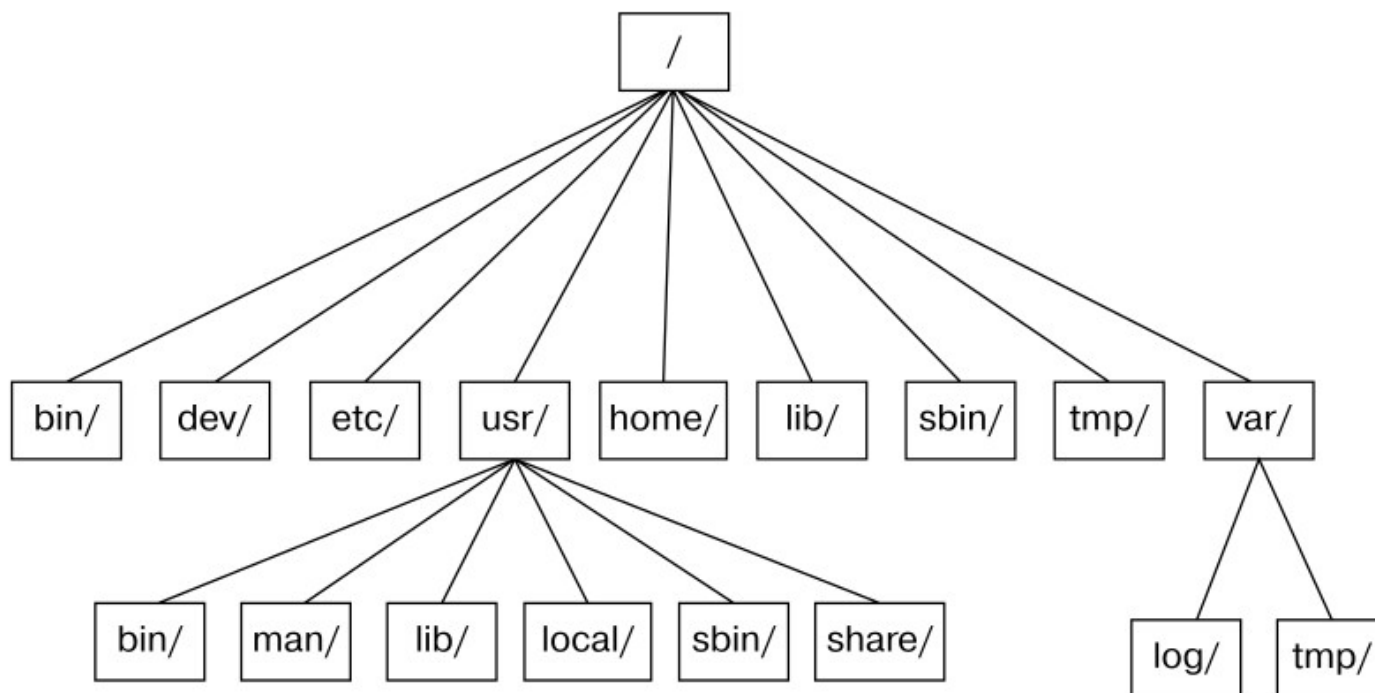


**Адміністрування комп'ютерних
систем**
**Модуль 2. Адміністрування ОС
Linux**
Лекція 5. Основи роботи в Linux

Кудін Олексій Володимирович
avk256@gmail.com

- Gentoo; ArchLinux; Linux From Scratch; Debian; Fedora; OpenSuse; Ubuntu; KaliLinux; Red Hat Enterprise Linux; Android..

- Типова структура каталогів:



uname – інформація про операційну систему;

uptime, vmstat – стан системи;

```
(base) avk256@avk256-G3:~$ uptime
09:35:33 up 1:14, 1 user, load average: 0,95, 0,74, 0,75
(base) avk256@avk256-G3:~$ vmstat
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0     0 1073980 142156 2839036    0    0   116   51  439 1038 10  4 86  0  0
```

adduser, passwd, userdel, groupadd, usermod – команди створення та керування обліковими записами користувачів;

/etc/passwd; /etc/group; /etc/shadow – файли з інформацією про користувачів.

```
(base) avk256@avk256-G3:/$ cat ./etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

grep – ВИВОДИТЬ СТРОКИ, ЯКІ ВІДПОВІДАЮТЬ деякому шаблону;

```
(base) avk256@avk256-G3:/$ grep root ./etc/passwd  
root:x:0:0:root:/root:/bin/bash
```

```
(base) avk256@avk256-G3:/$ grep root ./etc/*  
grep: ./etc/acpi: Is a directory  
grep: ./etc/alternatives: Is a directory  
./etc/anacrontab:HOME=/root  
./etc/anacrontab:LOGNAME=root  
grep: ./etc/anydesk: Is a directory  
grep: ./etc/apm: Is a directory
```

<code>pwd</code>	Вивести поточну директорію
<code>ls</code>	Вивести список файлів і каталогів
<code>ls -lax</code>	Вивести форматований список файлів і каталогів у тому числі і скриті
<code>cd</code>	Перехід у домашню директорію
<code>cd /home</code>	Перехід у директорію <code>/home</code>
<code>touch /home/example</code>	Створення порожнього файлу <code>/home/example</code>
<code>cat /home/example</code>	Показати вміст файлу <code>/home/example</code>
<code>tail /etc/shells</code>	Вивести кінець файлу <code>/etc/shells</code> . Це зручно для перегляду великих файлів
<code>nano /home/example</code>	Редагування файлу <code>/home/example</code> у текстовому редакторі <code>nano</code>
<code>gedit /home/example</code>	Редагування файлу <code>/home/example</code> у текстовому редакторі <code>gedit</code>
<code>cp /home/user1/example /home/user2/example</code>	Копіювання файлу <code>/home/user1/example</code> в <code>/home/user2/example</code>
<code>mkdir /home/user1/my_labs</code>	Створення директорії <code>my_labs</code>
<code>rmdir /home/user1/my_labs</code>	Видалення директорії <code>my_labs</code>
<code>cp -la /dir1 /dir2</code>	Копіювання директорій
<code>mv /dir1 /dir2</code>	Перейменування директорій

Команди термінала

<code>rpm -qa</code>	Вивести список встановлених RPM пакетів
<code>dpkg -l</code>	Вивести список встановлених DEB пакетів
<code>apt-cache search ім'я пакета</code>	Пошук доступних пакетів та виводить коротку інформацію про пакет
<code>apt-cache showpkg ім'я пакета</code>	Вивести повну інформацію про пакет
<code>sudo rpm -i pkgname.rpm</code>	Встановити RPM пакет <code>pkgname.rpm</code>
<code>sudo rpm -e pkgname</code>	Видалити RPM пакет <code>pkgname.rpm</code>
<code>sudo dpkg -i *.rpm</code>	Встановлення всіх RPM пакетів в директорії
<code>sudo apt-get update</code>	Оновлення списку доступних DEB пакетів
<code>sudo apt-get upgrade</code>	Оновлення доступної версії встановлених у системі пакетів
<code>sudo apt-get install pkgname</code>	Встановлення DEB пакета <code>pkgname</code>
<code>sudo apt-get remove pkgname</code>	Видалення DEB пакета <code>pkgname</code>
<code>sudo dpkg -i *.deb</code>	Встановлення всіх DEB пакетів в директорії
<code>top</code>	Вивести інформацію про завантажені процеси та використану оперативну пам'ять
<code>htop</code>	Розширена версія команди <code>top</code>
<code>dmesg</code>	Вивести log-файл завантаження ОС і знаходження нових пристроїв
<code>mpstat</code>	Вивести розширену статистику використання ресурсів системи (потребує встановлення пакета <code>sysstat</code>)
<code>vmstat</code>	Вивести статистику використання віртуальної пам'яті

<code>lsdev</code>	Вивести інформацію про встановлені пристрої (потребує встановлення пакета <code>procinfo</code>)
<code>cat /proc/cpuinfo</code>	Вивести повну інформацію про модель процесора
<code>cat /proc/meminfo</code>	Вивести розширену інформацію про використання оперативної пам'яті
<code>grep swaptotal /proc/meminfo</code>	Вивести інформацію про розмір розділу, який виділено під <code>swap</code>
<code>free -m</code>	Інформація про оперативну пам'ять і <code>swap</code> -розділ
<code>lshal</code>	Вивести список всіх пристроїв і їх параметри
<code>cat /proc/devices</code>	Вивести список всіх пристроїв і їх параметри
<code>lsusb -tv</code>	Вивести інформацію про знайдені <code>usb</code> -пристрої
<code>sudo dmidecode</code>	Вивести інформацію про версію встановлену версію BIOS

Процес – це програма яка виконується та має унікальний ідентифікатор (PID).

`ps` – отримати таблицю процесів.

`ps x` – всі процеси, які запущено користувачем

`ps ax` – всі процеси системи

`ps u` – деталізована інформація про процеси

`ps w` – повні назви команд

`&` - створення фонового процесу

```
(base) avk256@avk256-G3:/$ ps ux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
avk256    1215  0.0  0.1  77032  8120 ?        Ss   08:21   0:00 /lib/systemd/systemd --user
avk256    1216  0.0  0.0  279880  2840 ?        S    08:21   0:00 (sd-pam)
avk256    1231  0.4  2.2 1474088 177572 ?        Ssl  08:21   0:28 /home/avk256/.local/share/kite/current/kited --system-boot
avk256    1234  0.0  0.0  282884  7908 ?        Sl   08:21   0:00 /usr/bin/gnome-keyring-daemon --daemonize --login
avk256    1237  0.0  0.2  405316  22384 ?        Ssl  08:21   0:00 mate-session
```

Для виконання команд використовується командний інтерпретатор, який дозволяє розробляти сценарії (скрипти) для автоматизації типових дій в операційній системі.

nano /etc/shells

Приклад 1. Файл hello – сценарій виводу строки «Hello, world» на екран

```
1 #!/bin/bash
2 echo "Hello, world!"
```

chmod +x hello

./hello

В Linux та Windows часто використовуються змінні оточення. Це спеціальні змінні які містять деяку системну інформацію.

Приклад 2. Встановлення змінної оточення.

```
1 # Присвоєння значення змінній ADDRESS
2 $ADDRESS= www.znu.edu.ua
3 # Експорт змінної –
4 # після цього змінна ADDRESS
5 # доступна в інших сценаріях
6 export $ADDRESS
```

Приклад 3. Обробка параметрів сценарія.

```
1 # Сценарій повинен викликатися так:
2 # ім'я_сценарія параметр
3 # Аналізується перший параметр
4 case "$1" in
5 start)
6     # Дії при отриманні параметра start
7     echo "Enter \"start\""
8     ;;
9 stop)
10    # Дії при отриманні параметра stop
11    echo "Enter \"stop\""
12    ;;
13 *)
14    # Дії в інших випадках
15    # Вивести підказку, як потрібно
16    # Використовувати сценарій, і
17    # завершити роботу сценарія
18    echo "Usage: $0 {start|stop }"
19    exit 1
20    ;;
21 esac
```

Приклад 4. Використання циклів

```
1 for n in 1 2 3;
2 do
3 echo $n;
4 done
5 #####
6 n=1
7 while [ $n -lt 4 ]
8 do
9     echo "$n "
10    n=$(( $n+1 ));
11 done
```

-lt	Менше
-gt	Більше
-le	Менше або дорівнює
-ge	Більше або дорівнює
-eq	Дорівнює
-e файл	Істина, якщо файл існує
-d каталог	Істина, якщо каталог існує
-x фай	Істина, якщо файл виконуваний

Резюме (що встигли за лекцію)

- Розглянули деякі особливості ядра Linux.
- Розглянули команди для роботи з терміналом.
- Розглянули деякі можливості командного інтерпретатора.