ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ЛАБОРАТОРНАЯ РАБОТА № 1

Выполнил:

Студент группы: ИП-712

Алексеев С.В.

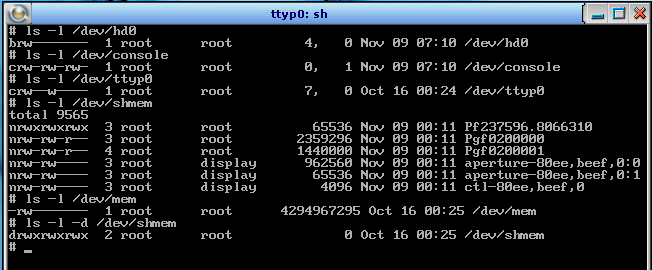
Проверил: профессор кафедры ПМиК

Фионов А.Н.

Новосибирск – 2020

**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.



Каталог /dev принадлежит менеджеру процессов и содержит файлы устройств

/dev/hd0 тип файла: блочное устройство (Block special file)

жесткие диски и разделы дисков, являются представлены специальными файлами, называемыми блочными устройствами. Эти файлы могут быть записаны и прочитаны случайным образом, чтобы считывайте и манипулируйте содержимым диска. Блочные устройства-это обозначается буквой b в первом символе списка ls-l

/dev/console тип файла: символьное устройство (Character special file)

Специальные файлы, называемые символьными устройствами представляют собой еще один вид устройств ввода-вывода. Подобно блочным устройствам, эти устройства устройства можно считывать и манипулировать ими, но в случае символа устройства, они должны быть прочитаны и записаны последовательно, а не случайным образом. Символьные устройства обозначаются буквой c в первом столбце выход ls-l.

/dev/ttyp0 тип файла: символьное устройство (Character special file)

Виртуальное устройство, принадлежащее менеджеру процессов (procnto),

которое преобразуется в управляющее терминальное устройство,

связанное с сеансом любого процесса, открывающего файл.

/dev/shmem тип файла: каталог (Directory)

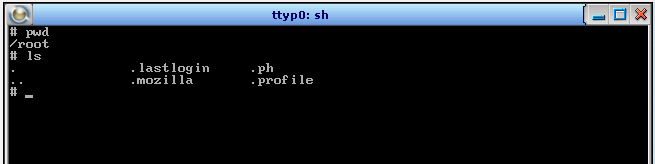
Содержит файлы, представляющие области общей памяти в системе

(также иногда используется для общих файлов с отображением памяти)

/dev/mem тип файла: обычный файл (Regular file)

Устройство, представляющее всю физическую память

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



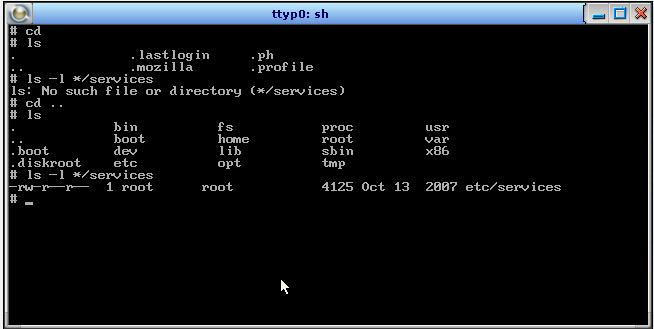
каталог /root

При первой установке ОСРВ QNX Neutrino автоматически создается единственная учетная запись пользователя с именем root.

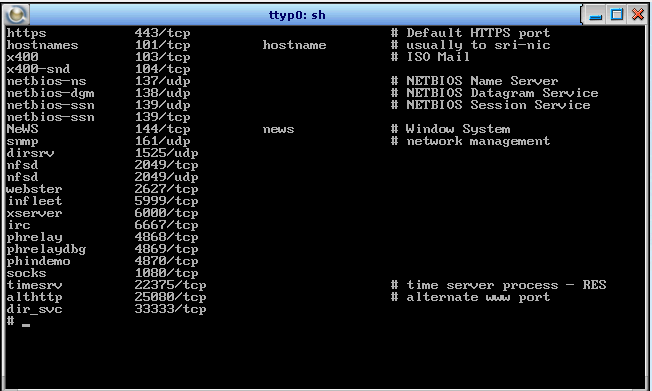
3) Создать каталог LAB1 и сделать его рабочим.



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

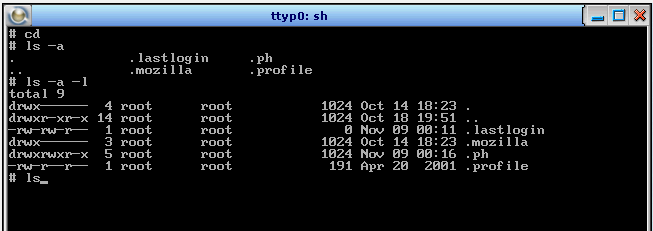


Файл Services содержится в каталоге etc/



Содержимое файла services можно посмотреть командой: cat ect/services

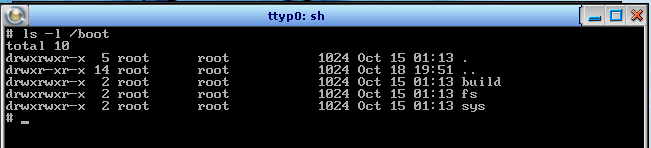
5) Сколько скрытых файлов в вашем домашнем каталоге?



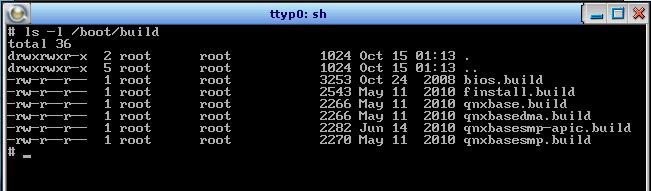
Посмотреть скрытые файлы в каталоге можно командой: ls -a

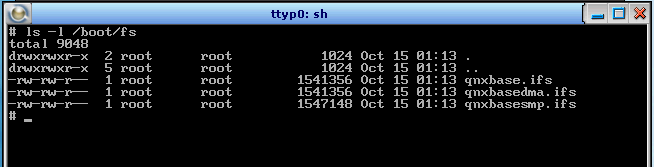
Как видим в домашнем каталоге 4 скрытых файла

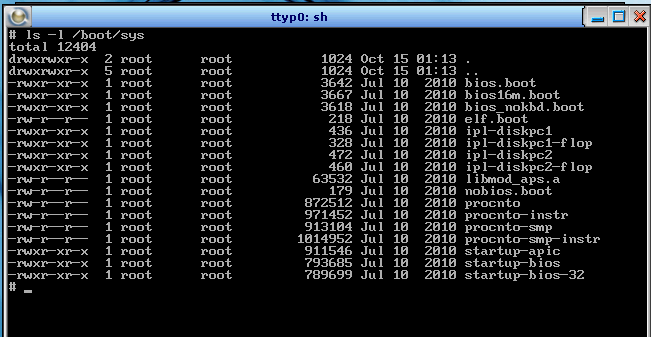
6) Определить полное дерево подкаталогов в /boot . Сколько там файлов, размер которых меньше 1К байт? Сколько там исполняемых файлов?

****

Подкаталоги в каталоге /boot: build, fs и sys



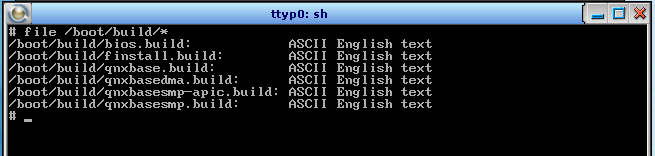


****

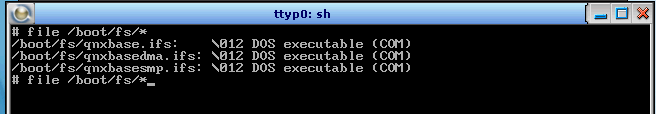
Видим, что в данных подкаталогах всего 6 файлов (elf.boot, ipl-diskpc1, ipl-diskpc1-flop, ipl-diskpc2, ipl-diskpc2-flop, nobios.boot), чей размер меньше 1 Кбайта или 1024 байт.

Исполняемыми файлами называются файлы, содержащие в себе готовые к запуску компьютерные программы.

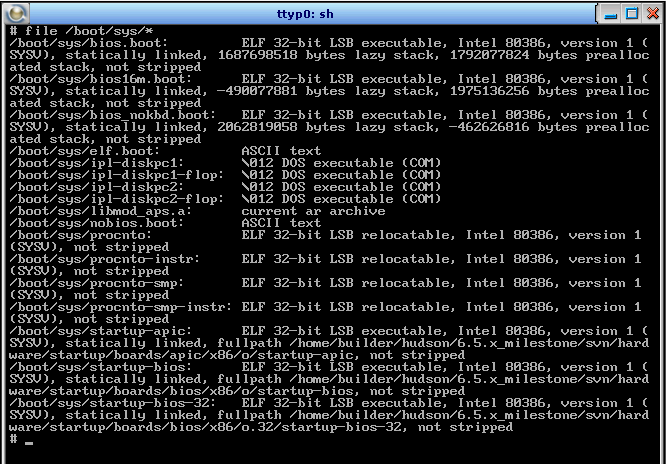
**Стандартный исполняемый файл** Executable and Linking Format **ELF формата содержит множество сегментов, например такие как: *.text* - сегмент кода, *.data* - сегмент инициализированных данных (обычно только для чтения, сюда попадают строки, константные массивы и т.п.), *.bss* - сегмент неинициализированных данных программы, *.dynamic* - сегмент служебных данных о подгружаемых библиотеках во время запуска программы, *.comment* - информация о версиях, *.init* - код инициализации процесса, *.fini* - код окончания выполнения процесса и т.д. Этих сегментов большое множество, но в основном они все стандартные и несут в себе информацию, необходимую для правильной загрузки исполняемого файла в память и требуемых им библиотек. Также есть сегменты, отвечающие за хранение отладочной информации (*.debug*) и прочих опциональных данных (*.line*, *.p\_paddr*), которые не являются критическими для загрузки и выполнения исполняемого файла и могут быть безопасно удалены.**

****

В файле /build нет исполняемых файлов

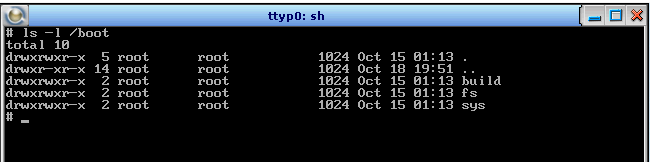


В файле /fs три исполняемых файла с расширением .ifs

В файле /sys 10 файлов ELF формата

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

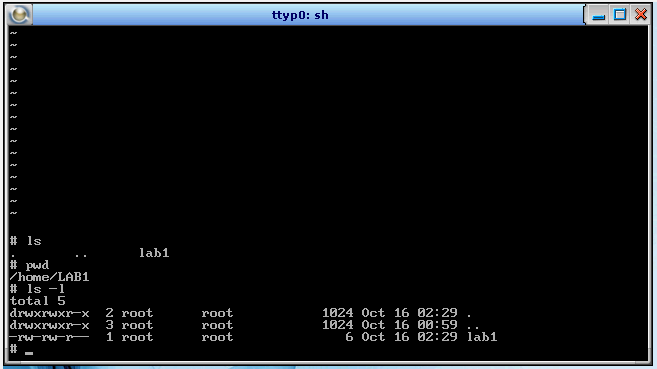
Жесткими связями у каталога называются структурные составляющие файла — описывающий его элемент каталога



total 10 отражает количество существующих по данному пути указателей нас существующие файлы.

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

Создали файл командой: vi lab1



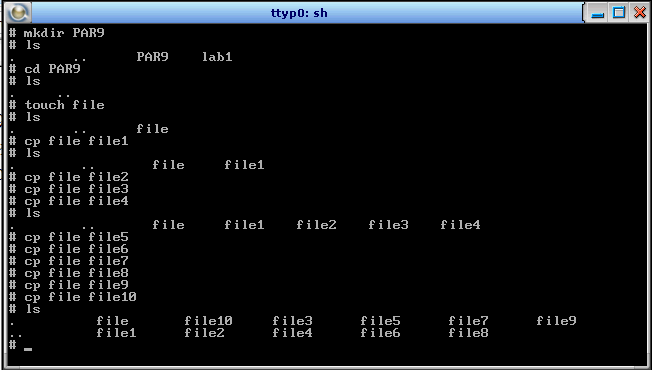
Флаги доступа у файла:

-rw-rw-r-- обозначают, что владелец файла и члены группы, к которой принадлежит владелец имеют право на чтение и запись, а остальные пользователи имеют право только на чтение файла.

Чтобы исправить это, нужно использовать команду UMASK “значение новой маски”

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.

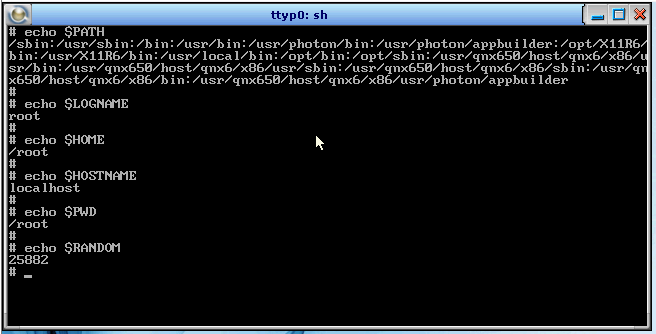
Создаем каталог «PAR9», создаем файл «file» в данном каталоге и делаем десять копий этого файла



Перемещаем три файла (file10, file9, file8) в вышестоящий каталог

И удаляем некоторые файлы из каталога PAR9 командой «rm» c опцией -i, для запрашивания подтверждения перед удалением каждого существующего файла

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



Переменная PATHпользователя root включает в себя каталоги, которые содержат системные исполняемые файлы. Меняется вручную

LOGNAME показывает имя пользователя под которым он вошел в систему. Меняется от пользователя

HOME содержит имя домашнего каталога под которым зашел пользователь. Меняется в зависимости какой пользователь пошел в систему

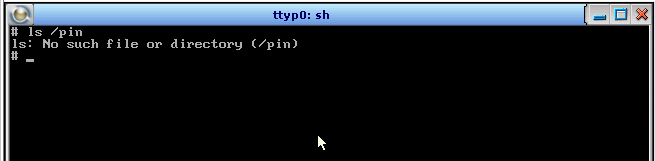
HOSTNAME показывает текущую хост-систему. Не меняется

PWD показывает текущий рабочий каталог. Меняется в зависимости от того, в каком каталоге работаем

RANDOM генерирует случайное десятичное число. Меняется с каждым запуском команды

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



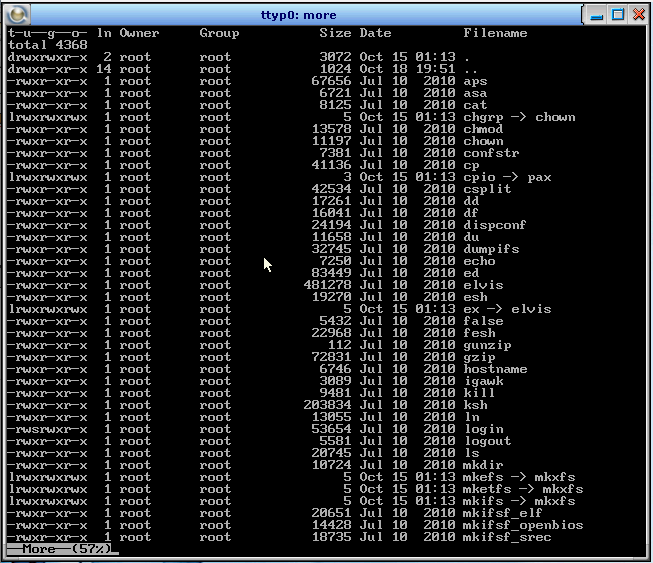


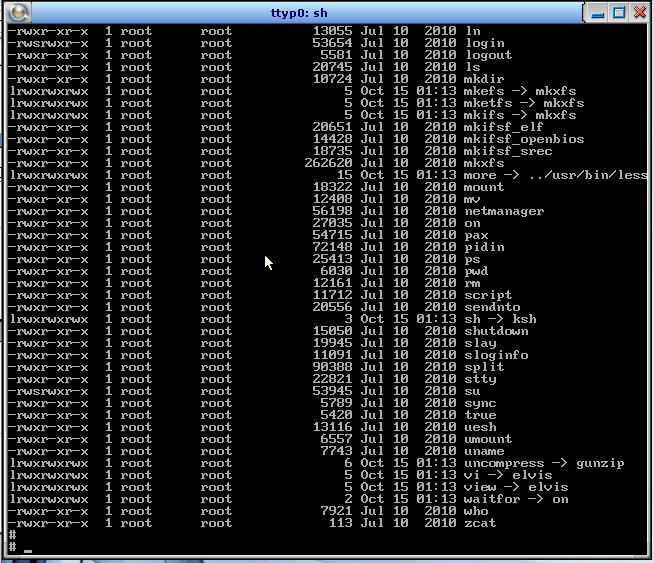
12) Вывести содержимое каталога /bin в файл в несколько колонок. Затем добавить к нему распечатку каталога /usr/bin.

Создали файл “bin” в каталоге “PAR9” и вывели в него содержимое каталога /bin в несколько колонок с помощью команд:

#touch bin

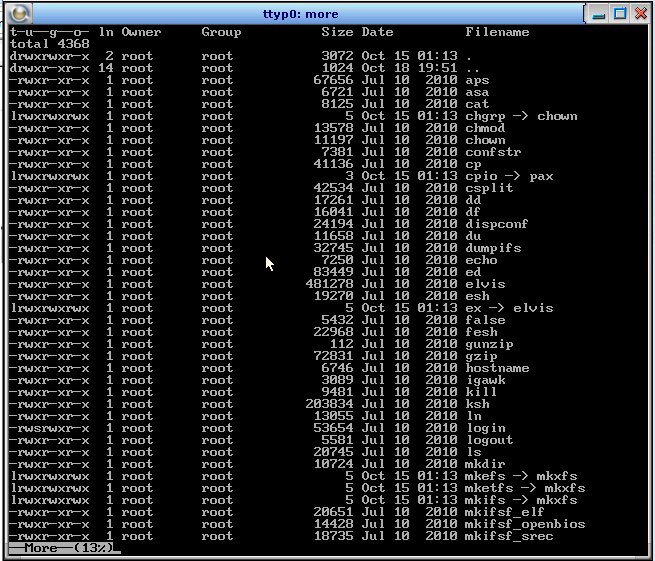
#ls -l -h /bin >bin

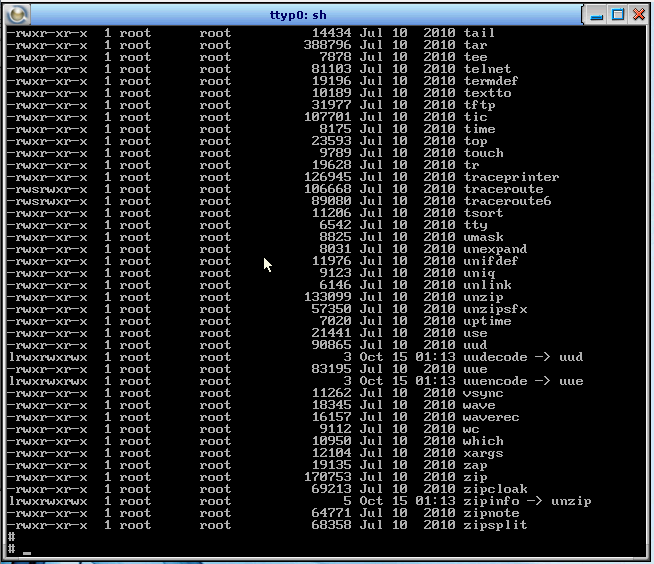




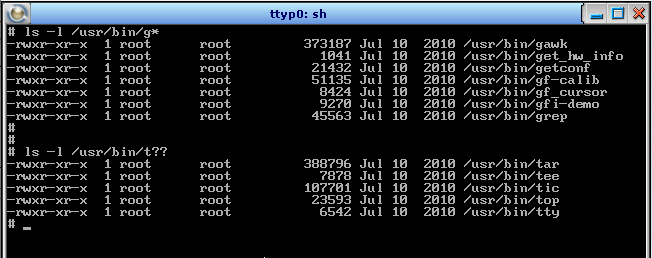
Затем добавили к файлу bin распечатку каталога /usr/bin командой

ls -l -h /usr/bin >>bin





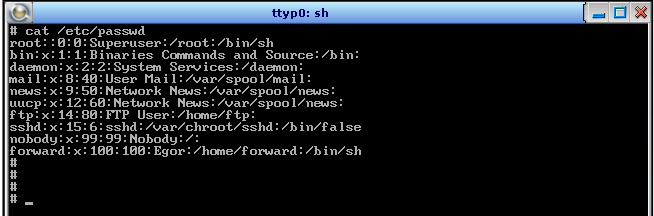
13) Сколько файлов удалили бы команды rm /usr/bin/g\* и rm /usr/bin/t?? ? (просьба файлы не удалять)



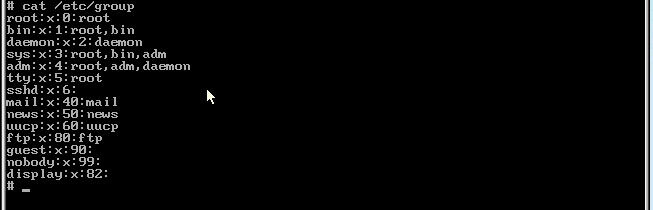
Команда «rm /usr/bin/g\*» удалила б 7 файлов

Команда «rm /usr/bin/t??» удалила б 5 файлов

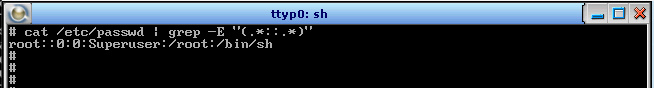
14) Сколько всего пользователей зарегистрировано в системе?



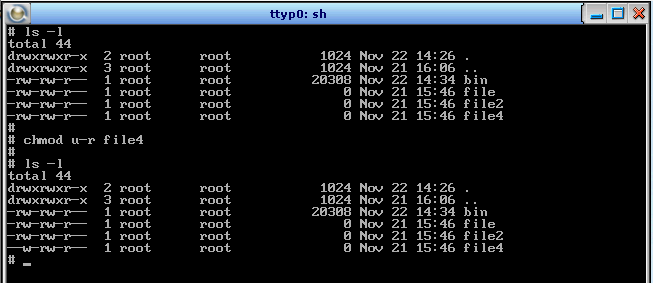
15) Сколько различных групп пользователей в системе?



16) Определить имена пользователей, у которых нет пароля.



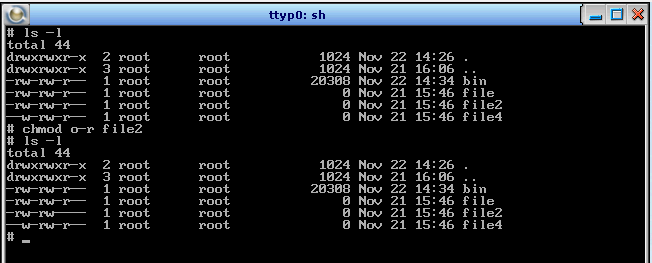
17) Защитить файл для чтения со стороны владельца, проверить.



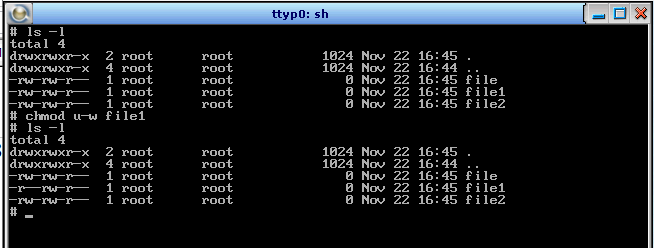
Защитили файл «file4» для чтения со стороны владельца командой

#chmod u-r file4

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

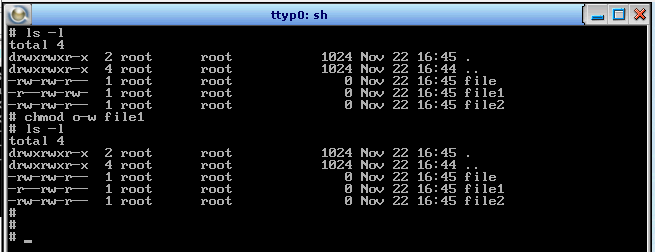


19) Защитить файл для записи со стороны владельца, проверить.



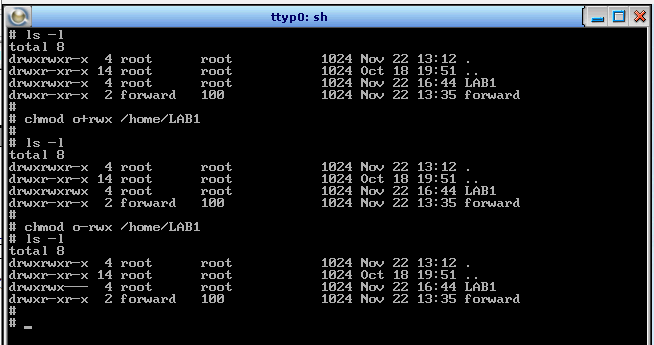
Защитили файл «file1» для записи со стороны владельца

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



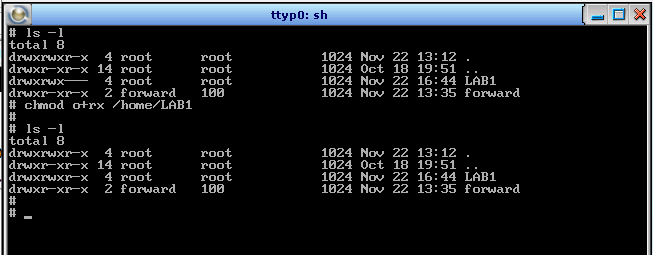
Защитили файл “file1” для записи со стороны других пользователей

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

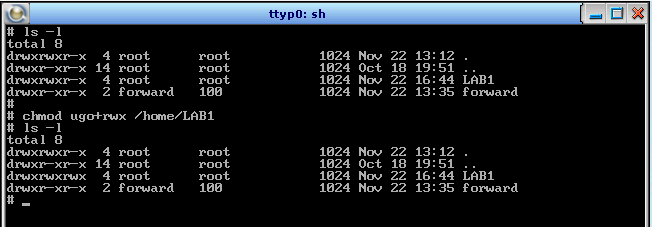


Открыли и закрыли каталог LAB1 для доступа со стороны других пользователей

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



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



P.S. Для проверки доступа к файлу

Создаем файл file5.txt:

#vi file5.txt

Защищаем файл от других пользователей:

#chmod o-rwx

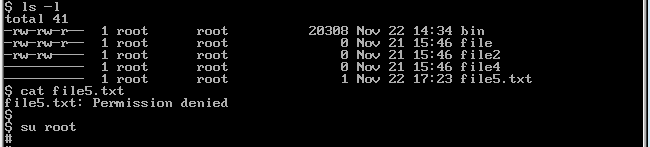
Заходим на другого пользователя, который существует в системе, в нашем случае пользователь «forward»:

#su – forward

И проверяем доступ к данному файлу:

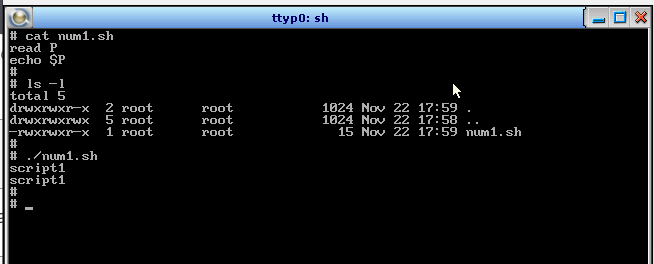
#cat file5.txt

В результате чего увидим:



### 2. Создание простых скриптов

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



Создали файл num1.sh:

#vi num1.sh

И прописали в нем команды:

read P

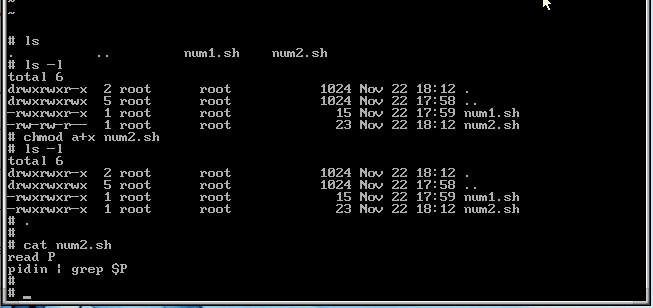
echo $P

Назначили права всем пользователям на исполнение:

#chmod a+x num1.sh

И запускаем скрипт и вводим любой параметр.

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



Создали файл «num2.sh»:

#vi num2.sh

Прописали в файле команды:

read P

pidin | grep $P

Назначили права всем пользователям на исполнение:

#chmod a+x num2.sh

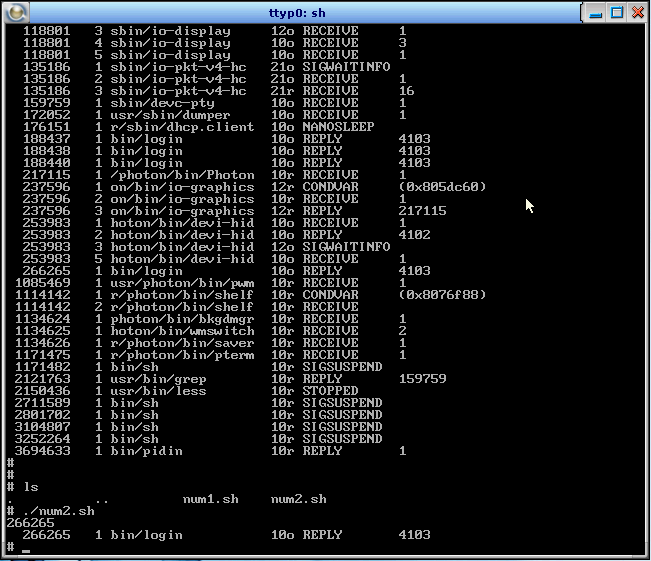
И запускаем наш скрипт и вводим имя процесса

#./num2.sh

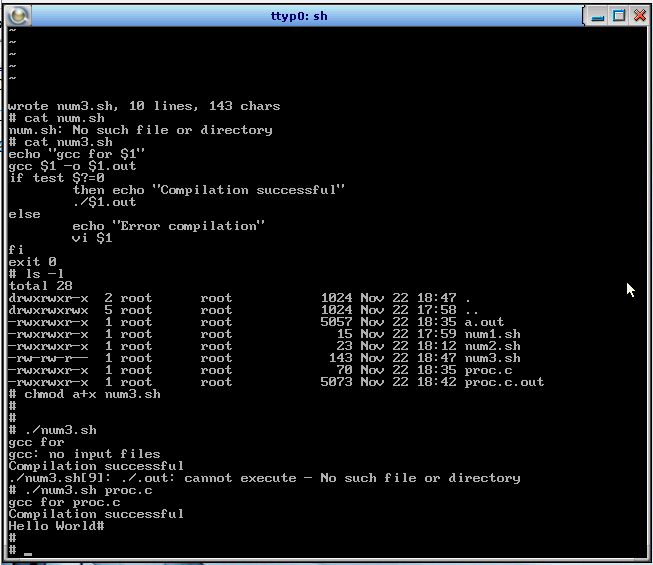
266265

Получаем информацию о процессе 266265 из pidin

Pidin - Просмотр статистики системы



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

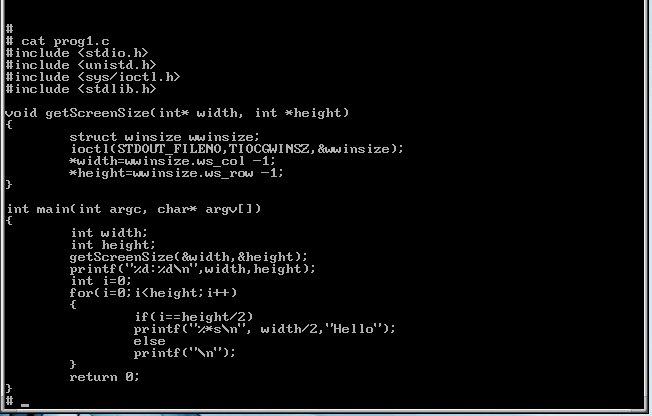


### 3. Разработка программ

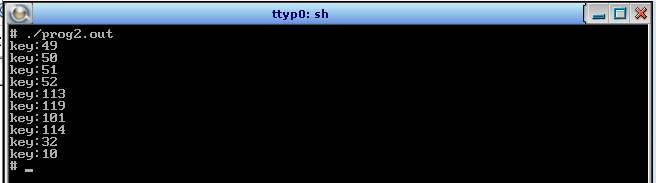
1) Написать программу, выводящую сообщение "HELLO" в центре чистого экрана.



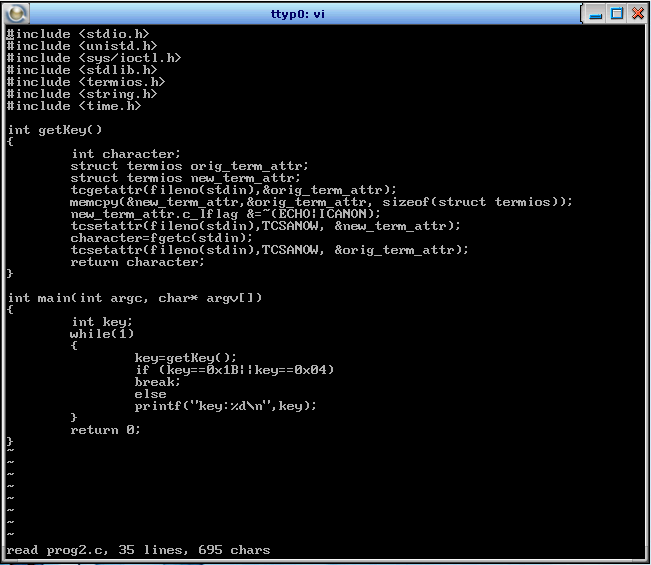
Файл «prog1.c»:



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

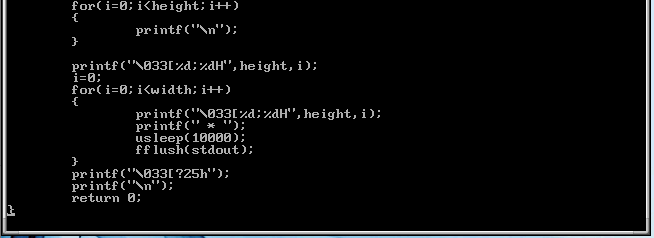
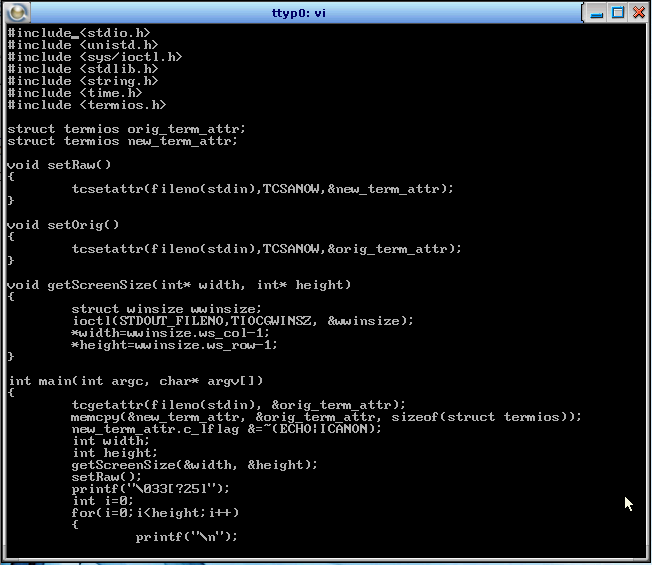


Файл **Prog2.c**



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

Файл **prog2.c**



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).

