

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

```
# ls -l /dev/hd0
brw----- 1 root    root      4,   0 Sep 23 04:53 /dev/hd0
# ls -l /dev/console
crw-rw-rw- 1 root    root      0,   1 Sep 23 04:53 /dev/console
# ls -l /dev/tty0
crw--w---- 1 root    tty       7,   0 Sep 22 21:58 /dev/tty0
# ls -l /dev/shmem
total 18716
-rwxrwxrwx  3 root    root      65536 Sep 22 21:53 Pf241693.8066310
-rw-rw-rw-  4 root    root      47520 Sep 22 21:53 Pgc01e0000
-rw-rw-rw-  4 root    root      27776 Sep 22 21:53 Pgc01e0001
-rw-rw----  3 root    display  9437184 Sep 22 21:53 aperture-80ee,beef,0:0
-rw-rw----  3 root    display   4096 Sep 22 21:53 ctl-80ee,beef,0
# ls -l /dev/mem
-rw----- 1 root    root     4294967295 Sep 22 21:58 /dev/mem
```

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

```
# pwd
/root
```

Потому что мы в сессии суперпользователя, то есть, root

3. Создать каталог LAB1 и сделать его рабочим.

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

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

```
# find / -name services
/etc/services
/usr/qnx650/target/qnx6/etc/services
find: Can't get stat. (/fs/cd0): No such device or address
# cat /etc/services
#
# 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
systat        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          timserver
time          37/udp          timserver
rtp           39/udp          resource
nameserver    42/tcp          name          # IEN 116
whois         43/tcp          nickname
domain        53/tcp          nameserver    # name-domain server
domain        53/udp          nameserver
mtp           57/tcp          # deprecated
# Bootp experimental (sellgren@vangogh)
bootp         67/udp          bootps        # bootp server
bootpc        68/udp          bootpc        # bootp client
#
```

5. Сколько скрытых файлов в вашем домашнем каталоге?

```
# ls -A ~ | grep -c '^\. '
6
```

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

```
# find /boot
/boot
/boot/fs
/boot/fs/qnxbase.ifs
/boot/fs/qnxbasedma.ifs
/boot/fs/qnxbasesmp.ifs
/boot/build
/boot/build/bios.build
/boot/build/finstall.build
/boot/build/qnxbase.build
/boot/build/qnxbasedma.build
/boot/build/qnxbasesmp-apic.build
/boot/build/qnxbasesmp.build
/boot/sys
/boot/sys/bios.boot
/boot/sys/bios16m.boot
/boot/sys/bios_nokbd.boot
/boot/sys/elf.boot
/boot/sys/ipl-diskpc1
/boot/sys/ipl-diskpc1-flop
/boot/sys/ipl-diskpc2
/boot/sys/ipl-diskpc2-flop
/boot/sys/libmod_aps.a
/boot/sys/nobios.boot
/boot/sys/procnto
/boot/sys/procnto-instr
/boot/sys/procnto-smp
/boot/sys/procnto-smp-instr
/boot/sys/startup-apic
/boot/sys/startup-bios
/boot/sys/startup-bios-32
# find /boot -type f -size -1024c | wc -l
6
# find /boot -type f -perm -111 | wc -l
10
```

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

```
# ls -ld /boot
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?

```
~
~
~

# ls -l file.txt
-rw-rw-r-- 1 root root 0 Sep 22 22:19 file.txt
```

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.

```
# mkdir dir
# i=1
# while [ $i -le 10 ]; do
> touch "dir/file$i.txt"
> i=`expr $i + 1`
> done
# ls dir
.                file10.txt      file4.txt       file7.txt
..               file2.txt      file5.txt       file8.txt
file1.txt        file3.txt      file6.txt       file9.txt
# mv dir/file1
file1.txt        file10.txt
# mv dir/file1.txt dir/file2.txt dir/file3.txt
mv: Target (dir/file3.txt) must be a directory in order to move
mv: directories or multiple files to it.
# mv dir/file1.txt dir/file2.txt dir/file3.txt .
# ls
.                dir             file1.txt       file3.txt
..               file.txt        file2.txt
# rm -i dir/file4.txt dir/file5.txt
rm: remove dir/file4.txt? (y/N) y
rm: remove dir/file5.txt? (y/N) y
# ls
.                dir             file1.txt       file3.txt
..               file.txt        file2.txt
# ls dir
.                file10.txt      file7.txt       file9.txt
..               file6.txt      file8.txt
```

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

```
# echo "$PATH"
/sbin:/usr/sbin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/opt/bin:/opt/sbin:/usr/qnx650
/host/qnx6/x86/usr/bin:/usr/qnx650/host/qnx6/x86/usr/sbin:/usr/qnx650/host/qnx6/x86/sbin:/u
sr/qnx650/host/qnx6/x86/bin:/usr/qnx650/host/qnx6/x86/usr/photon/appbuilder
# echo "$LOGNAME"
root
# echo "$HOME"
/root
# echo "$HOSTNAME"
localhost
# echo "$PWD"
/root/LAB1
# echo "$RANDOM"
29541
```

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

```
# 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
dispconf         ls              rundemo          waitfor
du               mkdir            script            who
dumpifs          mkefs           sendnto          zcat
echo             mketfs          sh
ed               mkifs           shutdown
# echo "$?"
0
# ls /pin
ls: No such file or directory (/pin)
# echo "$?"
1
```

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

```
# ls -C /bin > /tmp/bin_list.txt
# ls -C /usr/bin >> /tmp/bin_list.txt
# cat /tmp/bin_list.txt
.                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
dispconf         ls                    rundemo          waitfor
du               mkdir                  script           who
dumpifs          mkefs                 sendnto          zcat
echo             mketfs                sh
ed              mkifs                 shutdown
.               nslookup
..              nsupdate
alphalayer       ntos86-ld
arp              ntos86-ld-2.19
awk              ntpdc
basename         ntpq
bc               ntptrace
bindres          od
bunzip2          omshell
bzip2            op
bzcmp            openssl
bzdiff           passwd
bzegrep          paste
bzfgrep          patch
bzgrep           pdebug
bzip2            ph
bzip2recover     phin
bzless           phrelay
bzmore           phs-to-bjc
carmine-capture  phs-to-bmp
cfgopen          phs-to-escp2
```

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

```
# find /usr/bin -maxdepth 1 -name 'g*' -print | wc -l
7
# find /usr/bin -maxdepth 1 -name 't??' -print | wc -l
5
```

14. Сколько всего пользователей зарегистрировано в системе?

```
# cat /etc/passwd | wc -l
10
```

15. Сколько различных групп пользователей в системе?

```
# cat /etc/group | wc -l
15
```

16. Определить имена пользователей, у которых нет пароля.

```
# awk -F: '($2 == "" || $2 == "*") {print $1}' /etc/passwd
root
user
```

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

```
# ls
.          dir          file1.txt    file3.txt
..         file.txt     file2.txt
# chmod u-r file.txt
# cat file.txt
# ls -l file.txt
--w-rw-r-- 1 root      root              0 Sep 22 22:19 file.txt
```

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

```
# ls -l file.txt
--w-rw-r-- 1 root      root              0 Sep 22 22:19 file.txt
# chmod o-r file.txt
# ls -l file.txt
--w-rw---- 1 root      root              0 Sep 22 22:19 file.txt
#
```

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

```
# chmod u-w file.txt
# ls -l file.txt
---rw---- 1 root      root              0 Sep 22 22:49 file.txt
```

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

```
# ls -l file.txt
-rw-rw-r-- 1 root      root              0 Sep 22 22:51 file.txt
# chmod o-w file.txt
# ls -l file.txt
-rw-rw-r-- 1 root      root              0 Sep 22 22:51 file.txt
#
```

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

```
# ls -ld ~
drwxr-x--x 10 root      root          4096 Sep 22 22:01 /root
# chmod o+rx ~
# ls -ld ~
drwxr-xr-x 10 root      root          4096 Sep 22 22:01 /root
# chmod o-rx ~
# ls -ld ~
drwxr-x--- 10 root      root          4096 Sep 22 22:01 /root
```

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

```
# chmod u=rx,g=rx,o=rx ~
# ls -ld ~
dr-xr-xr-x 10 root      root          4096 Sep 22 22:01 /root
```

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

```
# chmod 000 file.txt file3.txt
# chmod 000 file.txt file3.txt
# chmod 644 file1.txt file2.txt
# cat file1.txt
# cat file2.txt
# cat file.txt
# cat file3.txt
# ls -l
total 24
drwxrwxr-x 3 root      root          4096 Sep 22 22:51 .
dr-xr-xr-x 10 root      root          4096 Sep 22 22:01 ..
drwxrwxr-x 2 root      root          4096 Sep 22 22:32 dir
----- 1 root      root              0 Sep 22 22:51 file.txt
-rw-r--r-- 1 root      root              0 Sep 22 22:31 file1.txt
-rw-r--r-- 1 root      root              0 Sep 22 22:31 file2.txt
----- 1 root      root              0 Sep 22 22:31 file3.txt
```

1. Написать скрипт, который просто выводит значения переданных ему параметров.

```
# cat 2.1.sh
#!/bin/sh

echo "Passed parameters: $@"
# ./2.1.sh one two three
Passed parameters: one two three
```

2. Написать скрипт, который с помощью утилит pidin и grep выводит на экран информацию об указанном по имени процессе.

```
# cat 2.2.sh
#!/bin/sh

if [ -z "$1" ]; then
    echo "Usage: $0 process_name"
    exit 1
fi

PROCESS_NAME="$1"

pidin | grep "$PROCESS_NAME"

# ps
      PID TTY          TIME CMD
  491539 ?                00:00:00 sshd
  499733 ?                00:00:00 -sh
  192535 ?                00:00:00 login
  192536 ?                00:00:00 login
  192537 ?                00:00:00 login
  270362 ?                00:00:00 login
  3141659 ?              00:00:00 ps
   245790 ?                00:00:00 phlogin2
   241695 ?                00:00:00 sshd
# ./2.2.sh sshd
  241695  1 usr/sbin/sshd      10o SIGWAITINFO
  491539  1 usr/sbin/sshd      10o READY
```

3. Написать скрипт, который компилирует указанную программу и при отсутствии ошибок запускает её. Если же есть ошибки, то автоматически вызывает редактор для их исправления.

```
# cat 2.3.sh
#!/bin/sh

if [ -z "$1" ]; then
    echo "Usage: $0 source_file.c"
    exit 1
fi

SOURCE_FILE="$1"
OUTPUT_FILE="${SOURCE_FILE%.c}.o"

gcc "$SOURCE_FILE" -o "$OUTPUT_FILE"
if [ $? -eq 0 ]; then
    ./"$OUTPUT_FILE"
else
    vi "$SOURCE_FILE"
fi
```

1. Написать программу, выводящую сообщение "HELLO" в центре чистого экрана.

```
# cat 3.1.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ioctl.h>
#include <unistd.h>

int main() {
    struct winsize w;
    ioctl(STDOUT_FILENO, TIOCGWINSZ, &w);

    const char *message = "HELLO";
    int row = w.ws_row / 2;
    int col = (w.ws_col - strlen(message)) / 2;

    system("clear");

    for (int i = 0; i < row; i++) {
        printf("\n");
    }

    for (int i = 0; i < col; i++) {
        printf(" ");
    }

    printf("%s\n", message);

    return 0;
}
```

2. Написать программу, позволяющую определять коды нажимаемых клавиш и восстанавливающую исходный вид терминала (цвет, курсор) при выходе.

```
# cat 3.2.c
#include <stdio.h>
#include <stdlib.h>
#include <termios.h>
#include <unistd.h>

struct termios orig_term;

void restore_terminal(void) {
    tcsetattr(STDIN_FILENO, TCSANOW, &orig_term);
    printf("\033[0m\033[?25h");
}

int main(void) {
    struct termios new_term;
    unsigned char c;

    if (tcgetattr(STDIN_FILENO, &orig_term) != 0) return 1;
    atexit(restore_terminal);

    new_term = orig_term;
    new_term.c_lflag &= ~(ICANON | ECHO);
    new_term.c_cc[VMIN] = 1;
    new_term.c_cc[VTIME] = 0;
    if (tcsetattr(STDIN_FILENO, TCSANOW, &new_term) != 0) return 1;

    printf("\033[2J\033[H");
    printf("Press keys (ESC to exit):\n");

    while (1) {
        ssize_t n = read(STDIN_FILENO, &c, 1);
        if (n <= 0) break;
        if (c == 27) break;
        printf("Key code: %d\n", c);
    }

    return 0;
}
```

3. Написать программу, рисующую движущийся символ (при выключенном курсоре, без использования функции стирания экрана).

```
# cat 3.3.c
#include <stdio.h>
#include <unistd.h>
#include <termios.h>

int main(void) {
    int x, y;
    int dx = 1, dy = 1;
    int width = 40, height = 10;
    char ch = '0';

    printf("\033[?25l");

    x = 1;
    y = 1;

    while (1) {
        printf("\033[%d;%dH%c", y, x, ch);
        fflush(stdout);
        usleep(100000);

        printf("\033[%d;%dH ", y, x);

        x += dx;
        y += dy;

        if (x <= 1 || x >= width) dx = -dx;
        if (y <= 1 || y >= height) dy = -dy;
    }

    printf("\033[?25h");
    return 0;
}
```

4. Написать программу, рисующую бесконечно движущийся символ. Характер движения (скорость, направление, цвет и т.д.) задавать с помощью параметров командной строки. Предусмотреть восстановление параметров дисплея (цвет, курсор) при принудительном завершении программы. Осуществить запуск нескольких экземпляров программы с разными параметрами движения (запуск с одного терминала, вывод на другой).
-