

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Институт информатики и вычислительной техники

09.03.01 "Информатика и вычислительная техника"
профиль "Программное обеспечение средств
вычислительной техники и автоматизированных систем"

Кафедра прикладной математики и кибернетики

Операционные системы реального времени

Лабораторная работа №1

Среда пользователя UNIX

Выполнил:

студент гр.ИП-213

«__» _____ 2025 г.

Дмитриев Атон Александрович
ФИО студента

Проверил:

Преподаватель

ФИО преподавателя

«__» _____ 2025 г.

Оценка _____

Новосибирск 2025 г.

1. Работа с командной строкой (составить протокол по выполнению всех пунктов)

1. Определить тип файлов /dev/hd0, /dev/console, /dev/ttyp0, /dev/shmem, /dev/mem.

```
# ls -l /dev/hd0
brw----- 1 root      root          4,   0 Sep 13 00:52 /dev/hd0
# ls -l /dev/console
crw-rw-rw- 1 root      root          0,   1 Sep 13 00:52 /dev/console
# ls -l /dev/ttyp0
crw--w---- 1 root      root          7,   0 Sep 12 18:02 /dev/ttyp0
# ls -l /dev/shmem
total 18568
nrwxrwxrwx 3 root      root          65536 Sep 12 17:52 Pf237596.8066310
nrw-rw---- 3 root      display       9437184 Sep 12 17:52 aperture-80ee.beef.0:0
nrw-rw---- 3 root      display       4096 Sep 12 17:52 ctl-80ee.beef.0
# ls -l /dev/mem
-rw----- 1 root      root          4294967295 Sep 12 18:02 /dev/mem
```

Первый символ:

- — обычный файл;
- d — каталог;
- b — блочное устройство;
- c — символьное устройство;
- l — ссылка на др. файл;
- s — сокет;
- p — межпроцессорное взаимодействие;
- n — разделяемая память

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

Рабочий каталог при входе в систему — это домашний каталог пользователя. Он задается в файле /etc/passwd для каждого пользователя.

```
# pwd
/root
```

3. Создать каталог LAB1 и сделать его рабочим.

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

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

```
# ls -l /etc/services
-rw-r--r-- 1 root      root          4125 Oct 13 2007 /etc/services

# find / -name services -type f 2>/dev/null
/etc/services
/usr/qnx650/target/qnx6/etc/services

# cat /etc/services | less. только вниз( экономит оперативную память)
# cat /etc/services | more. вверх и вниз

#
# 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     ttyst source
chargen       19/udp    ttyst source
ftp           21/tcp
telnet        23/tcp
smtp          25/tcp     mail
time          37/tcp     timserver
time          37/udp    timserver
rlp           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
mtp           57/tcp    # deprecated
# Bootp experimental (sellgren@vangogh)
bootp          67/udp    bootps   # bootp server
bootpc         68/udp    # bootp client
--More--
```

5. Сколько скрытых файлов в вашем домашнем каталоге?

```
# ls -a ~ | grep "\." | wc -l
6
```

Скрытые файлы начинаются с точки.

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

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

```
# find /boot -type f -size -1k | wc -l
6
```

```
# find /boot -type f -executable | wc -l
find: Unrecognised primary expression (-executable)
0
```

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

```
# ls -ld /boot
drwxrwxr-x 5 root      root          4096 Sep 04 2019 /boot
```

У каталогов всегда как минимум 2 жесткие связи: одна на себя и одна из родительского каталога.

```
# ls -ld /boot/*
drwxrwxr-x 2 root      root          4096 Sep 04 2019 /boot/build/
drwxrwxr-x 2 root      root          4096 Sep 04 2019 /boot/fs/
drwxrwxr-x 2 root      root          4096 Sep 04 2019 /boot/sys/
```

2 (базовые связи) + 3 подкаталога.

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

```
# vi test.txt
```

Перешли в режим редактирования введя i

```
hello world!
~
```

Нажав esc перешли в командный режим

```
:wq
```

записали и вышли из файла

```
# cat test.txt
hello world!
```

Просмотр прав доступа

```
# ls -l test.txt
-rw-rw-r-- 1 root      root          13 Sep 12 19:28 test.txt
```

Права доступа по умолчанию для новых файлов определяются значением umask.

у вновь созданного файла установлены следующие флаги доступа:
-rw-rw-r-- (в числовом представлении: **664**).

Если требуется изменить права доступа (например, сделать файл более защищенным), используйте команду chmod:

chmod 644 test.txt - Установит права rw-r--r-- (только владелец может писать)

Или измените значение umask перед созданием файлов:

umask 022 - Теперь новые файлы будут создаваться с правами 644 (rw-r--r--)

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 test_dir
# cd test_dir/
# touch original.txt
# ls
.
..
# i=0
# while ((i++ < 10));
> do cp original.txt "copy_$i.txt"; done
# mv copy_1.txt copy_2.txt copy_3.txt ..
# ls
.
..
copy_10.txt
# cd ..
# ls
.
..
# 
# cd test_dir/
# rm -i copy_9.txt copy_10.txt
rm: remove copy_9.txt? (y/N) y
rm: remove copy_10.txt? (y/N) y
# ls
.
..
copy_4.txt
# 
$ ls -ld .
drwxrwxr-x 2 user      user          4096 Sep 12 21:15 .
$ rm copy_4.txt
$ ls
copy_10.txt   copy_6.txt   copy_8.txt   orig.txt
copy_5.txt   copy_7.txt   copy_9.txt
$ chmod -w .
$ rm copy_5.txt
rm: Can't unlink copy_5.txt: Permission denied

```

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/  
sr/bin:/usr/qnx650/host/qnx6/x86/usr/sbin:/usr/qnx650/host/qnx6/x86/sbin:/usr/q  
x650/host/qnx6/x86/bin:/usr/qnx650/host/qnx6/x86/usr/photon/appbuilder  
# echo $LOGNAME  
root  
# echo $HOME  
/root  
# echo $HOSTNAME  
vbox  
# echo $PWD  
/root/LAB1/test_dir
```

```
# echo $RANDOM  
8239  
# echo $RANDOM  
27164  
# echo $RANDOM  
14868
```

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

```
# ls /bin  
.  
..  
aps  
asa  
cat  
chgrp  
chmod  
chown  
confstr  
cp  
cpio  
csplit  
dd  
df  
dispconf  
du  
dumpifs  
echo  
ed  
# echo $?  
0
```

elvis	mkifsf_elf	slay
esh	mkifsf_openbios	slogininfo
ex	mkifsf_srec	split
false	mkxfs	stty
fesh	more	su
gunzip	mount	sync
gzip	mv	true
hostname	netmanager	uesh
igawk	on	umount
kill	pax	uname
ksh	pidin	uncompress
ln	ps	vg
login	pwd	vi
logout	rm	view
ls	rundemo	waitfor
mkdir	script	who
mkefs	sendnto	zcat
mketfs	sh	
mkifs	shutdown	

```
# 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 /bin | pr -3 -t > output.txt
```

```

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

```

```
# ls /usr/bin | pr -3 -t >> output.txt
```

```

.          gzip      pwd
..         hostname  rm
aps        igawk    rundemo
asa        kill     script
cat        ksh      sendnto
chgrp     ln       sh
chmod     login    shutdown
chown     logout   slay
confstr   ls       sloginfo
cp         mkdir   split
cpio      mkefs   stty
csplit    mketfs
dd         mkifs   su
df         mkifsf_elf
dispconf  mkifsf_openbios
du         mkifsf_srec
dumpifs   mkxfs
echo      more    sync
ed         mount   true
elvis     mv      uesh
esh       netmanager
ex        on      umount
false     pax     uname
fesh      pidin  uncompress
gunzip   ps      vg
.
..        ftp       patch
alphalayer fullpath pdebug
arp        funzip   ph
awk       gawk     phin
basename  get_hw_info phrelay
lsof     getconf  phs-to-bjc

```

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

```

# echo /usr/bin/g* | wc -w
    7
# echo /usr/bin/t?? | wc -w
    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 == ""){print $1}" /etc/passwd  
root  
user  
...
```

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

```
$ echo "test" > test.txt  
$ cat test.txt  
test  
$ chmod u-r test.txt  
$ ls -l test.txt  
--w-rw-r-- 1 user      user          5 Sep 12 21:09 test.txt  
$ cat test.txt  
test.txt: Permission denied  
$
```

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

```
# echo "testuser::1001:1001:Test user:/home/testuser:/bin/sh" >> /etc/passwd  
# cat /etc/passwd  
root::0:0:Superuser:/root:/bin/sh  
bin:x:1:1:Binaries Commands and Scripts:/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  
testuser::1001:1001:Test user:/home/testuser:/bin/sh  
# mkdir /home/testuser  
# chown 1001:1001 /home/testuser  
  
$ chmod o-r copy_5.txt  
$ cat copy_5.txt  
$ su testuser -c "cat copy_5.txt"  
copy_5.txt: Permission denied  
*
```

Chmod:

u - владелец файла;
g - группа файла;
o - все остальные пользователи;
- - забрать право;
+ - выдать право;
r - чтение;
w - запись;
x - выполнение;
s - выполнение от имени суперпользователя (дополнительный);

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

```
$ echo "some" > some.txt
$ echo "some" >> some.txt
$ cat some.txt
some
some
$ chmod u-w some.txt
$ echo "some" >> some.txt
/bin/sh: cannot create some.txt: Permission denied
$ su testuser -c "echo 'test' >> some.txt"
sh: cannot create some.txt: Permission denied
```

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

```
$ chmod u+w some.txt
$ chmod o-w some.txt
$ echo "some1" >> some.txt
$ cat some.txt
some
some
some1
$ su testuser -c "echo 'test' >> some.txt"
sh: cannot create some.txt: Permission denied
```

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

```
$ chmod o-rwx ~  
$ su testuser -c "ls ~$USER"  
ls: Permission denied (/home/user/)  
$ chmod o+rwx ~  
$ su testuser -c "ls ~$USER"  
LAB1
```

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

```
$ chmod o+r ~  
  
$ chmod o-w ~  
  
$ su testuser -c "ls ~$USER"  
LAB1 private.txt public.txt  
$ su testuser -c "touch ~$USER/file"  
/home/user//file: Permission denied
```

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

```
$ chmod o-r ~  
$ chmod o+r ~/public.txt  
$ su testuser -c "cat ~$USER/public.txt"  
$ echo "test" >> public.txt  
$ su testuser -c "cat ~$USER/public.txt"  
test  
$ su testuser -c "ls ~$USER"  
ls: Permission denied (/home/user/)  
$ su testuser -c "cat ~$USER/private.txt"  
$ chmod o-r ~/private.txt  
$ su testuser -c "cat ~$USER/private.txt"  
/home/user//private.txt: Permission denied  
$
```

2. Создание простых скриптов

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

```
$ chmod +x print_param.sh
$ ./print_param.sh param1 param1 param3
count params: 3
all param: param1 param1 param3
parametr 1: param1
parametr 2: param1
parametr 3: param3
$ cat print_param.sh
#!/bin/sh

echo "count params: $#"
echo "all param: $@"

count=1
for param in "$@"; do echo "parametr $count: $param"; count=$((count + 1)); done
$
```

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

```
$ chmod +x process_info.sh
$ ./process_info.sh sbin/tinit
./process_info.sh[6]: [1: cannot execute - No such file or directory
      2   1 sbin/tinit          10o REPLY      1
$ cat process_info.sh
#!/bin/sh

if [ $# -eq 0 ]; then
    echo "print name of proces in parametr"
    exit 1
fi
PROCESS_NAME=$1

pidin | grep -i "$PROCESS_NAME"
```

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

```
#!/bin/sh

if [ $# -eq 0 ]; then
    echo "Usage: $0 <source.c>"
    exit 1
fi
basename=$(echo "$1" | sed "s/\..c$//")
cc -o "basename" "$1" 2> errors.log

if [ $? -eq 0 ]; then
    ./basename"
else
    echo "Compilation errors. Opening editor..."
    vi errors.log
fi
$ ./run_or_die.sh
Usage: ./run_or_die.sh <source.c>
$ ./run_or_die.sh source.c
Hello, test's done!/n$ _
```

```
#include <stdio.h>

int main(){
    printf("Hello, test's done!/n");
    return 0;
}
```

3. Разработка программ

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

```
#include <stdio.h>
#include <string.h>

int main(){
    int i;
    printf("\033[2J");
    printf("\033[16;50H");
    printf("HELLO");
    printf("\033[20;1H");
    return 0;
}
```

```
HELLO
```

```
$ -
```