

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет
телекоммуникаций и информатики»

Лабораторная работа №1
«Часть 1 – Работа с командной строкой»

Выполнил: студент 4 курса

ИВТ, гр. ИП-713

Михеев Н.А.

Проверил: старший преподаватель кафедры ПМиК

Милешко А.В.

Новосибирск, 2020 г.

1. Определить тип файлов /dev/hd0, /dev/console, /dev/tty0, /dev/shmem, /dev/mem.

Для выполнения буду пользоваться командой `file -b <путь к файлу>`, которая отображает тип файла. Флаг `-b` (briefly) показывает только тип файла, можно передать несколько файлов сразу, что я и сделал, запустив команду:

```
# file -b /dev/hd0 /dev/console /dev/tty0 /dev/shmem /dev/mem
```



```
# file -b /dev/hd0 /dev/console /dev/tty0 /dev/shmem /dev/mem
block special (4/0)
character special (0/1)
character special (7/0)
directory
data
# _
```

/dev/hd0 – блочный файл, файл устройства, который предоставляет буферизированный доступ к оборудованию системы

/dev/console – символьный файл – файл устройств, который обеспечивает не буферизованный последовательный доступ к системным аппаратным компонентам.

/dev/tty0 – тоже символьный файл

/dev/shmem – директория

/dev/mem – файл с данными – содержит текст, данные или программные инструкции

2. Определить, какой каталог делается рабочим при входе в систему. Почему?

Чтобы определить рабочий каталог, воспользуемся командой `# pwd – print working directory` – вывести рабочую директорию.



```
data
# pwd
/root
# _
```

/root – рабочий каталог по умолчанию для суперпользователя. Является рабочим по умолчанию, потому что так было решено разработчиками ОС.

3. Создать каталог LAB1 и сделать его рабочим.

Для создания нового каталога воспользуемся командой `# mkdir <название нового каталога>`. Далее сделаем его рабочим с помощью команды `# cd <название каталога для перехода>`. Проверим результат работы с помощью `pwd`.

```
# mkdir LAB1
# cd LAB1
# pwd
/root/LAB1
#
```

4. Определить (с помощью программы ls), в каком каталоге содержится файл services. Посмотреть его содержимое.

Использование команды `ls` нецелесообразно, так как простой поиск с ее помощью не возможен, необходимо составлять команду с использованием рекурсивного обхода по всем директориям вместе с утилитой `grep`, где будет задан поиск по имени файла и вывод директории перед именем файла (пр. `# ls -R | grep -e services -e “:”`) и все равно вывод останется более чем нечитаемым. Проще воспользоваться стандартной утилитой `find` с параметрами: `“/ -name services”` – где `/` – начальный каталог поиска, `-name` – флаг для указания имени файла, `services` – название искомого файла.

```
ttyp0: sh
# find / -name services
/etc/services
/usr/qnx650/target/qnx6/etc/services
```

Чтобы просмотреть содержимое файла воспользуемся командой `# cat /etc/services`

```
ttyp0: more
#
# Network services, Internet style
#
#      @(#)services      5.8 (Berkeley) 5/9/91
#
echo      7/tcp
echo      7/udp
discard   9/tcp      sink null
discard   9/udp      sink null
sysstat    11/tcp     users
daytime    13/tcp
daytime    13/udp
netstat    15/tcp
qotd       17/tcp      quote
chargen    19/tcp      ttytst source
chargen    19/udp      ttytst source
ftp        21/tcp
telnet     23/tcp
smtp       25/tcp      mail
time       37/tcp      timeserver
time       37/udp      timeserver
rtp        39/udp      resource      # resource location
nameserver 42/tcp      name          # IEN 116
whois      43/tcp      nickname
domain     53/tcp      nameserver   # name-domain server
domain     53/udp      nameserver
ntp        57/tcp      # deprecated
#
# Bootp experimental (sellgren@vangogh)
bootp      67/udp      bootps       # bootp server
bootpc     68/udp      # bootp client
#
tftp       69/udp
rje        77/tcp      netrjs
finger     79/tcp
link       87/tcp      ttylink
supdup     95/tcp
hostnames  101/tcp      hostname     # usually from sri-nic
tsap       102/tcp      # part of ISODE.
#csnet-cs  105/?
--More--
```

5. Сколько скрытых файлов в вашем домашнем каталоге?

Для начала необходимо перейти в домашний каталог по пути /home/user. Далее введя команду ls с флагом -a, произойдет вывод всех файлов, в том числе и скрытых, начинающихся с символа точка

```
# ls -al
total 33
drwxr-xr-x  4 user      user      4096 Sep 17  2019 .
drwxrwxr-x  3 root      root      4096 Sep 04  2019 ..
-rw-rw-r--  1 user      user        0 Sep 20  2019 .lastlogin
drwx-----  3 user      user      4096 Sep 17  2019 .mozilla
drwxrwxr-x  7 user      user      4096 Sep 20  2019 .ph
-rw-r--r--  1 user      user      191 Sep 04  2019 .profile
#
```

В домашнем каталоге 4 скрытых файла

6. Определить полное дерево подкаталогов в /boot . Сколько там файлов, размер которых меньше 1К байт? Сколько там исполняемых файлов?

Для отображения дерева подкаталогов в директории /boot можно использовать команду find с параметром -type d, что отображает пути файлов типа директория.

```
# find /boot -type d
/boot
/boot/fs
/boot/build
/boot/sys
#
```

3 каталога

```
# find /boot -size -1k
/boot/sys/elf.boot
/boot/sys/ipl-diskpc1
/boot/sys/ipl-diskpc1-flop
/boot/sys/ipl-diskpc2
/boot/sys/ipl-diskpc2-flop
/boot/sys/nobios.boot
#
```

6 файлов меньше 1Кб, используя дополнительный флаг -size

```
# find /boot -perm -110 -type f
/boot/sys/bios.boot
/boot/sys/bios16m.boot
/boot/sys/bios_nokbd.boot
/boot/sys/ipl-diskpc1
/boot/sys/ipl-diskpc1-flop
/boot/sys/ipl-diskpc2
/boot/sys/ipl-diskpc2-flop
/boot/sys/startup-apic
/boot/sys/startup-bios
/boot/sys/startup-bios-32
#
```

10 исполняемых пользователем файлов, используя флаг -perm -110 – для файлов, исполнимых пользователем

7. Сколько жестких связей у каталога /boot и почему?

Для этого воспользуемся командой `# ls -l /`. -l полный вывод информации, второй параметр как раз показывает количество жестких связей, у каталога boot их 5.

```
drwxrwxr-x 5 root root 4096 Sep 04 2019 boot
```

8. Создать текстовый файл с помощью редактора vi. Какие флаги доступа устанавливаются у вновь создаваемого файла? Почему? Как это исправить?

Для создания воспользуемся командой `# vi test`. Сохранимся и выйдем. По команде `# ls -l` можно посмотреть флаги доступа у файла test.

```
# ls -l
total 17
drwxrwxr-x 2 root root 4096 Sep 13 17:35 .
drwxr-x--x 10 root root 4096 Sep 13 16:19 ..
-rw-rw-r-- 1 root root 1 Sep 13 17:35 test
```

По умолчанию установились флаги доступа:

- 1) Для пользователя (user) – чтение (r), запись(w).
- 2) Для группы (group) - чтение (r), запись(w).
- 3) Для других (other) – чтение(r).

Такие флаги доступа установились по умолчанию так, чтобы был доступ к записи файла только для процессов, созданных пользователем или группой.

Для изменения флагов доступа существует утилита `chmod`

9. Сделать каталог и создать в нем 10 копий некоторого файла. Перенести три из них в вышестоящий каталог. Удалить (с подтверждением) некоторые из оставшихся файлов. Проверить влияние флага w на команду удаления файла.

Для создания каталога – `mkdir testdir`, для создания файла – `touch file`. Для того чтобы скопировать файл 10 раз напишем shell скрипт:

```
# for i in 0 1 2 3 4 5 6 7 8 9; do cp file file_"$i"; done
```

Результат работы цикла:

```
# for i in 0 1 2 3 4 5 6 7 8 9; do cp file file_"$i"; done
# ls
.      file      file_1    file_3    file_5    file_7    file_9
..     file_0     file_2    file_4    file_6    file_8
#
```

Цикл для копирования 3 файлов в вышестоящий каталог:

```
# for i in 0 1 2; do cp file_"$i" ../; done
# ls
.          file      file_1    file_3    file_5    file_7    file_9
..         file_0    file_2    file_4    file_6    file_8
# ls ..
.          file_0    file_2    testdir
..         file_1    test
#
```

Скопированные файлы остались и были скопированы.

Цикл для удаления 2 файлов из папки.

```
# for i in 3 4; do rm -i file_"$i"; done
rm: remove file_3? (y/N) y
rm: remove file_4? (y/N) y
# ls
.          file      file_1    file_5    file_7    file_9
..         file_0    file_2    file_6    file_8
#
```

Файлы были удалены с подтверждением.

Состояние флага w не влияет на удаление файла. Что подтверждается скриншотом.

```
# chmod 777 file
# chmod 000 file
# rm -i file
rm: remove file? (y/N) y
# ls
.          file_0    file_2    file_6    file_8
..         file_1    file_5    file_7    file_9
#
```

10. Определить значения переменных среды PATH, LOGNAME, HOME, HOSTNAME, PWD, RANDOM. Меняются ли они со временем?

```
# echo $PATH
/sbin:/usr/sbin:/bin:/usr/bin:/usr/photon/bin:/usr/photon/appbuilder:/opt/X11R6/
bin:/usr/X11R6/bin:/usr/local/bin:/opt/bin:/opt/sbin:/usr/qnx650/host/qnx6/x86/u
sr/bin:/usr/qnx650/host/qnx6/x86/usr/sbin:/usr/qnx650/host/qnx6/x86/sbin:/usr/qn
x650/host/qnx6/x86/bin:/usr/qnx650/host/qnx6/x86/usr/photon/appbuilder
# echo $LOGNAME
root
# echo $HOME
/root
# echo $HOSTNAME
localhost
# echo $PWD
/root/LAB1/testdir
# echo $RANDOM
10217
#
```

\$PATH — это переменная среды, используемая для указания оболочке, где искать исполняемые файлы. Не меняется со временем, но поддается редактированию.

\$LOGNAME – содержит имя пользователя, так же не меняется со временем, но поддается редактированию.

\$HOME – домашний каталог, можно изменить, со временем не меняется.

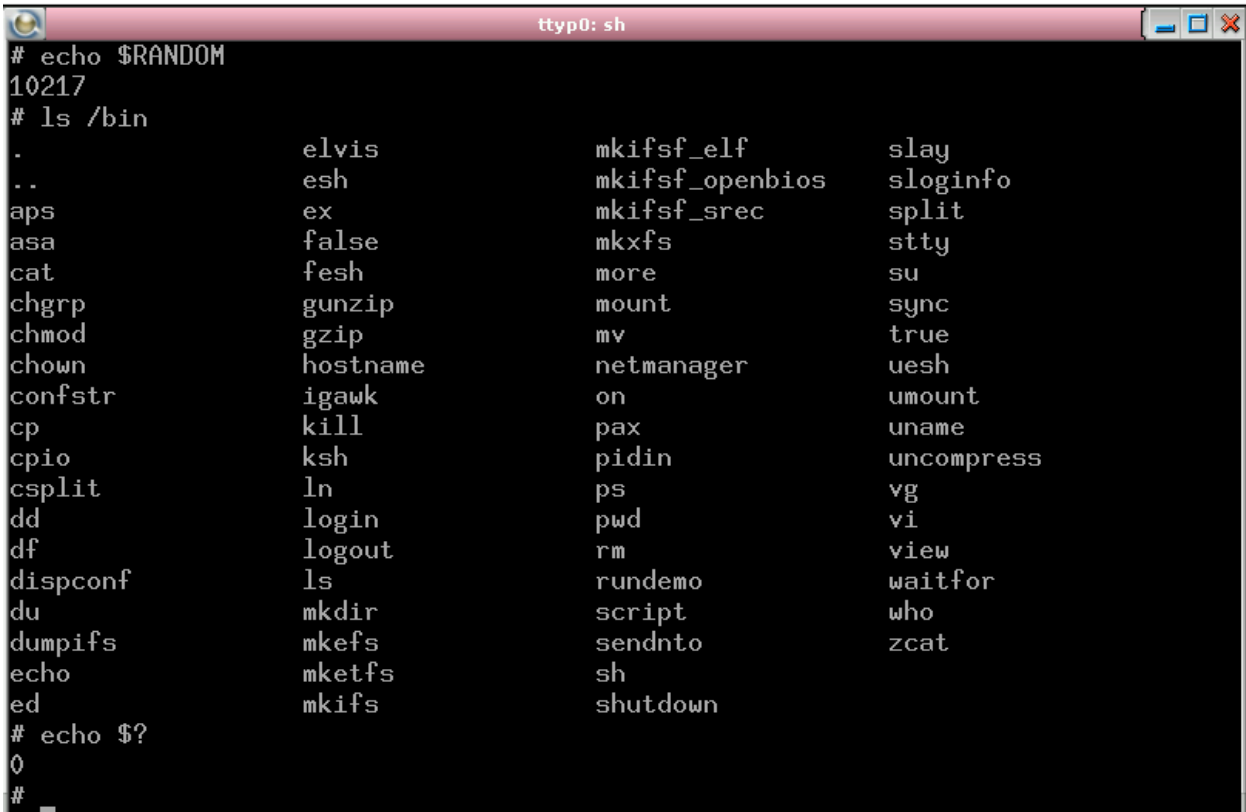
\$HOSTNAME – имя компьютера, не изменяется.

\$PWD – рабочий каталог, изменяется при изменении рабочего каталога.

\$RANDOM – случайное число, изменяется в зависимости от текущего времени.

11. Определить коды завершения команд `ls /bin` и `ls /pin`

Коды завершения команд можно посмотреть, введя `# echo $?` - после выполнения команды.



```
ttyp0: sh
# echo $RANDOM
10217
# ls /bin
.                elvis            mkifs_elf        slay
..               esh              mkifs_openbios  sloginfo
aps              ex               mkifs_srec       split
asa              false            mkxfs            stty
cat              fesh             more             su
chgrp            gunzip           mount            sync
chmod            gzip             mv               true
chown            hostname         netmanager       uesh
confstr          igawk            on               umount
cp               kill             pax              uname
cpio             ksh              pidin            uncompress
csplit           ln               ps               vg
dd               login            pwd              vi
df               logout           rm               view
disconf          ls               rundemo          waitfor
du               mkdir            script           who
dumpifs          mkefs            sendnto          zcat
echo             mketfs           sh
ed              mkifs            shutdown
# echo $?
0
#
```

Код завершения - 0 – т.к. команда успешно завершилась, выведя результат.

```
ttyp0: sh
# ls /pin
ls: No such file or directory (/pin)
# echo $?
1
#
```

Код завершения – 1. Т.к. работа команды завершилась ошибкой – нет каталога.

12. Вывести содержимое каталога /bin в файл в несколько колонок. Затем добавить к нему распечатку каталога /usr/bin.

Чтобы результат работы команды добавить в файл, необходимо после команды добавить символ «>» и ввести название файла.

```
# ls -C /bin > ls_res.txt
# cat ls_res.txt
.          elvis          mkifsf_elf    slay
..         esh          mkifsf_openbios sloginfo
aps        ex          mkifsf_srec   split
asa        false        mkxfs         stty
cat        fesh         more          su
chgrp      gunzip        mount         sync
chmod      gzip          mv            true
chown      hostname       netmanager   uesh
confstr    igawk          on            umount
cp         kill           pax           uname
cpio       ksh            pidin         uncompress
csplit     ln             ps            vg
dd         login         pwd           vi
df         logout        rm            view
disconf    ls             rundemo       waitfor
du         mkdir          script        who
dumpifs    mkefs          sendnto       zcat
echo       mketfs         sh
ed         mkifs          shutdown
#
```

Cat – проверка что файл успешно записался. Чтобы добавить таким же образом данные в уже существующий файл необходимо использовать управляющий символ «>>»


```
ttty0: sh
.      elvis      mkifsf_elf      slay
..     esh        mkifsf_openbios  sloginfo
aps    ex         mkifsf_srec    split
asa    false     mkxfs         stty
cat     fesh      more         su
chgrp  gunzip     mount        sync
chmod  gzip       mv          true
chown  hostname  netmanager  uesh
confstr igawk     on          umount
cp      kill      pax         uname
cpio    ksh       pidin       uncompress
csplit  ln        ps          vg
dd       login    pwd         vi
df       logout   rm          view
dispconf ls         rundemo     waitfor
du       mkdir    script      who
dumpifs mkefs       sendnto     zcat
echo     mketfs   sh
ed       mkifs    shutdown
.
..
alphalayer
arp
awk
--More--
.      nslookup
..     nsupdate
.      ntox86-ld
..     ntox86-ld-2.19
.      ntpdc
```

Красной линией я обозначил момент, с которого идет вставка результата второй команды ls. Как видно, то файл не перезаписался.

13. Сколько файлов удалили бы команды `rm /usr/bin/g*` и `rm /usr/bin/t??` ? (просьба файлы не удалять)

Для того чтобы узнать это, можно воспользоваться командой `ls -l <путь к каталогу>` совместно с утилитой `wc` с флагом `-l` для подсчета количества строк в результате работы команды `ls`.

```
# ls -l /usr/bin/g* |wc -l
7
# ls -l /usr/bin/t?? |wc -l
5
#
```

Сначала удалится 7 файлом, а затем 5.

14. Сколько всего пользователей зарегистрировано в системе?

Количество пользователей можно посмотреть в файле `/etc/passwd`, имеющем синтаксис:

имя_пользователя:пароль:ид:ид_группы:группа:домашний_каталог:оболочка

Пользователи с `ID < 100` – системные, они были созданы во время установки. Пользователь с `ID = 0` – суперпользователь.

```
# cat /etc/passwd
root::0:0:Superuser:/root:/bin/sh
bin:x:1:1:Binaries Commands and Source:/bin:
daemon:x:2:2:System Services:/daemon:
mail:x:8:40:User Mail:/var/spool/mail:
news:x:9:50:Network News:/var/spool/news:
uucp:x:12:60:Network News:/var/spool/news:
ftp:x:14:80:FTP User:/home/ftp:
sshd:x:15:6:sshd:/var/chroot/sshd:/bin/false
nobody:x:99:99:Nobody:/:
user::100:100:User:/home/user:/bin/sh
#
```

В данном случае пользователей 2 – root – суперпользователь и вручную зарегистрированный – user. Всего их 10.

15. Сколько различных групп пользователей в системе?

Группы пользователей хранятся в файле /etc/group и их 15.

```
# cat /etc/group
root:x:0:root
bin:x:1:root,bin
daemon:x:2:daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
tty:x:5:root
sshd:x:6:
mail:x:40:mail
news:x:50:news
uucp:x:60:uucp
ftp:x:80:ftp
guest:x:90:
nobody:x:99:
display:x:82:
user::100:
# cat /etc/group |wc -l
15
#
```

16. Определить имена пользователей, у которых нет пароля.

Из задания 14 можно узнать, что это «пользователи» с символом X на втором месте, а именно: bin, daemon, main, news, uucp, ftp, sshd, nobody – 8 «пользователей»

17. Защитить файл для чтения со стороны владельца, проверить.

По умолчанию установились флаги доступа: для пользователя (user) – чтение (r), запись(w); для группы (group) - чтение (r), запись(w); для других (other) – чтение(r).

```
# ls -l
total 25
drwxrwxr-x 3 root    root    4096 Sep 13 18:58 .
drwxr-x--x 10 root    root    4096 Sep 13 16:19 ..
-rw-rw-r-- 1 root    root      1 Sep 13 17:35 test
```

Как я понял, защитить файл для чтения со стороны владельца значит, что только владелец сможет читать файл и никто другой. Сначала очищу полностью все права доступа и для дальнейших заданий в том числе командой `# chmod 000 test`. Затем задам права чтения владельцу командой `# chmod u+r test`. Проверяем:

```
# chmod 000 test
# ls -l
total 25
drwxrwxr-x 3 root    root    4096 Sep 13 18:58 .
drwxr-x--x 10 root    root    4096 Sep 13 16:19 ..
----- 1 root    root      1 Sep 13 17:35 test
drwxrwxr-x 2 root    root    4096 Sep 13 18:39 testdir
# chmod u+r test
# ls -l
total 25
drwxrwxr-x 3 root    root    4096 Sep 13 18:58 .
drwxr-x--x 10 root    root    4096 Sep 13 16:19 ..
-r----- 1 root    root      1 Sep 13 17:35 test
drwxrwxr-x 2 root    root    4096 Sep 13 18:39 testdir
#
```

Как видно, только у владельца остались права чтения для файла test.

18. Защитить файл для чтения со стороны других пользователей, проверить.

Добавим остальным пользователям возможность чтения файлов командой `# chmod go+r test`, проверим

```
# chmod go+r test
# ls -l
total 25
drwxrwxr-x 3 root    root    4096 Sep 13 18:58 .
drwxr-x--x 10 root    root    4096 Sep 13 16:19 ..
-r--r--r-- 1 root    root      1 Sep 13 17:35 test
drwxrwxr-x 2 root    root    4096 Sep 13 18:39 testdir
#
```

Как видно все могут читать файл test.

19. Защитить файл для записи со стороны владельца, проверить.

Добавим владельцу права записи командой `# chmod u+w test`, проверяем

```
# chmod u+w test
# ls -l
total 25
drwxrwxr-x  3 root    root    4096 Sep 13 18:58 .
drwxr-x--x 10 root    root    4096 Sep 13 16:19 ..
-rw-r--r--  1 root    root      1 Sep 13 17:35 test
```

20. Защитить файл для записи со стороны других пользователей, проверить.

Добавим остальным возможность записи в файл командой `# chmod go+w test`, проверяем

```
# chmod go+w test
# ls -l
total 25
drwxrwxr-x  3 root    root    4096 Sep 13 18:58 .
drwxr-x--x 10 root    root    4096 Sep 13 16:19 ..
-rw-rw-rw-  1 root    root      1 Sep 13 17:35 test
```

Все могут читать и записывать в файл.

21. Открыть / закрыть свой основной каталог для доступа со стороны других пользователей, проверить.

Основной каталог `usr` имеет изначально такие права доступа:

```
drwxrwxr-x 11 root    root    4096 Sep 04 2019 usr
```

Закроем его для доступа всем, кроме владельца. Доступ к каталогу осуществляется в зависимости от состояния маркера `x` - выполнение. Команда `# chmod go-x /usr`, проверяем

```
drwxrw-r-- 11 root    root    4096 Sep 04 2019 usr
```

Как видно возможность «исполнить», открыть каталог есть только у владельца. Откроем обратно командой `#chmod go+x /usr`

```
drwxrwxr-x 11 root    root    4096 Sep 04 2019 usr
```

22. Разрешить доступ к своему основному каталогу, но запретить его изменение, проверить.

Изменение каталога – маркер `w`, запретим всем его изменение командой `#chmod a-w /usr`, проверяем:

```
dr-xr-xr-x 11 root    root    4096 Sep 04 2019 usr
```

Все могут зайти в него, посмотреть, но не изменить.

23. Разрешить доступ к файлам только с известными именами, проверить.

Сначала под суперпользователем, для каталога /LAB1 у остальных пользователей были отобраны права для чтения. # chmod o-r LAB1.

```
drwxrwx--x  3 root    root    4096 Sep 18 14:48 LAB1
```

Теперь перелогинившись под обычным юзером, доступа к каталогу /LAB1 нет, но обращаясь к заранее созданному файлу по известному имени, можно считать его содержимое:

```
$ ls /root
ls: Permission denied (/root)
$ ls /root/LAB1
ls: Permission denied (/root/LAB1)
$ ls /root/LAB1/test_23.txt
/root/LAB1/test_23.txt
$ cat /root/LAB1/test_23.txt
Test file for ex.23 :)$
```