Федеральное государственное бюджетное образовательное учреждение «Сибирский государственный университет телекоммуникаций и информатики»

Кафедра ПМиК

Лабораторная работа №1

по дисциплине

«Операционные системы реального времени»

Выполнил: студент 4 курса

ИВТ, гр. ИП-813 Пещеров В.А.

Проверила:

Белевцова Е.А.

Новосибирск 2021

Оглавление

[Задание 3](#_Toc64498226)

[Выполнение 5](#_Toc64498227)

[Листинг проекта 9](#_Toc64498228)

# Задание

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

1. Определить тип файлов /dev/hd0, /dev/console, /dev/ttyp0, /dev/shmem, /dev/mem.
2. Определить, какой каталог делается рабочим при входе в систему. Почему?
3. Создать каталог LAB1 и сделать его рабочим.
4. Определить (с помощью программы ls), в каком каталоге содержится файл services. Посмотреть его содержимое.
5. Сколько скрытых файлов в вашем домашнем каталоге?
6. Определить полное дерево подкаталогов в /boot . Сколько там файлов, размер которых меньше 1К байт? Сколько там исполняемых файлов?
7. Сколько жестких связей у каталога /boot и почему?
8. Создать текстовый файл с помощью редактора vi. Какие флаги доступа устанавливаются у вновь создаваемого файла? Почему? Как это исправить?
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.
10. Определить значения переменных среды PATH, LOGNAME, HOME, HOSTNAME, PWD, RANDOM. Меняются ли они со временем?
11. Определить коды завершения команд ls /bin и ls /pin
12. Вывести содержимое каталога /bin в файл в несколько колонок. Затем добавить к нему распечатку каталога /usr/bin.
13. Сколько файлов удалили бы команды rm /usr/bin/g\* и rm /usr/bin/t?? ? (просьба файлы не удалять)
14. Сколько всего пользователей зарегистрировано в системе?
15. Сколько различных групп пользователей в системе?
16. Определить имена пользователей, у которых нет пароля.
17. Защитить файл для чтения со стороны владельца, проверить.
18. Защитить файл для чтения со стороны других пользователей, проверить.
19. Защитить файл для записи со стороны владельца, проверить.
20. Защитить файл для записи со стороны других пользователей, проверить.
21. Открыть / закрыть свой основной каталог для доступа со стороны других пользователей, проверить.
22. Разрешить доступ к своему основному каталогу, но запретить его изменение, проверить.
23. Разрешить доступ к файлам только с известными именами, проверить.

### 2. Создание простых скриптов

1. Написать скрипт, который просто выводит значения переданных ему параметров.
2. Написать скрипт, который с помощью утилит pidin и grep выводит на экран информацию об указанном по имени процессе.
3. Написать скрипт, который компилирует указанную программу и при отсутствии ошибок запускает её. Если же есть ошибки, то автоматически вызывает редактор для их исправления.

### 3. Разработка программ

1. Написать программу, выводящую сообщение "HELLO" в центре чистого экрана.
2. Написать программу, позволяющую определять коды нажимаемых клавиш и восстанавливающую исходный вид терминала (цвет, курсор) при выходе.
3. Написать программу, рисующую движущийся символ (при выключенном курсоре, без использования функции стирания экрана).
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).
5. Программно реализовать команду по заданию преподавателя.

# Выполнение

### **1. Работа с командной строкой (составить протокол по выполению всех пунктов)**

1. file /dev/hd0 - block special (4/0)

file /dev/console - character special (0/1)

file /dev/ttup0 - character special (7/0)

file /dev/shmem - directory

file /dev/mem - writable, regular file, no read permission

2. USER: pwd - /home/user - обычный пользователь

SUPERUSER: pwd - /root - суперпользователь(администратор)

3. mkdir LAB1

cd LAB1

4. ls /etc

cat /etc/services

5. ls -a | wc -l

6 файлов (с учетом . и ..)

6. find /boot - дерево подкаталогов

find /boot -size 1[c] | wc -l - количество файлов, меньше 1КБ

(6 файлов)

cd /boot/sys

ls -F\* (17)

cd ../boot/fs

ls -F\* (3)

cd ../boot/build

ls -F\* (6)

7. ls -l ../.. (из папки home/user)

boot имеет 5 жестких связей

8. vi test

Исходные флаги доступа -rw-rw-r--

-rw- права на чтение и запись для владельца

-rw- права на чтение и запись для группы пользователей

r-- права на чтение для остальных пользователей в системе

Флаги ставятся для защиты файла от несанкционированного доступа

С помощью chmod можно установить новые права

9. mkdir LAB1

cd dir1

vi test

cp test test1

cp test test2

cp test test3

cp test test4

cp test test5

cp test test6

cp test test7

cp test test8

cp test test9

cp test test10

mv test5 .. перемещение 3-х файлов из lab1 в user (вышележащий каталог)

mv test6 ..

mv test7 ..

rm -i test8 удаление с подтверждением

rm -i test9

rm -i test10

Если изменить права доступа на директорию (-t), то файл с флагом w нельзя удалить.

10. echo $HOME - /home/user

echo $LOGNAME - user

echo $PATH - /bin:/usr/bin:.......(очень длинное)

echo $PWD - /home/user - текущий рабочий каталог

echo $RANDOM - 31574 (меняется со временем)

11. ls /bin

echo $? - 0 (успех)

ls /pin

echo $? - 1 (провал)

12. ls -C /bin > test

cat test

ls -C /usr/bin > test

cat test

13. echo rm /usr/bin/g\* - 7 файлов

echo rm /usr/bin/t\* - 14 файлов

14. cat /etc/passwd | wc -l - 10 пользователей

15. cat /etc/group | wc -l - 15 групп

16. cat /etc/passwd | grep -v x | wc -l - 2 пользователя (root и user)

17. chmod u-r test - u-user, - - запретить, r - read

18. chmod o-r test - o-other, - - запретить, r - read

19. chmod u-w test - u-user, - - запретить, r - write

20. chmod o-w test - o-other, - - запретить, r - write

21. chmod o+rwx LAB1 - o-other, + - разрешить, r - read, w - write, x -eXecute

chmod o-rwx LAB1 - o-other, - - запретить, r - read, w - write, x -eXecute

22. chmod ugo+rx LAB1 - u - user, g - group, o - other, + - разрешить, r - read, x - eXecute

chmod ugo-w LAB1 - u - user, g - group, o - other, - - запретить, w - write

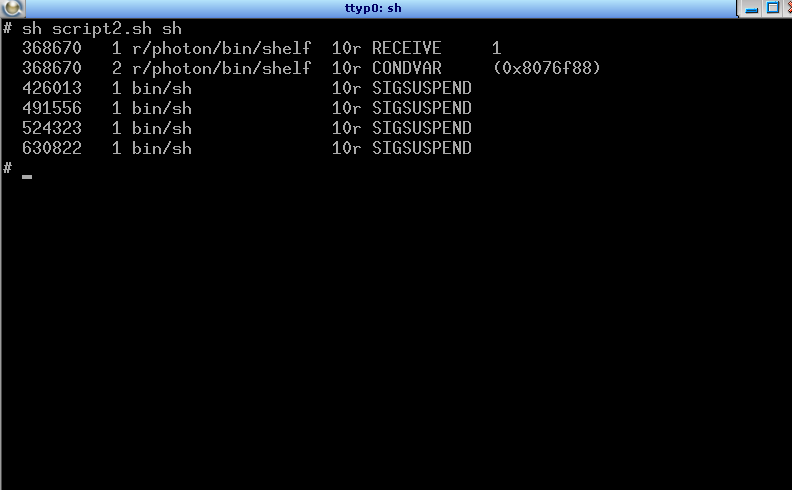
23. chmod ug+x LAB1 - u - user, g - group, + - разрешить, x - eXecute

chmod o-x LAB1 - o - other, x - eXecute

### **2. Создание простых скриптов**



Скрипт 1



Скрипт 2

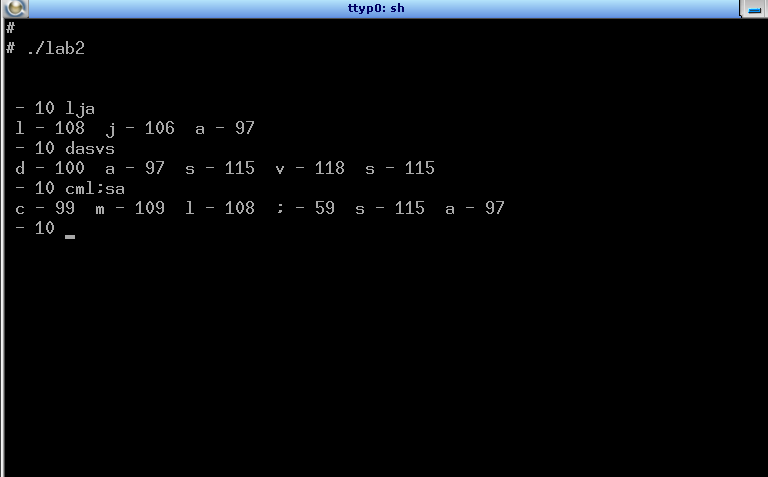


Скрипт 3

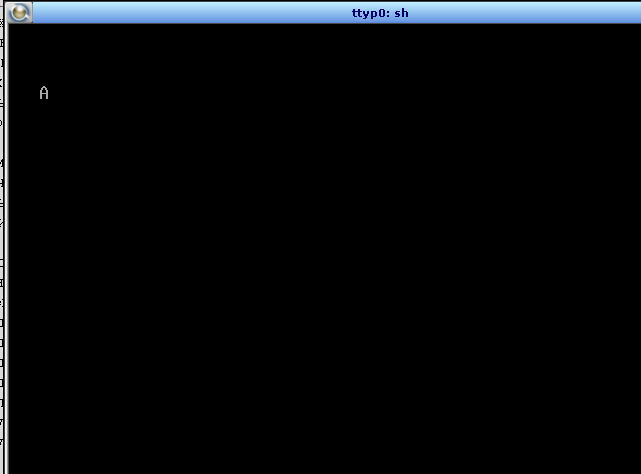
### **3. Разработка программ**



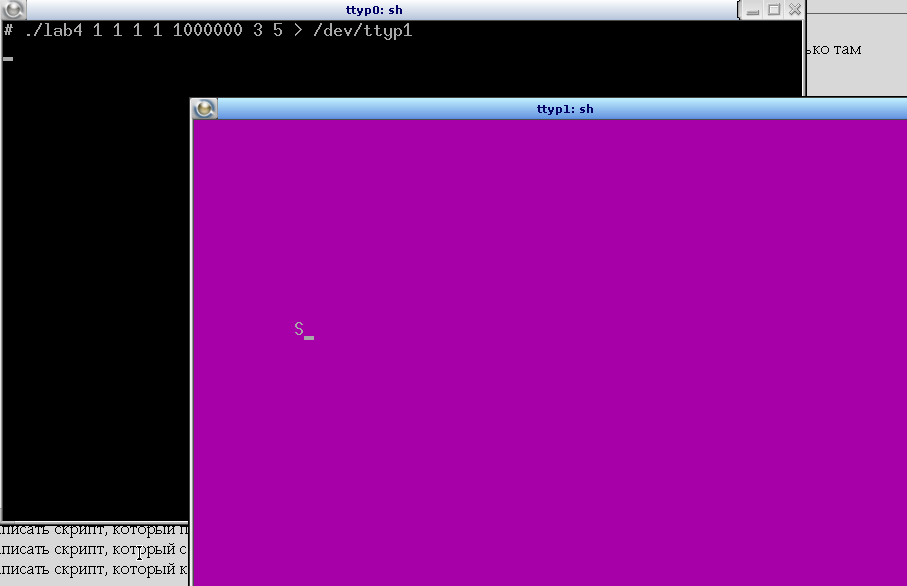
Программа 1



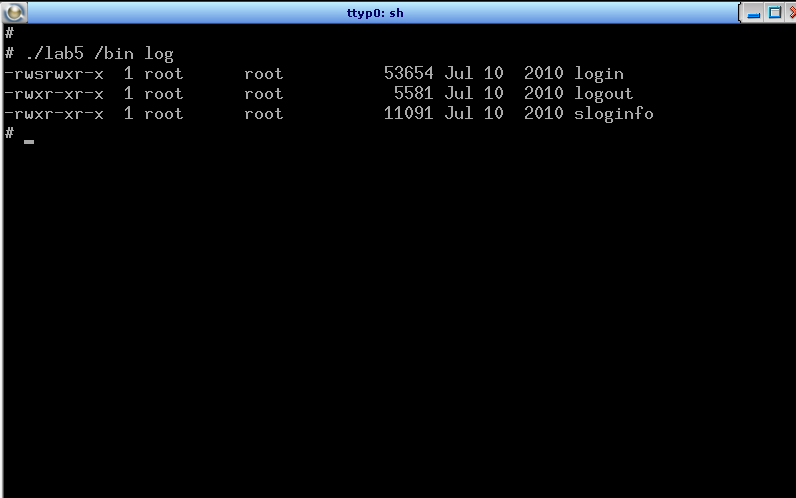
Программа 2



Программа 3



Программа 4



Программа 5

# Листинг проекта

Скрипт 1:

#!/bin/bash

i=1

for argument in "$@"

do

echo "$argument"

let "i+=1"

done

exit 0;

Скрипт 2:

#!/bin/bash

pidin | grep $1

if [ $? -ne 0 ]; then

echo "Bad command name"

exit 1;

fi

exit 0;

Скрипт 3:

#!/bin/bash

gcc $1.c -o $1

if [ $? -ne 0 ]; then

gedit $1.c

exit 1;

fi

./$1

exit 0;

Программа 1:

#include <stdio.h>

int main(void)

{

printf("\033[2J");

printf("\033[12;34H");

printf("Hello, world!\n");

return 0;

}

Программа 2:

#include <stdio.h>

#include </root/raw.h>

int main(void)

{

setraw();

int c = 0;

while (c != 50)

{

c = getchar();

printf("%c - %d ", (char)c, c);

}

printf("\033[2J");

printf("\033[1;1H");

printf("\033[=7F");

printf("\033[=0G");

unsetraw();

return 0;

}

Программа 3:

#include <stdio.h>

int main(void)

{

int x = 1;

int y = 1;

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

printf("\033[2J");

while (x != 10)

{

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

printf("A");

fflush(stdout);

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

printf(" ");

sleep(1);

x++;

y++;

}

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

printf("\033[2J");

return 0;

}

Программа 4:

#include <stdio.h>

#include <stdlib.h>

#include <signal.h>

#include <unistd.h>

void sigint\_handler(int sig)

{

printf("\033[=7F");

printf("\033[=0G");

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

printf("\033[2J");

exit(1);

}

int main(int argc, char\* argv[])

{

int x = atoi(argv[1]);

int y = atoi(argv[2]);

int x\_dx = atoi(argv[3]);

int y\_dy = atoi(argv[4]);

int speed = atoi(argv[5]);

int symbol\_color = atoi(argv[6]);

int background\_color = atoi(argv[7]);

signal(SIGINT, sigint\_handler);

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

printf("\033[=%dF", symbol\_color);

printf("\033[=%dG", background\_color);

printf("\033[2J");

fflush(stdout);

for (;;)

{

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

printf("A");

fflush(stdout);

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

printf(" ");

usleep(speed);

x += x\_dx;

y += y\_dy;

}

return 0;

}

Программа 5:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/wait.h>

#include <sys/types.h>

int main(int argc, char\* argv[])

{

char \*a1 = "ls";

char \*a2 = "-l";

char \*a3 = argv[1];

char \*a4 = "grep";

char \*a5 = argv[2];

int fd[2];

if (pipe(fd) == -1)

{

fprintf(stderr, "Pipe create error\n");

exit(1);

}

switch (fork())

{

case -1:

printf("Proc 1 create error\n");

exit(1);

case 0:

if (close(fd[0]) == -1)

{

fprintf(stderr, "Closing pipe 1 error\n");

exit(1);

}

if (fd[1] != 1)

{

if (dup2(fd[1], 1) == -1)

{

fprintf(stderr, "Dup2 error\n");

exit(1);

}

if (close(fd[1]) == -1)

{

fprintf(stderr, "Closing pipe 2 error\n");

exit(1);

}

}

execlp(a1,a1,a2,a3, 0);

exit(1);

default:

break;

}

switch (fork())

{

case -1:

printf("Proc 2 create error\n");

exit(1);

case 0:

if (close(fd[1]) == -1)

{

fprintf(stderr, "Closing pipe 2 error\n");

exit(1);

}

if (fd[0] != 0)

{

if (dup2(fd[0], 0) == -1)

{

fprintf(stderr, "Dup1 error\n");

exit(1);

}

if (close(fd[0]) == -1)

{

fprintf(stderr, "Closing pipe 2 error\n");

exit(1);

}

}

execlp(a4,a4,a5, 0);

exit(1);

default:

break;

}

if (close(fd[0]) == -1)

{

fprintf(stderr, "Closing pipe 1 error\n");

exit(1);

}

if (close(fd[1]) == -1)

{

fprintf(stderr, "Closing pipe 2 error\n");

exit(1);

}

if (wait(NULL) == -1)

{

exit(1);

}

if (wait(NULL) == -1)

{

exit(1);

}

return 0;

}