Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

Лабораторная работа №1

«[Среда пользователя UNIX](https://eios.sibsutis.ru/mod/assign/view.php?id=81235)»

Выполнил: студент 4 курса

ИВТ, гр. ИП-813

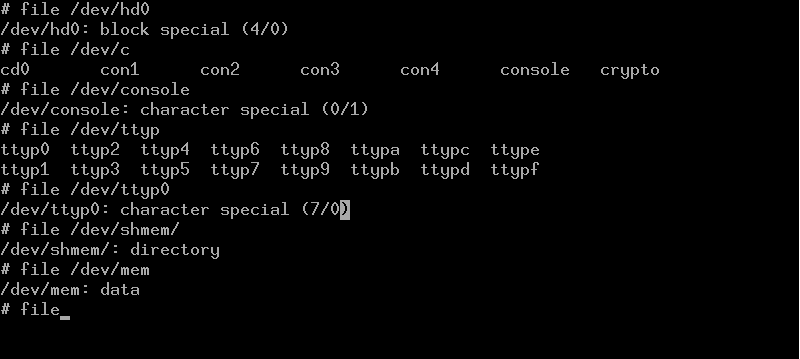
Бурдуковский И.А. Проверил: преподаватель кафедры ПМиК

Белевцова Екатерина Андреевна

Новосибирск, 2021 г.

# Определить тип файлов /dev/hd0, /dev/console, /dev/ttyp0, /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/ttypo0 /dev/shmem /dev/mem

/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/ttyp0 – тоже символьный файл

/dev/shmem – директория

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

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

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

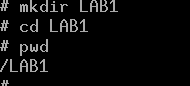


**/root –** рабочий каталог по умолчанию для суперпользователя. Является рабочим по умолчанию, потому что так было решено разработчиками ОС.

# 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.

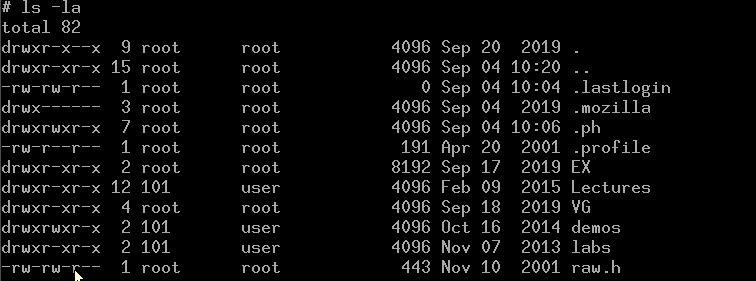


# Определить (с помощью программы ls), в каком каталоге содержится файл services. Посмотреть его содержимое.

Чтобы просмотреть содержимое файла воспользуемся командой # cat

/etc/services

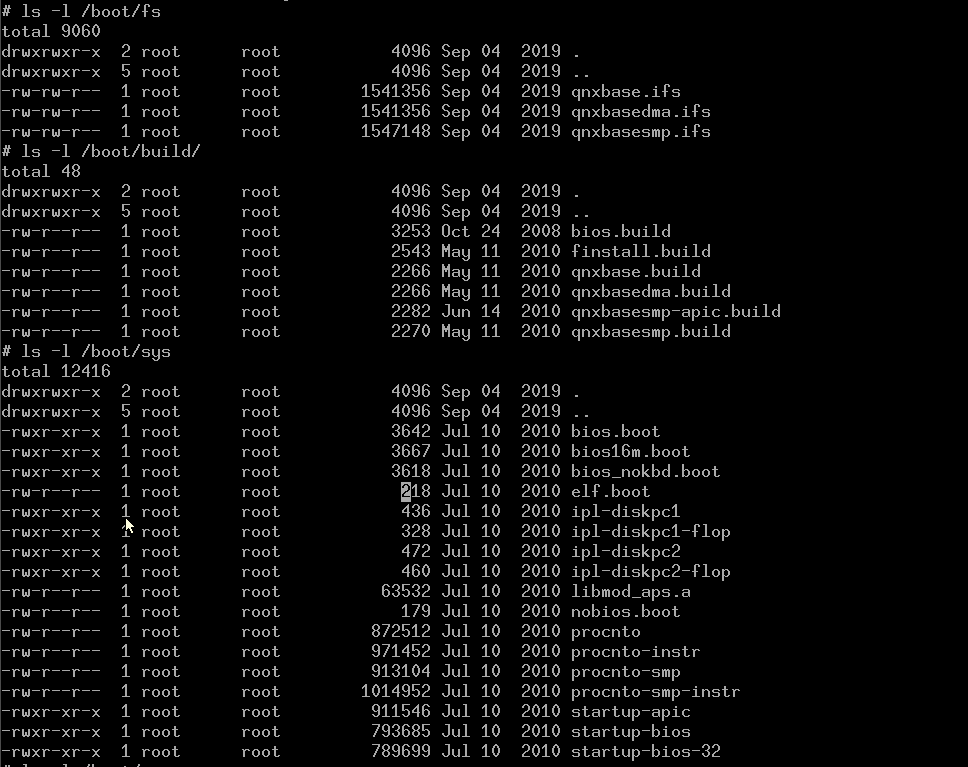
# Сколько скрытых файлов в вашем домашнем каталоге?

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

В домашнем каталоге 4 скрытых файла

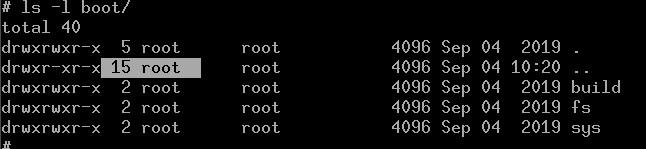
# Определить полное дерево подкаталогов в /boot . Сколько там файлов, размер которых меньше 1К байт? Сколько там исполняемых файлов?

Для отображения дерева подкаталогов в директории /boot использовать команду ls чтобы увидеть структуру, и, впоследствии, использовать команду ls на самих подкаталогах. Имеем 6 файлов весом меньше 1КБ.



# 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.



# Создать текстовый файл с помощью редактора vi. Какие флаги

**доступа устанавливаются у вновь создаваемого файла? Почему? Как это исправить?**

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



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.

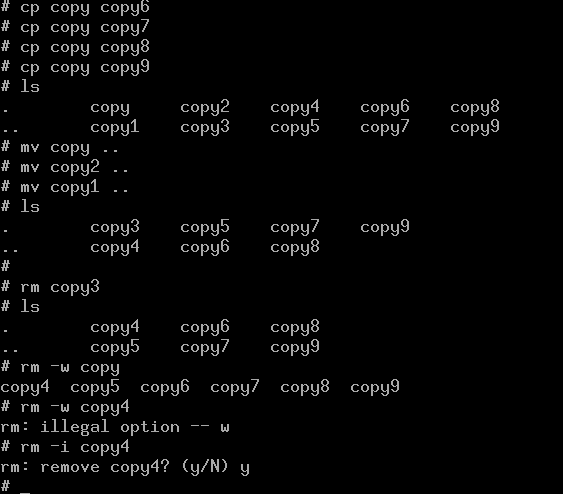
Для изменения флагов доступа существует утилита chmod

# 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, для создания файла – touch. Для того чтобы скопировать файл 10 раз напишем copy. Для переноса файлов – mv, для удаления rm.



Файлы были удалены с подтверждением.

Состояние флага w не влияет на удаление файла.

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

$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.

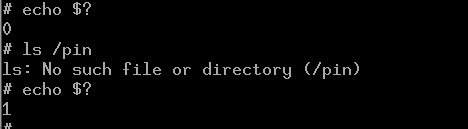
$HOSTNAME – имя компьютера, не изменяется.

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

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

# Определить коды завершения команд ls /bin и ls /pin

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

Код завершения - 0 – т.к. команда успешно завершилась, выведя результат.  
  
Код завершения – 1. Т.к. работа команды завершилась ошибкой – нет каталога.

# 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.

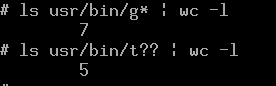


Cat – проверка что файл успешно записался. Чтобы добавить таким же образом данные в уже существующий файл необходимо использовать управляющий символ «>>»

# 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.



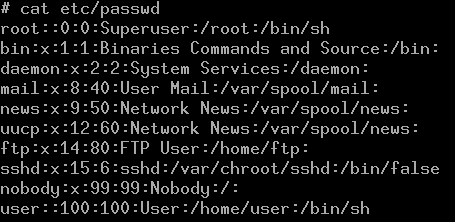
Сначала удалится 7 файлом, а затем 5.

# 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:обо лочка

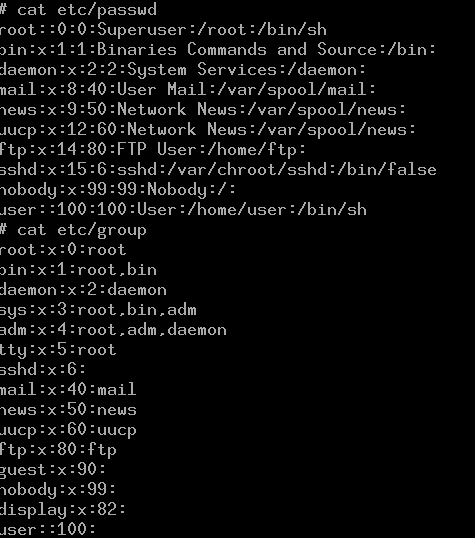
Пользователи с ID < 100 – системные, они были созданы во время установки. Пользователь с ID = 0 – суперпользователь.



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

# Сколько различных групп пользователей в системе?

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



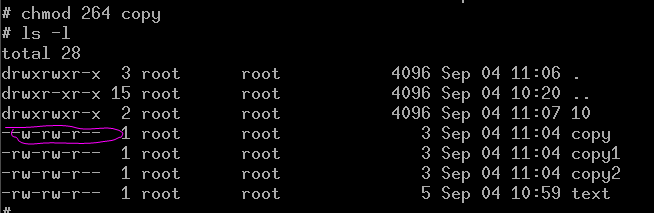
# Определить имена пользователей, у которых нет пароля.

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»

# Защитить файл для чтения со стороны владельца, проверить.

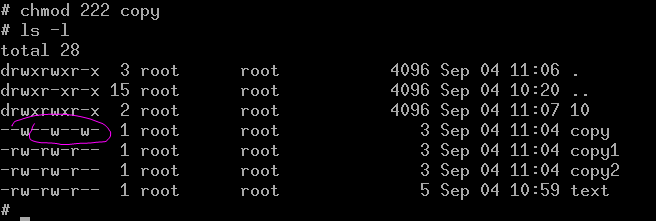
По умолчанию установились флаги доступа: для пользователя (user) – чтение (r), запись(w); для группы (group) - чтение (r), запись(w); для других (other) – чтение(r).



Как видно, только у владельца нет прав чтения для файла test.

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

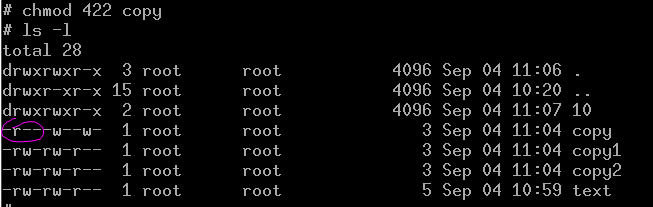
Изменим права доступа на запись для файла



Как видно все не могут читать файл test.

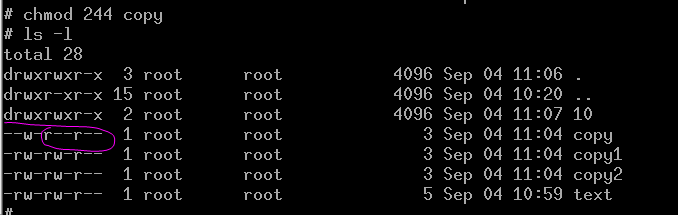
# Защитить файл для записи со стороны владельца, проверить.

Изменяем права, проверяем



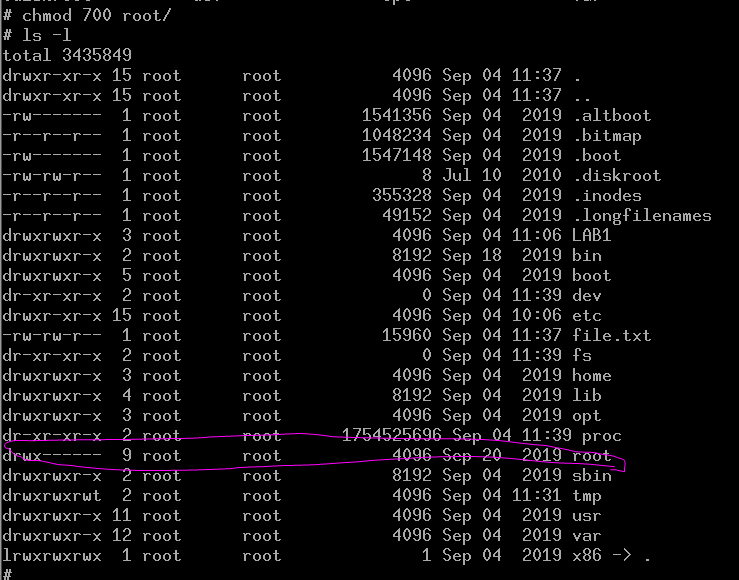
# Защитить файл для записи со стороны других пользователей, проверить.

Изменяем права, проверяем



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

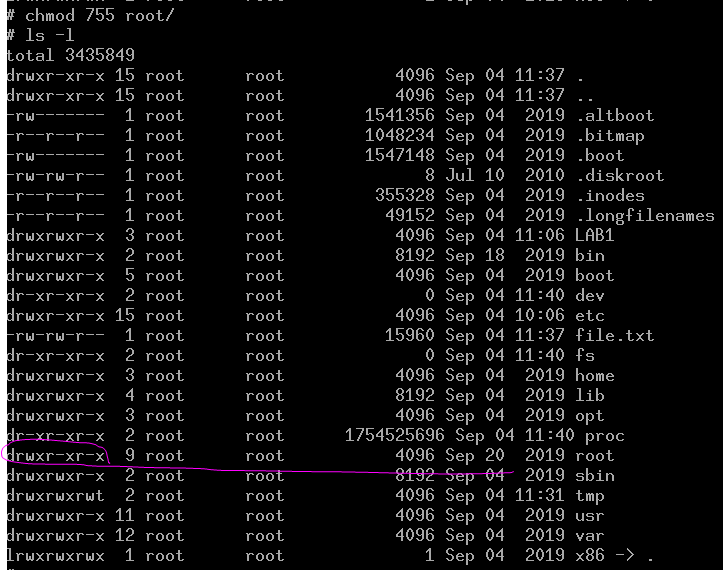
Закроем его для доступа всем, кроме владельца.



Как видно возможность «исполнить», открыть каталог есть только у владельца.

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

Изменение каталога:



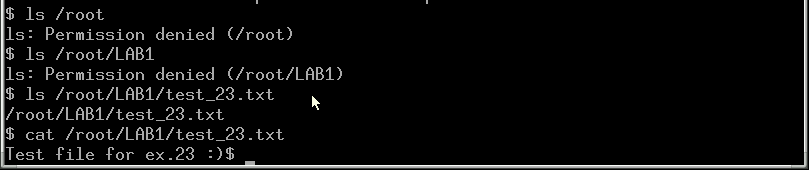
Все могут зайти в него, посмотреть, но не изменить.

# 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.

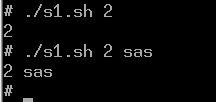
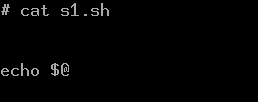


Теперь перелогинившись под обычным юзером, доступа к каталогу /LAB1 нет, но обращаясь к заранее созданному файлу по известному имени, можно считать его содержимое:



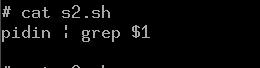
# 2. Создание простых скриптов

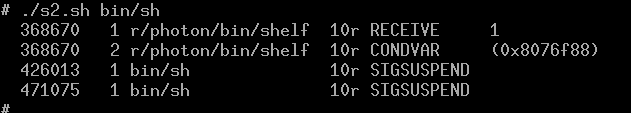
# Написать скрипт, который просто выводит значения переданных ему параметров.

Результат работы скрипта с 1 аргументом и с 2 различными аргументами, как видно скрипт работает корректно.

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

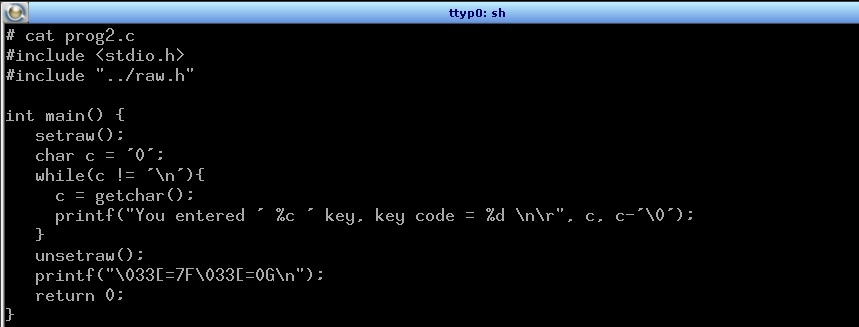
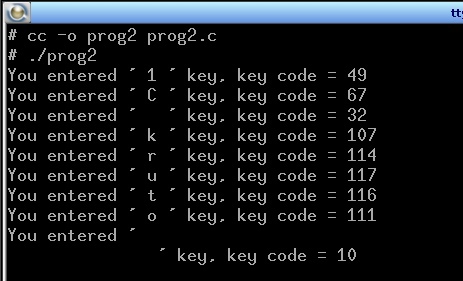
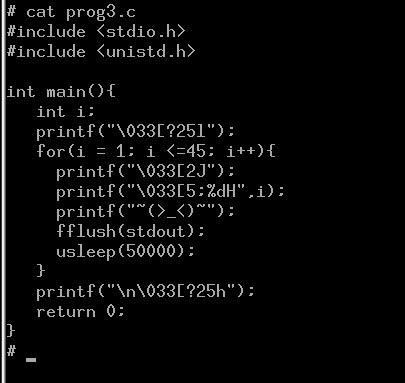
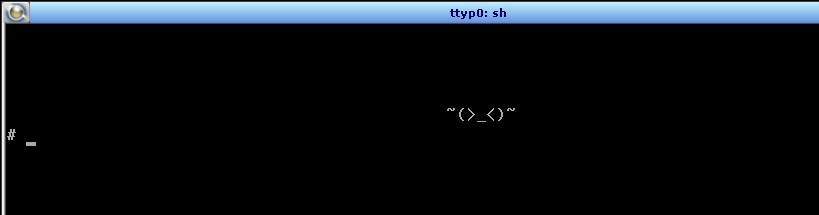
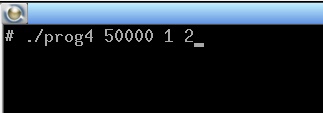
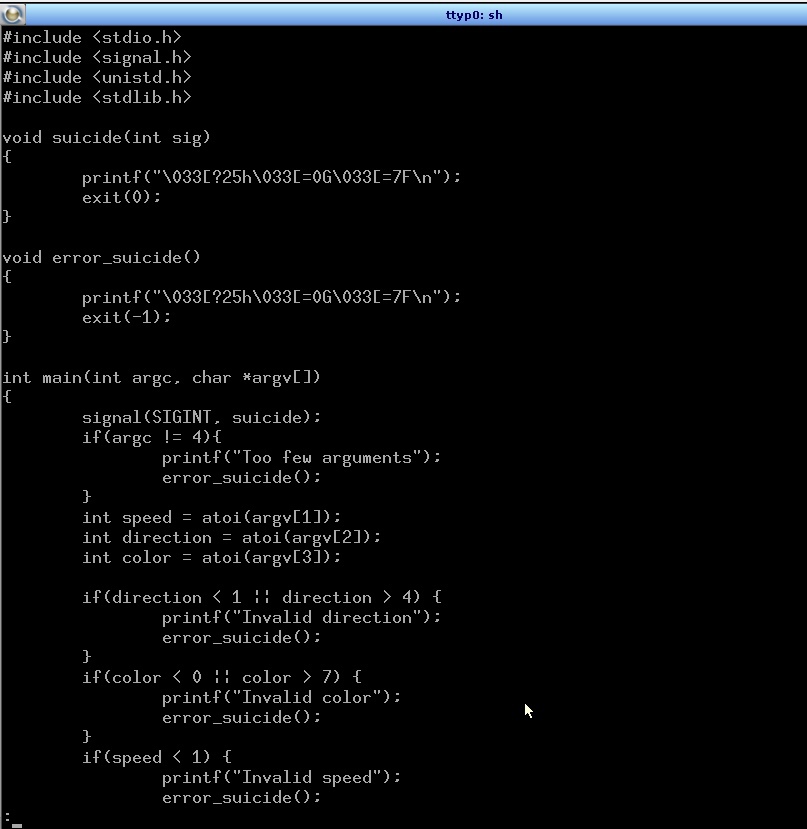
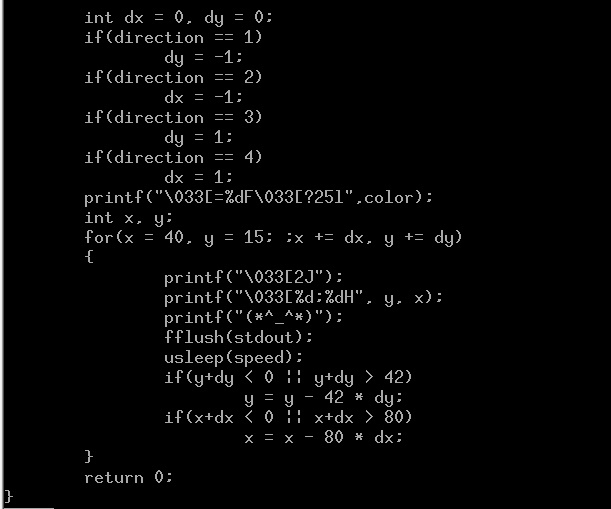
Данный скрипт требует только один аргумент – имя процесса. Получив имя процесса запускается сам скрипт, внутри которого pipeline из команд pidin и grep результата по переданному имени процесса через аргумент.



# Написать скрипт, который компилирует указанную программу и при отсутствии ошибок запускает её. Если же есть ошибки, то автоматически вызывает редактор для их исправления. Данный скрипт сначала компилирует указанную программу, далее следует проверка – был ли создан исполняемый файл, если да – запускает его; если нет – то запускается редактор vi для исправления ошибок. Пример удачной работы скрипта: Пример неудачной работы:

**3. Создание простых скриптов**

# Написать программу, выводящую сообщение “HELLO” в центре экрана Для выполнения задания была написана программа с использованием esc- последовательности: «\033[y;xH» - установка курсора в позицию (x,y).

1. **Написать программу, позволяющую определять коды нажимаемых клавиш и восстанавливающую исходный вид терминала (цвет, курсор) при выходе.**
2. **Написать программу, рисующую движущийся символ (при выключенном курсоре, без использования функции стирания экрана).**Для выполнения данного задания, была реализована программа, где вначале идет отключение курсора специальной последователь последовательностью, затем начинается цикл от 1 до 25 для движения по строкам, там идет отрисовка символа, ожидание 50000usec, сдвиг обратно, затирание символа и далее заново по циклу.
3. **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).**Для выполнения данного задания была написана программа, в которой сначала идет считывание аргументов запуска – скорости, направления и цвета символа. Если аргументов не 3 или они неправильные – программа завершается с ошибкой.  
     
      
     
     
     
     
   
4. **Программно реализовать команду по заданию преподавателя.**Была модифицирована программа из 4го и 3го заданий – добавлена возможность управлять символом при помощи клавиш на клавиатуре.  
     
   