Klokun

Відповідно до індивідуального варіанту, охарактеризуйте чотири команди командного рядка Linux і наведіть приклади їх використання. Охарактеризувати словами, що вона робить, навести приклади використання.

`kill`

Використання:

```
kill [-s <signal> | -p] [--] pid ...
kill -l [signal]
```

Команда `kill` надсилає сигнал `signal` процесу з ідентифікатором процесу або назвою `pid`. Якщо сигнал не заданий, посилається сигнал `TERM`.

Команда `kill` найчастіше використовується для завершення процесів, оскільки сигнал `TERM` означає вимогу акуратно завершити процес. Деякі процеси не реагують або перехоплюють сигнал `TERM`, тому для їх завершення необхідно використати сигнал `KILL` (числовий код 9)

`-s` — задає сигнал, який треба надіслати. Сигнал може бути заданий як у числовій, так і у символьній формі.

`-p` — не надсилати сигнал, а лише вивести pid процесів, заданих назвою. (Необов'язкове розширення).

`-l` — вивести список усіх сигналів і їх імен.

Щоб надіслати сигнал `TERM` процесу з ідентифікатором `5512`, достатньо запустити `kill` так:

```
$ kill 5512
```

Щоб надіслати сигнал `KILL` тому ж процесу:

```
$ kill -s 9 5512
```

Щоб надіслати сигнал `TERM` процесу з назвою chrome:

```
$ kill chrome
```

locale

Використання:

```
locale [-am]
locale [-ck] NAME
```

Команда `locale` виводить дані про локаль, яка зараз використовується. Щоб вивести назви всіх локалей:

```
$ locale -a
```

Щоб вивести назви всіх символьних таблиць:

```
$ locale -m
```

Щоб вивести назви обраних категорій використовують параметр `-c`:

```
$ locale -c LC_NUMERIC
$ locale -c charmap
```

Щоб вивести назви та значення обраних ключових слів, використовують параметр `-k`. Наприклад:

```
$ locale -k LC_MONETARY
$ locale -k charmap
```

open

Використання:

```
open FILES...
```

Команда open відкриває файли `FILES...` програмою за замовчуванням з вікна терміналу.

Наприклад, щоб відкрити всі PNG-файли у поточній директорії з терміналу програмою за замовчування:

```
$ open *.png
```

umount

Використання:

```
umount [OPTION] [<posixpath>]
```

Команда umount відмонтовує підключені файлові системи.

Відмонтувати файлову систему «силою» (у випадку недосяжних мережевих файлових систем):

```
$ umount -f
```

Відмонтувати систему зараз, а прибрати посилання на задану файлову систему лише після того, як вона звільниться:

```
$ umount -l
```

Відмонтувати систему, не виводячи повідомлень, що вона не була змонтована.

```
$ umount -q
```

o → Охарактеризуйте функцію High Availability в програмних засобах VMware ESXi, VMware vSphere.

High Availability — це функція в програмному засобі VMware vSphere, яка надає можливість надлишкового захисту від апаратних та програмних проблем в рамках віртуалізованого середовища. High Availability дозволяє:

1. Відстежувати VMware vSphere хости та віртуальні машини, щоб виявляти неполадки гостьової ОС та апаратні неполадки.
2. Перезавантажувати віртуальні машини на інших хостах VMware vSphere в межах кластеру без ручного втручання, коли була виявлена непрацездатність сервера.
3. Зменшити неробочий час завдяки автоматичному перезавантаженню віртуальних машин у разі виявлення непрацездатності операційної системи.

Оберіть один з дистрибутивів, перша літера назви якого співпадає з четвертою літерою прізвища. Охарактеризуйте його.

K — Kali Linux — це дистрибутив Linux, заточений для задач тестування інформаційної безпеки програм, мереж та комп'ютерного обладнання.

Наприклад, в стандартний комплект поставки включені такі утиліти:

1. aircrack-ng — пакет утиліт для перевірки безпеки Wi-Fi мереж та проникнення в них.
2. JohnTheRipper — утиліта для відновлення паролів за їх хешем, яка використовує відеокарту для прискорення пошуку.
3. Metasploit — набір інструментів для запуску експлоїтів у різноманітних програмах.
4. Nmap — сканер відкритих портів.
5. Wireshark — аналізатор мережевих пакетів.

Kali Linux побудований на основі Debian і є нащадком дистрибутиву BackTrack, він розроблюється, підтримується і фінансується компанією Offensive Security. Менеджер пакетів — dpkg, метод оновлення — APT з декількома доступними фронт-ендами (synaptic, aptitude, KPackage тощо).

Напишіть скрипт

Можливості:

1. Розкриття та підстановка (результату обчислення арифметичних виразів, підстановка змінних, розкриття тільди, тощо)
2. Використання шаблонів та регулярних виразів.
3. Оператори `test` та `select`/`case`.

Написати коментарі.

```
#!/bin/bash
# Частина скрипту для отримання та валідації чисельного параметра

# Оголосити змінну — код завершення для випадку, коли заданий параметр не є
# числом
E_WRONGARGS=85
```

```
if [ "$1" == "set-lines" ]
then
    # Оператор вибору: до якого випадку належить перший параметр, з яким був зап
    case "$1" in
    # Якщо параметр пустий, встановити значення змінної lines = 50
    ""      ) lines=50;;
    # Якщо параметр містить будь-який символ, відмінний від 0-9 (перевіряється
    # регулярним виразом), тобто не є числом, вивести повідомлення про помилку
    # та закінчити роботу з кодом помилки зі змінної E_WRONGARGS
    *[!0-9]*) echo "Usage: `basename $0` set-lines <lines-to-cleanup>";
     exit $E_WRONGARGS;;
    # В інших випадках встановити значення змінної lines рівним заданому парамет
    *       ) lines=$1;;
    esac # Закінчити оператор вибору
else
    echo "Usage: `basename $0` set-lines <lines-to-cleanup>"
fi

echo $lines
```