтри використовують для фільтрування за певними параметрами. Прикладом фільтрування є табличка системних процесів.

1. Яке призначення директорії файлу /dev/null?

/ dev / null - спеціальний файл в системах класу UNIX, що представляє собою так зване «пусте пристрій». Запис в нього відбувається успішно, незалежно від обсягу «записаної» інформації. Читання з / dev / null еквівалентно зчитування кінця файлу (EOF).

В DOS є псевдофайла NUL з аналогічними властивостями.

**Висновок:**

В ході лабораторної роботи ми отримали практичні навички роботи з командною оболонкою Bash. Ознайомились з базовими діями при роботі з довідкою, файлами та каталогами.

Із труднощами не зіткнулись.