

Звіт

Лабораторна робота №6

Тема: “Робота з текстом в CLI-режимі Linux та
створення скриптових сценаріїв”

Попов Владислав
Комнацький Олександр
РПЗ-83Б
Викладач Повхліб В.С.
31 марта 2021 г.

Мета роботи:

1. Отримання практичних навиків роботи з командною оболонкою.
2. Знайомство з базовими діями при роботі з довідкою.
3. Знайомство з базовими діями при роботі з файлами та каталогами.

Матеріальне забезпечення заняття

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 мережової 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.
2. На базі розглянутого матеріалу дайте відповіді на наступні питання:
 - 2.1. Яке призначення команд cat, less, more, head and tail? Зробіть короткий опис кожної команди та виділіть їх основні параметри.

cat – утиліта UNIX-подібних систем, яка здійснює операцію конкатенації вмісту вхідних аргументів та виводить їх на стандартний вивід. Приклад об'єднання файлів file1,file2 у file3 :

```
cat file1 file2 > file3
```

less – консольна програма в UNIX-подібних системах для проглядання (але не зміни) вмісту текстових файлів на екрані. Відображає файл з можливістю прокрутки. less – поліпшення утиліти more.

`less [параметри] <назва_файлу>`

more - використовується для проглядання вмісту текстових файлів в операційних системах, заснованих на Unix. На відміну від *less*, *more* виводить вміст файла на екран окремими сторінками.

head - це програма, яка друкує перші рядки введення. За замовчуванням він надрукує перші 10 рядків, але ми можемо змінити це за допомогою аргументу командного рядка.

- *head* [-кількість рядків для друку] [шлях]

tail - це команда, яка друкує останні рядки введення. За замовчуванням він надрукує останні 10 рядків, але ми можемо змінити це за допомогою аргументу командного рядка.

- *tail* [-кількість рядків для друку] [шлях]

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

Командна оболонка в UNIX є інтерфейсом командного рядка в Unix-подібних операційних системах, тобто виконує команди, які подає користувач, або які читаються з файлів. Такі файли з командами оболонки називаються сценаріями (скриптами, програмами) оболонки. Ці сценарії не компілюються, а інтерпретуються 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 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.

2.3. Яке призначення команди grep?

grep – утиліта інтерфейсу командного рядка, яка знаходить на вводі рядки, що відповідають заданому регулярному виразу, і виводить їх.

Назва утиліти є послідовністю команд пошуку регулярних виразів у редакторі ed – g/re/p.

3. Вивчіть матеріали онлайн-курсів академії Cisco:

- NDG Linux Essentials (Chapter 10-12 all Topics)

4. На базі розглянутого матеріалу у онлайн курсах дайте відповіді на наступні питання:

4.1 Охарактеризуйте поняття скриптового сценарію у командній оболонці.

Скриптовий сценарій - це програма, яка автоматизує деяке завдання, яке без сценарію користувач робив би вручну, використовуючи інтерфейс програми. Мови таких скриптів спочатку орієнтувалися на використання як внутрішні керуючі мови у складних системах. Багато хто з них, проте, вийшли за межі сфери свого початкового застосування і використовуються нині в зовсім інших областях. Характерними особливостями даних мов є, по-перше, їх інтерпретованість (компіляція або неможлива, або небажана), по-друге, простий синтаксис, а по-третє, легка розширюваність. Таким чином, вони ідеально підходять для використання в часто змінюваних програмах, дуже невеликих програмах або у випадках, коли для виконання операторів мови витрачається час, незрівнянно більший за час їх синтаксичного розбору інтерпретатором.

4.2 Яким чином створюються та редагуються скрипти, що треба зробити щоб запустити скрипт?

Приклад створення скрипту:

```
#!/bin/bash  
echo "Hello world»
```

Щоб зробити файл виконуваним в linux виконайте:

```
chmod ugo+x файл_скрипту
```

Запуск скрипту:

```
./файл_скрипту
```

5. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:

- Chapter 10 Exam
- Chapter 11 Exam

6. Підготувати в електронному вигляді початковий варіант звіту:

- Титульний аркуш, тема та мета роботи
- Словник термінів
- Відповіді на п.2.1-2.3 та п.4.1-4.2 з завдань для попередньої підготовки

Хід роботи

1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:

- 1.1. Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її. Виконайте вхід в систему під користувачем: CentOS, пароль для входу: reverse (якщо виконуєте ЛР у 401 ауд.) та запустіть термінал.
- 1.2. Запустіть віртуальну машину Ubuntu_PC (якщо виконуєте завдання ЛР через академію netacad)
- 1.3. 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

```
echo "Hello World"  
echo "Hello World" > mymessage  
cat mymessage
```

```
sysadmin@localhost:~$ echo "Hello World"  
Hello World  
sysadmin@localhost:~$ echo "Hello World" > mymessage  
sysadmin@localhost:~$ cat mymessage  
Hello World  
sysadmin@localhost:~$
```

```
echo "Greetings" > mymessage  
cat mymessage
```

```
sysadmin@localhost:~$ echo "Greetings" > mymessage  
sysadmin@localhost:~$ cat mymessage  
Greetings  
sysadmin@localhost:~$
```

```
cat mymessage  
echo "How are you?" >> mymessage  
cat mymessage
```

```
sysadmin@localhost:~$ cat mymessage  
Greetings  
sysadmin@localhost:~$ echo "How are you?" >> mymessage  
sysadmin@localhost:~$ cat mymessage  
Greetings  
How are you?  
sysadmin@localhost:~$
```

```
find ~ -name "*bash"
```

```
sysadmin@localhost:~$ find ~ -name "*bash*"  
/home/sysadmin/.bash_logout  
/home/sysadmin/.bashrc  
sysadmin@localhost:~$
```

```
find /etc -name hosts 2> err.txt
```

```
cat err.txt
```

```
sysadmin@localhost:~$ find /etc -name hosts 2> err.txt  
/etc/hosts  
sysadmin@localhost:~$ cat err.txt  
find: '/etc/ssl/private': Permission denied  
sysadmin@localhost:~$
```

```
find /etc -name hosts > std.out 2> std.err
```

```
cat std.err
```

```
cat std.out
```

```
sysadmin@localhost:~$ find /etc -name hosts > std.out 2> std.err  
sysadmin@localhost:~$ cat std.err  
find: '/etc/ssl/private': Permission denied  
sysadmin@localhost:~$ cat std.out  
/etc/hosts
```

```
find /etc -name hosts > find.out 2>&1
```

```
cat find.out
```

```
sysadmin@localhost:~$ find /etc -name hosts > find.out 2>&1  
sysadmin@localhost:~$ cat find.out  
/etc/hosts  
find: '/etc/ssl/private': Permission denied  
sysadmin@localhost:~$
```

tr a-z A-Z

```
sysadmin@localhost:~$ tr a-z A-Z
this is interesting
THIS IS INTERESTING
how do I stop this?
HOW DO I STOP THIS?
```

tr A-Z a-z > myfile

cat myfile

```
sysadmin@localhost:~$ tr A-Z a-z > myfile
Wow, I SEE NOW
This WORKS!
sysadmin@localhost:~$ cat myfile
wow, i see now
this works!
sysadmin@localhost:~$
```

ls -l /etc | more

```
sysadmin@localhost:~$ ls -l /etc | more
total 372
-rw-r--r-- 1 root root 2981 Jan 28 2015 adduser.conf
-rw-r--r-- 1 root root 10 Jan 28 2015 adjtime
drwxr-xr-x 1 root root 900 Jan 29 2015 alternatives
drwxr-xr-x 1 root root 114 Jan 29 2015 apparmor.d
drwxr-xr-x 1 root root 168 Oct 1 2014 apt
-rw-r--r-- 1 root root 2076 Apr 3 2012 bash.bashrc
drwxr-xr-x 1 root root 72 Jan 28 2015 bash_completion.d
drwxr-sr-x 1 root bind 342 Jan 29 2015 bind
-rw-r--r-- 1 root root 356 Apr 19 2012 bindresvport.blacklist
-rw-r--r-- 1 root root 321 Mar 30 2012 blkid.conf
lrwxrwxrwx 1 root root 15 Jun 18 2014 blkid.tab -> /dev/.blkid.tab
drwxr-xr-x 1 root root 16 Jan 29 2015 ca-certificates
-rw-r--r-- 1 root root 7464 Jan 29 2015 ca-certificates.conf
drwxr-xr-x 1 root root 14 Jan 29 2015 calendar
drwxr-xr-x 1 root root 24 Jan 29 2015 cron.d
drwxr-xr-x 1 root root 134 Jan 29 2015 cron.daily
drwxr-xr-x 1 root root 24 Jan 29 2015 cron.hourly
drwxr-xr-x 1 root root 24 Jan 29 2015 cron.monthly
-rw-r--r-- 1 root root 2969 Mar 15 2012 debconf.conf
--More--
```

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

```
sysadmin@localhost:~$ cut -d: -f1 /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
libuuid
syslog
bind
sshd
operator
```

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

```
sysadmin@localhost:~$ cut -d: -f1 /etc/passwd | sort
backup
bin
bind
daemon
games
gnats
irc
libuuid
list
lp
mail
man
news
nobody
operator
proxy
root
sshd
sync
sys
```

```
cat /etc/passwd
```

```
sysadmin@localhost:~$ 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
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
```

```
head /etc/passwd
```

```
sysadmin@localhost:~$ head /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:/usr/sbin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin:/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin:/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin:/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin:/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin:/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin:/nologin
sysadmin@localhost:~$
```

```
tail /etc/passwd
```

```
sysadmin@localhost:~$ tail /etc/passwd
_apt:x:100:65534::/nonexistent:/usr/sbin:/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd/netif:/usr/
sbin:/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nol
ogin
uuidd:x:103:105::/run/uuid:/usr/sbin:/nologin
syslog:x:104:108::/home/syslog:/usr/sbin:/nologin
messagebus:x:105:109::/nonexistent:/usr/sbin:/nologin
bind:x:106:111::/var/cache/bind:/usr/sbin:/nologin
sshd:x:107:65534::/run/sshd:/usr/sbin:/nologin
operator:x:1000:37::/root:/bin/bash
sysadmin:x:1001:1001:System Administrator,,,:/home/sysadmin:/bin/bash
sysadmin@localhost:~$
```

```
ls /etc | tail -5
```

```
sysadmin@localhost:~$ ls /etc | tail -5
vim
vtrgb
wgetrc
xdg
zsh_command_not_found
sysadmin@localhost:~$ █
```

```
cd /etc
```

```
grep sshd passwd
```

```
sysadmin@localhost:~$ cd /etc
sysadmin@localhost:/etc$ grep sshd passwd
sshd:x:107:65534::/run/sshd:/usr/sbin:/nologin
sysadmin@localhost:/etc$ █
```

```
lscpu
```

```
sysadmin@localhost:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                16
On-line CPU(s) list:  0-15
Thread(s) per core:   1
Core(s) per socket:   2
Socket(s):             8
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 63
Model name:            Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz
Stepping:              2
CPU MHz:               2497.108
BogoMIPS:              4994.21
Virtualization:        VT-x
Hypervisor vendor:    VMware
Virtualization type:  full
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
```

```
head -n 20 /proc/cpuinfo
```

```
sysadmin@localhost:~$ head -n 20 /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 63
model name     : Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz
stepping        : 2
microcode      : 0x3d
cpu MHz         : 2497.108
cache size     : 30720 KB
physical id    : 0
siblings        : 2
core id         : 0
cpu cores      : 2
apicid          : 0
initial apicid : 0
fpu             : yes
fpu_exception   : yes
cpuid level    : 15
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_
tsc arch_perfmon nopl xtopology tsc_reliable nonstop_tsc cpuid pni pclmulqdq vmx
```

```
free -m
```

```
free -g
```

```
sysadmin@localhost:~$ free -m
              total        used        free      shared  buff/cache   available
Mem:       64422       18155      12647        20      33619       46728
Swap:        2047       1384        663
sysadmin@localhost:~$ free -g
              total        used        free      shared  buff/cache   available
Mem:          62         17         12          0         32         45
Swap:           1          1          0
sysadmin@localhost:~$
```

lspci

```
sysadmin@localhost:~$ lspci
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 1.0)
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI (rev 01)
00:11.0 PCI bridge: VMware PCI bridge (rev 02)
00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.1 PCI bridge: VMware PCI Express Root Port (rev 01)
```

lspci -k

```
sysadmin@localhost:~$ lspci -k
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)
    Subsystem: VMware Virtual Machine Chipset
    Kernel driver in use: agpgart-intel
lspci: Unable to load libkmod resources: error -12
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)
    Subsystem: VMware Virtual Machine Chipset
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)
    Subsystem: VMware Virtual Machine Chipset
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
    Subsystem: VMware Virtual Machine Chipset
    Kernel driver in use: ata_piix
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
    Subsystem: VMware Virtual Machine Chipset
00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 1.0)
    Subsystem: VMware Virtual Machine Communication Interface
    Kernel driver in use: vmw_vmci
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
    Subsystem: VMware SVGA II Adapter
    Kernel driver in use: vmwgfx
00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X Fusion-
```

lsmod

```
sysadmin@localhost:~$ lsmod
Module           Size  Used by
nf_log_ipv4      16384  0
nf_log_common    16384  1 nf_log_ipv4
xt_LOG           16384  0
xt_limit          16384  0
xt_TCPMS          16384  0
bluetooth        548864  0
ecdh_generic     24576  1 bluetooth
binfmt_misc      20480  1
af_packet_diag   16384  0
netlink_diag     16384  0
vsock_diag       16384  0
sctp_diag         16384  0
sctp              311296  997 sctp_diag
dccp_diag         16384  0
dccp              73728  1 dccp_diag
tcp_diag          16384  0
udp_diag          16384  0
raw_diag          16384  0
inet_diag         24576  5 tcp_diag,sctp_diag,raw_diag,udp_diag,dccp_diag
unix_diag         16384  0
xt_nat            16384  999
ufs               77824  0
```

fdisk -l

```
sysadmin@localhost:~$ fdisk -l
Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000571a2

      Device Boot   Start     End   Blocks  Id  System
/dev/sda1    *     2048 39845887 19921920  83  Linux
/dev/sda2        39847934 41940991 1046529   5  Extended
/dev/sda5        39847936 41940991 1046528  82 Linux swap / Solaris

sysadmin@localhost:~$
```

3. Створіть таблицю команд вивчених у п.2 ходу роботи у наступному

вигляді:

Назва команди	Її призначення та функціональність
echo "Hello World" echo "Hello World" > mymessage cat mymessage	Перша команда передає повідомлення (stdout) на термінал. Друга команда перенаправляє вихідні дані; замість надсилання вихідних даних на термінал, він надсилається у файл, який називається mymessage. Остання команда відображає вміст файла mymessage.
echo "Greetings" > mymessage cat mymessage	Коли ви використовуєте символ > для перенаправлення stdout, вміст файла спочатку знищується.
cat mymessage echo "How are you?" >> mymessage cat mymessage	Ви можете уникнути клобування файла, використовуючи >> замість >. За допомогою >> ви додаєте файл. Зверніть увагу, що за допомогою >> всі існуючі дані зберігаються, а нові дані додаються в кінці файла.

Назва команди	Її призначення та функціональність
find ~ -name "*bash*"	Команда find - це хороша команда, щоб продемонструвати, як працює stderr. Ця дуже гнучка команда дозволяє шукати безліч опцій, таких як ім'я файлу, розмір, дата, тип та дозвіл. Команда find розпочне пошук у вказаному каталозі та здійснить рекурсивний пошук у всіх підкаталогах.
find /etc -name hosts 2> err.txt cat err.txt	Щоб перенаправити stderr (повідомлення про помилки) у файл, виконайте таку команду
find /etc -name hosts > std.out 2> std.err cat std.err cat std.out	Ви також можете перенаправити stdout та stderr у два окремі файли
find /etc -name hosts > find.out 2>&1 cat find.out	Щоб перенаправити як стандартний вивід (stdout), так і стандартну помилку (stderr) в один файл, спочатку перенаправіть stdout у файл, а потім перенаправіть stderr на той самий файл, використовуючи позначення 2> & 1. Частина 2> & 1 команди означає "відправити stderr (канал 2) туди ж, куди прямує stdout (канал 1)".
tr a-z A-Z	Команда tr перекладає символи, але вона приймає лише дані з stdin, ніколи з імені файлу, заданого як аргумент. Це чудово, коли ви хочете зробити щось на зразок написання великих літер даних, які вводяться з клавіатури.
tr A-Z a-z > myfile	Команда tr приймає введення з клавіатури (stdin), переводить символи, а потім перенаправляє вихід на stdout. Щоб створити файл із всіх малих символів, виконайте наступне
cat myfile	Щоб підтвердити, що ви створили файл, виконайте таку команду
ls -l /etc more	Ця команда дозволяє отримати вихідні дані команди ls і надіслати їх у команду more, яка відображає по одній сторінці даних за раз
cut -d: -f1 /etc/passwd	Дозволяє витягти всі імена користувачів з бази даних з назвою /etc/passwd (файл, що містить інформацію про обліковий запис користувача).
cut -d: -f1 /etc/passwd sort	Дозволяє вивести команду cut і надіслати її в команду sort, щоб надати певний порядок на виході

Назва команди	Її призначення та функціональність
cat /etc/passwd	Дозволяє відобразити весь вміст файлу / etc / passwd.
head /etc/passwd	За допомогою команди head можна відобразити верхню частину файлу.
tail /etc/passwd	Відображає останні десять рядків файлу / etc / passwd
ls /etc tail -5	дозволяє перевести вихідні дані команди ls до команди tail, відображаючи останні п'ять імен файлів у каталозі / etc
cd /etc grep sshd passwd	Використання grep у найпростішій формі полягає у пошуку заданого рядка символів, наприклад sshd у файлі / etc / passwd. Команда grep надрукує весь рядок, що містить відповідність
vi myfile	Щоб створити новий файл, виконайте таку команду
lscpu	Дозволяє визначити тип процесора.
head -n 20 /proc/cpuinfo	дозволяє перерахувати перші 20 рядків файла cpuinfo.
free -m free -g	Дозволяє дізнатися, скільки використовується оперативної пам'яті та місця підкачки.
lspci	Дозволяє побачити, які пристрої підключенні до шини PCI.
lspci -k	Використовуйте команду lspci з опцією -k, щоб показати пристрої разом із драйвером ядра та використовуваними модулями
lsmod	Дозволяє переглянути завантажені модулі
fdisk -l	Виконайте команду fdisk, щоб перерахувати дискові пристрої. Параметр -l перелічує таблиці розділів для вказаних пристроїв, а потім виходить.

4. Створіть скриптові сценарії з виводом текстових повідомлень для

користувача:

- сценарій має виводити інформацію про апаратну конфігурацію поточної системи

-lscpu

-lspci

-lsmod
-lspci -k
-fdisk -l

Контрольні запитання

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

Утиліти використовують тільки стандартні потоки. Для таких програм оболонка дозволяє незалежно перенаправляти потоки введення/виводу. Наприклад, можна придушити вивід повідомень rcp помилки, встановити введення або виведення з файлу.

Окремий випадок використання механізму перенаправлення потоків - перенаправлення в /dev/null, що дозволяє позбутися непотрібних повідомень на екран.

Перенаправлення потоків введення-виведення здійснюється, подібно DOS (Точніше, синтаксис перенаправлення потоків ОС DOS сприйняла піт UNIX) за допомогою символів:

*> - перенаправлення стандартного потоку виводу
>> - перенаправлення стандартного потоку виводу у режимі записування
< - перенаправлення стандартного потоку вводу
<< - отримання дані зі стандартного потоку вводу до тих пір, поки не зустрінеться розділювач*

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

Команди-фільтри виводять рядки, відповідні заданим шаблоном. Вони читають рядки з файлу або стандартного вводу і виводять всі відповідні рядки за умовчанням на стандартний висновок.

Основная команда – *grep*, её варианты – это то же самое, что и использование определенных опций для *grep*, как показано в примере ниже:

egrep = *grep -E*

fgrep = *grep -F*

rgrep = *grep -r*

3. Яке призначення директорії файлу */dev/null*?

/dev/null - спеціальний файл в системах класу UNIX, що представляє собою так зване «пустепристрій». Запис в нього відбувається успішно, незалежно від обсягу «записаної» інформації. Читання з */dev/null* еквівалентно зчитування кінця файлу. Найчастіше перенаправлення в */dev/null* використовується для придушення стандартного виводу (вихідного потоку) і / або виведення повідомлень про помилки (потоку діагностики) програми їх перенаправленням в */dev/null*, таке придушення найчастіше використовується в командних сценаріях (*shell scripts*) для придушення небажаного виведення на консоль.

Висновок: виконали лабораторну роботу, отримали практичні навички роботи з командною оболонкою *Bash*, ознайомилися з базовими діями при роботі з довідкою, ознайомилися з базовими діями при роботі з файлами та каталогами.