МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Дисциплина: Операционные системы**

**Тема: «РАБОТА С ТЕРМИНАЛОМ И ОБОЛОЧКОЙ BASH»**

Работу выполнил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ахвердян Г.А

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль): Математическое и программное

обеспечение компьютерных технологий

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Нигодин

Краснодар

2025

После того как мы запустили agetty введем имя учетной записи, agetty запустит программу login и передаст ей введенное значение. Приложение login запросит у нас пароль, выполнит проверку учетных данных и в случае успешной аутентификации запустит оболочку командной строки Bash из-под этого пользователя.

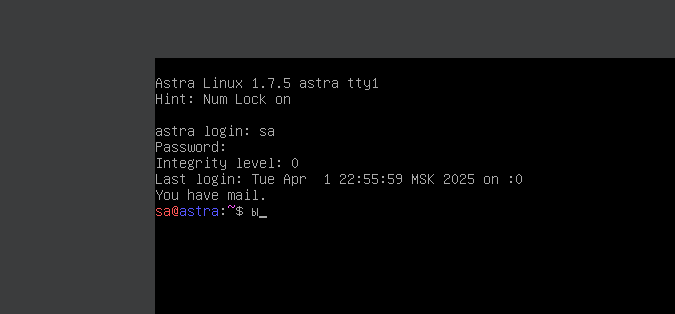


Рисунок 1 – Текстовая консоль 1

Выполним команду «nano script», которая откроет нам текстовый редактор «nano».



Рисунок 2 – Открытие текстового редактора

Проделываем тоже самое, только в tt2. Далее, с помощью команды «ls» просмотрим какие файлы и каталоги находятся в текущей директории

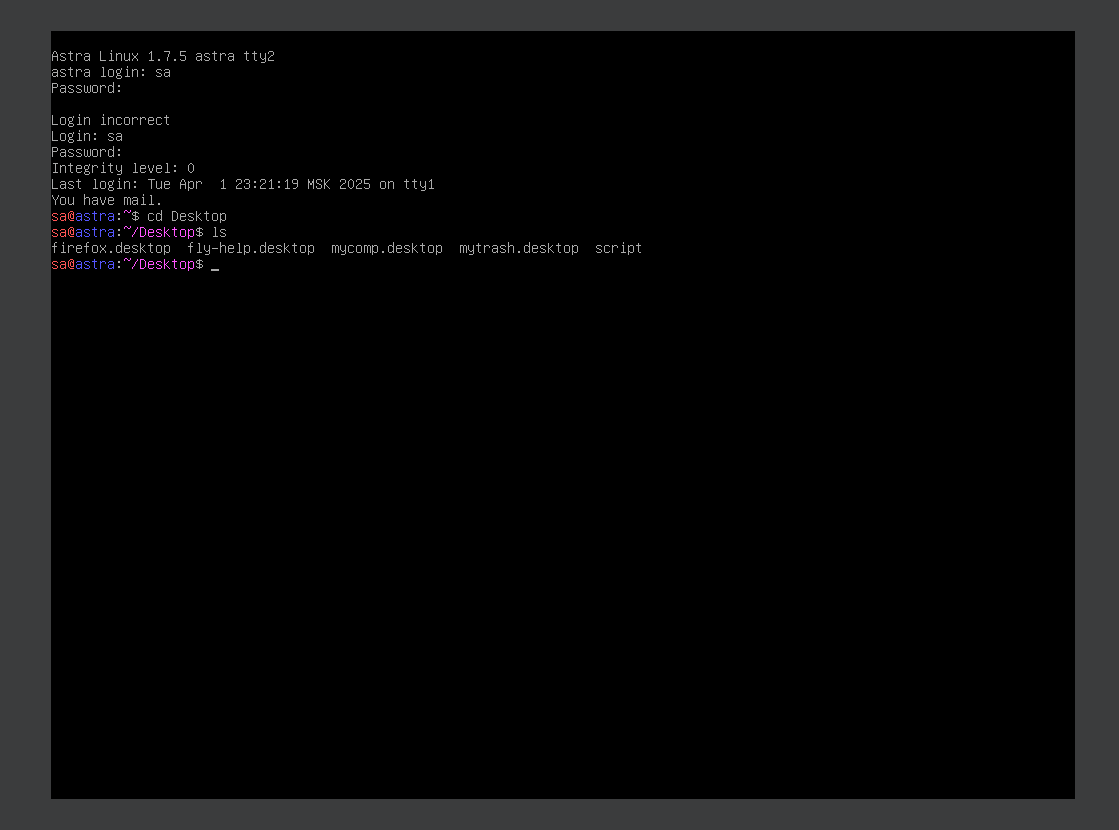


Рисунок 3 – Файлы и каталоги

Теперь запустим скрипт, выполним команду «sudo chmod u+x script», которая разрешит выполнять этот файл, и команду «./script», которая непосредственно запустит этот скрипт.

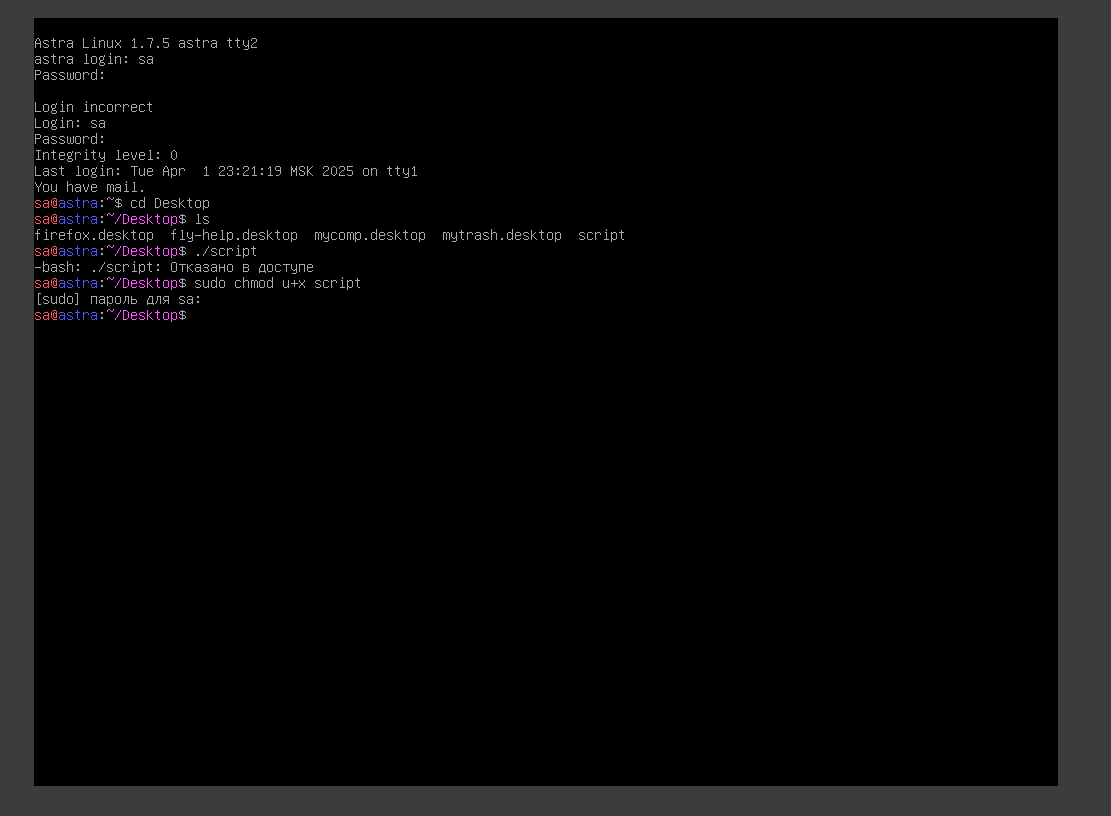




Рисунок 4 – Запуск скрипта

Переключимся на первую консоль tty1, и немного изменим наш скрипт



Рисунок 5 – Изменения скрипта

Вновь вызвав скрипт в tty2, получили новый вывод

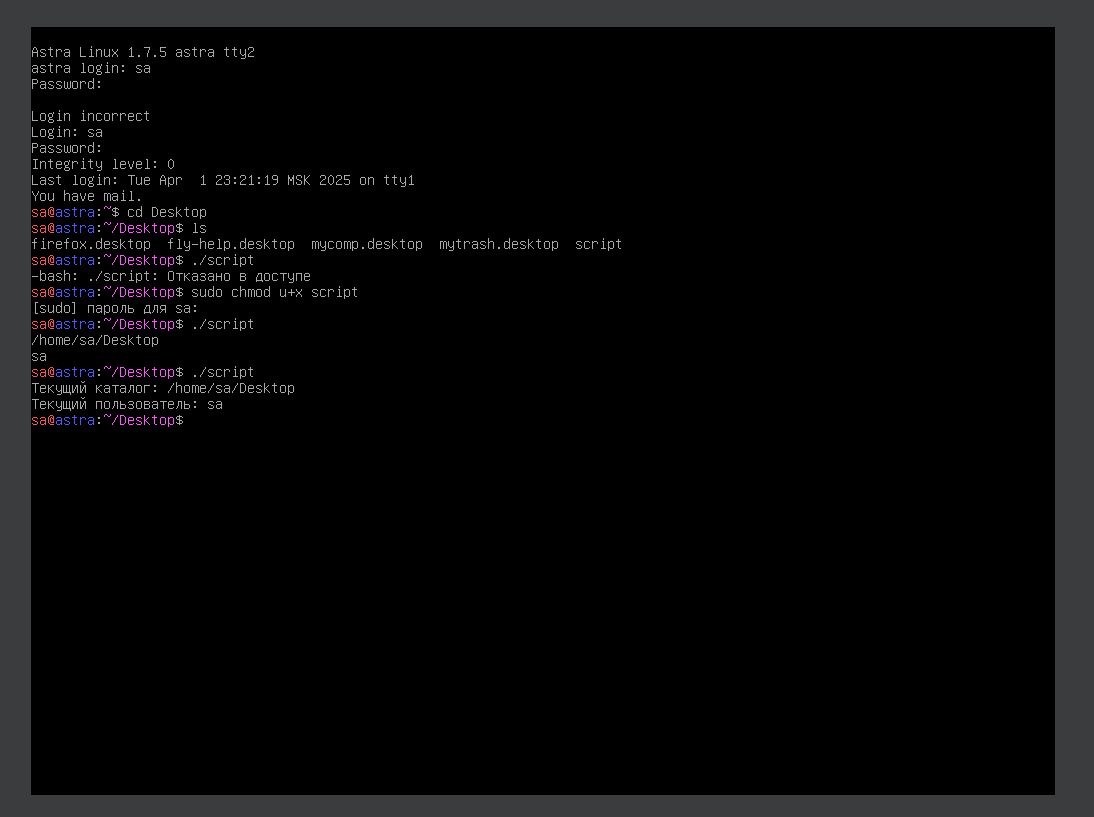


Рисунок 6 – Новый вывод

Перейдем в виртуальную консоль tty1 от имени админа и откроем текстовый редактор «nano»



Рисунок 7 – Редактор нано

Перейдем в виртуальную консоль tty2 и выполним вход от имени админа, затем выполним команду, которая выведет на экран подробную информацию о файлах tty1-tty7

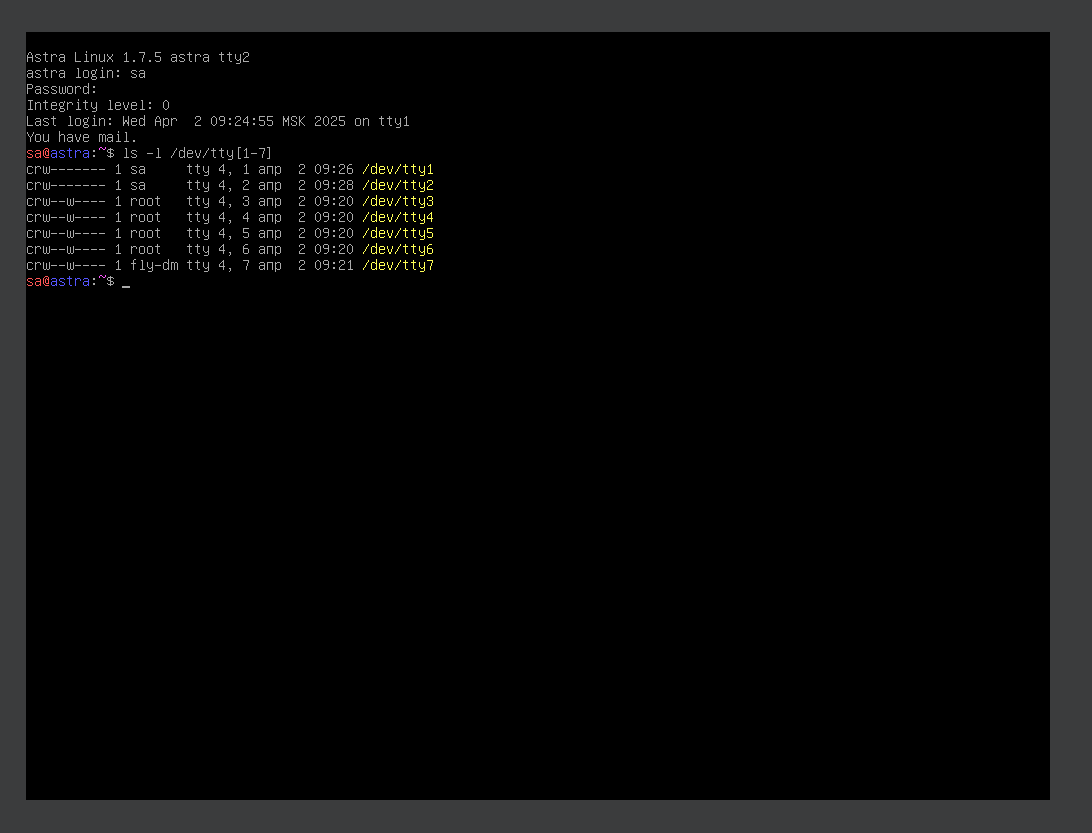


Рисунок 8 – Информация о файлах

Вводим в консоль в tty2 данный код, который позволяет передать написанное сообщение в консоль tty1



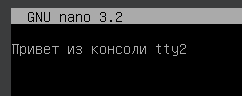


Рисунок 9 – Передача сообщений

В случае, если мы захотим переслать сообщение в tty3, нам будет отказано в доступе, в связи с тем, что мы не авторизировались в нем, и, по сути, не являемся ее владельцами

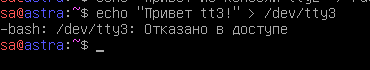


Рисунок 10 – Ошибка авторизации

Терминал fly-term. Узнаем номер псевдотерминала с которым работает наш терминал

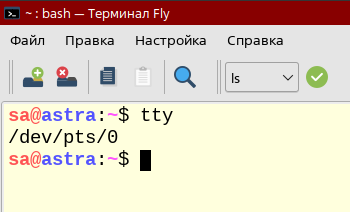


Рисунок 11 – Номер псевдотерминала

В командных оболочках мы можем выполнять команды, большинство из которых вызываются как внешние утилиты, но некоторые команды все-таки встроены в саму оболочку. Например, команда echo по умолчанию является встроенной командой, поэтому при выполнении type echo мы увидим «echo — это встроенная команда bash». Но если отключить эту встроенную команду, то для выполнения echo будет вызываться уже внешняя утилита

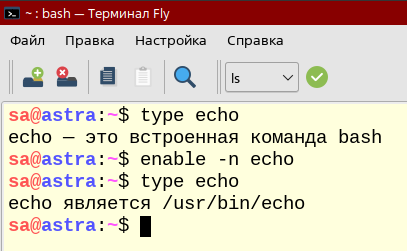


Рисунок 12 – Структура и синтаксис команд

Список встроенных команд, доступных в оболочке



Рисунок 13 – Встроенные команды

Познакомимся с наиболее часто употребляемыми командами. Команда «whoami» - команда выводит на экран текущего пользователя, от имени которого запущен терминал:



Рисунок 14 – Базовые команды BASH

«Echo» - команда для отображения строки текста



Рисунок 15 – Базовые команды BASH

«Pwd» (Print Working Directory, напечатать рабочую директорию) — команда для вывода на экран текущего каталога



Рисунок 16 – Базовые команды BASH

«Cd» (Change Directory – изменить директорию) — команда для перехода по дереву каталогов. Выполним команду cd / и перейдем в корень файловой системы, а с помощью команды cd - мы вернемся обратно в каталог, в котором были

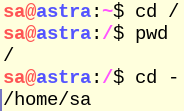


Рисунок 17 – Базовые команды BASH

Действия над переменными. Если значение переменной не содержит пробелов, то кавычки можно опустить, в противном случае они обязательны

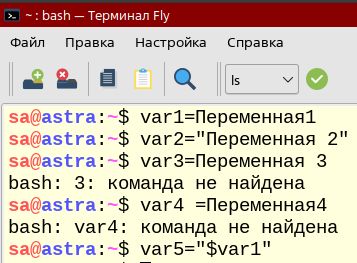


Рисунок 18 – Объявление переменной и присваивание значения к ней

Вывод значения переменной. При выводе переменной необходимо использовать не имя этой переменной, а ссылку на её значение. Если в значении переменной между словами более одного пробела, то её значение необходимо обернуть в кавычки, иначе пробелы в выводе «пропадут». Внутри одинарных кавычек, а также при экранировании символа $ не производится подстановка значений переменных

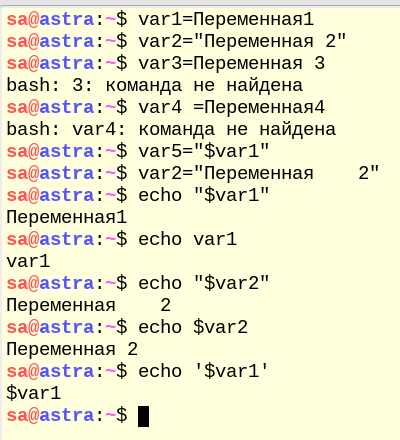


Рисунок 19 – Объявление и присваивание

Обнуление и удаление переменной. Для обнуления значения переменной достаточно присвоить ей пустое значение. Для удаления переменной предназначена команда «unset <имя переменной>»

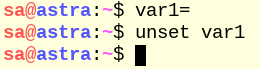


Рисунок 20 – Обнуление и удаление

Превращение пользовательской переменной в переменную среды. Для превращения пользовательской переменной, доступной только в пределах процесса текущей оболочки, в переменную среды, которая доступна и в дочерних процессах, необходимо воспользоваться командой export. Если мы теперь откроем дочерний процесс bash и попытаемся вывести значения переменных, то мы увидим, что переменная var2 нам доступна, а переменная var3 – нет

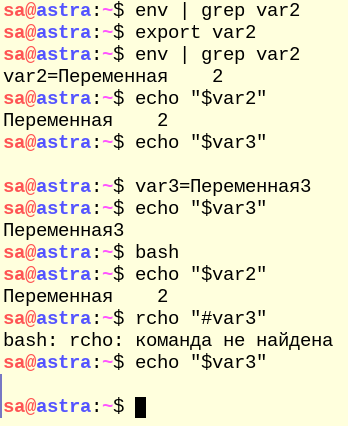


Рисунок 21 – Превращение в переменную среды

Для создания псевдонима существует команда: «alias имя\_псевдонима="значение"». Созданный таким образом псевдоним будет работать только в текущей консоли. Чтобы сделать его постоянным, необходимо добавить эту команду в файл ~/.bash\_profile или ~/.bashrc.

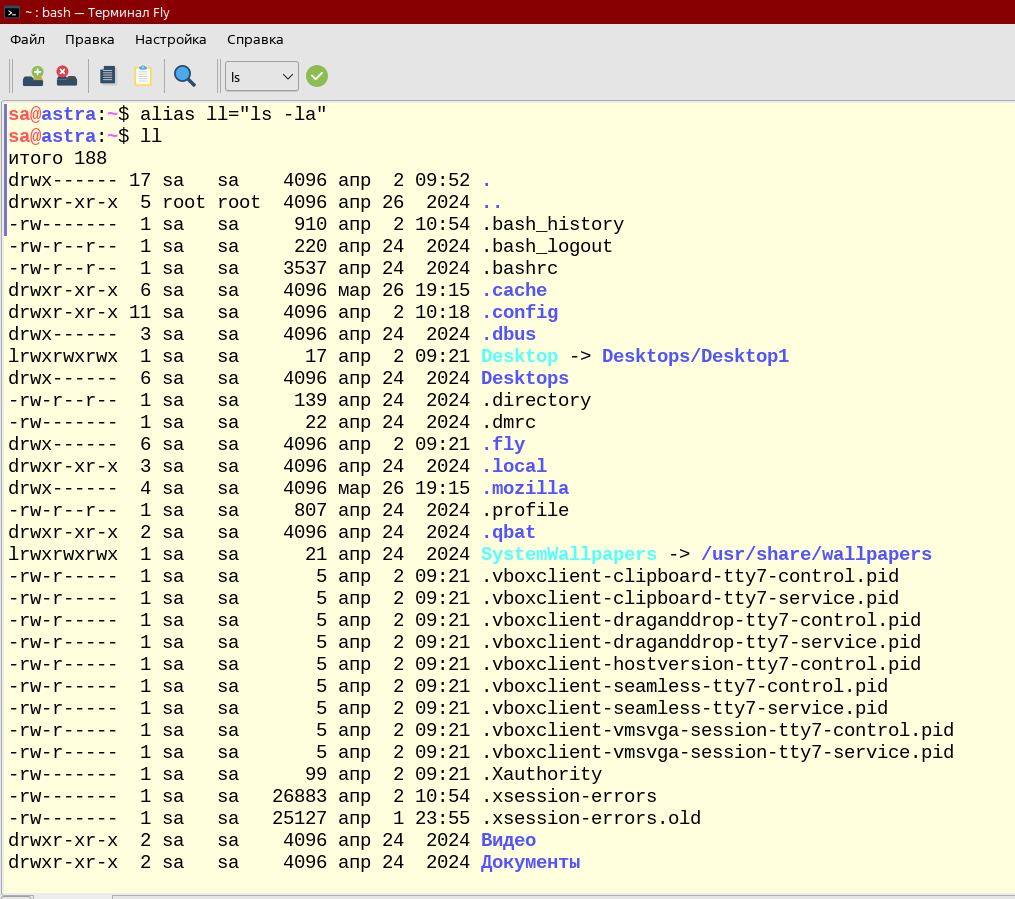


Рисунок 22 – Создание псевдонимов

Чтобы посмотреть все псевдонимы, необходимо использовать ключ -p: «alias –p». Для удаления псевдонима существует специальная команда «unalias»

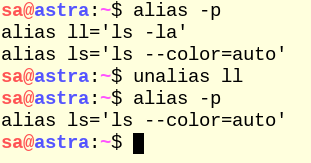


Рисунок 23 – Просмотр и удаление

Для того что бы вывести историю используемых команд, необходимо в терминале прописать команду «history». Для того, что бы вывести, условно, последние 4 команды, добавим к history четыре – «history4»

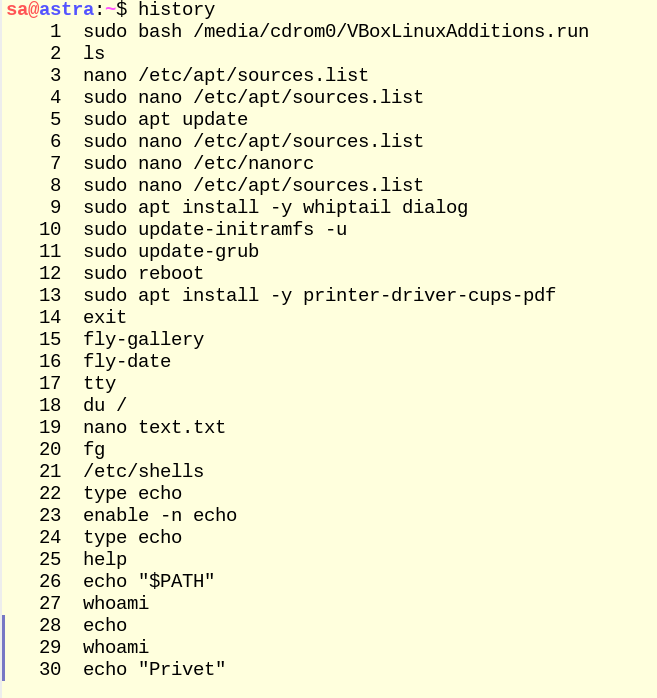


Рисунок 24 – Просмотр команд и передвижение по ней

Команды в истории можно искать по совпадению с введенной строкой. Ctrl + R - включение режима поиска. После включения режима поиска ищем текст, который должен содержаться в искомой команде, и будет найдено первое совпадение от конца к началу истории. Для перехода к следующему совпадению нажимаем это же сочетание Ctrl + R . Чтобы выйти из режима поиска, используем Ctrl + C .



Рисунок 25 – Поиск команды в истории

С помощью символа « ! » можно выполнить команду из истории по её номеру. А специальным выражением !! можно быстро вызвать предыдущую команду еще раз

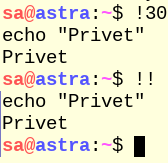
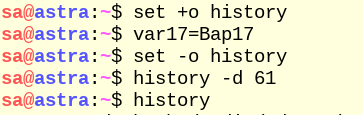


Рисунок 26 – Выполнение команд из истории

Если нам потребуется исключить из занесения в историю несколько команд, то мы можем временно отключить журналирование. Если же команда уже попала в историю, то мы можете ее удалить оттуда. Для временного отключения журналирования команд необходимо выполнить команду «set +o history, а после set -o history». Если конфиденциальные данные уже попали в историю команд, то их можно удалить. Можно выборочно удалять команды или очистить всю историю сразу. «history -d <номер\_команды>» удаление команды из истории по её номеру, «history -c» - удаление всей истории



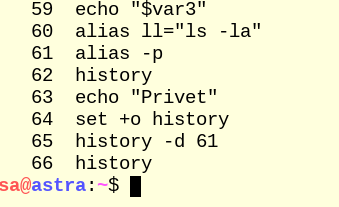


Рисунок 27 – Отчистка истории команд

Для того, что бы запустить несколько команд, необходимо использовать «;». Так же в различных ситуациях различные логические символы

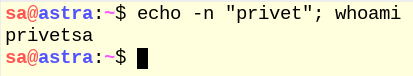


Рисунок 28 – Запуск нескольких команд за раз

Перенаправление результата выполнения команды в файл. команда cat, копирует стандартный поток ввода в стандартный поток вывода. Так же было использовано перенаправление результата выполнения команды в конец файла.

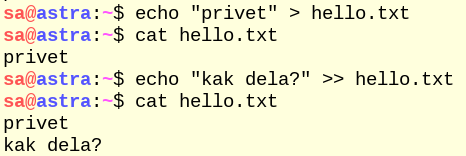


Рисунок 29 – Стандартные потоки и перенаправление потока ввода\вывода

«ls» – команда выводящая содержимое текущей директории. Если мы хотим вывести на экран только файлы (или каталоги), содержащие в имени «hello», мы можем использовать команду grep, фильтрующую строки по выбранному шаблону, передав в неё результат выполнения команды ls через конвейер.

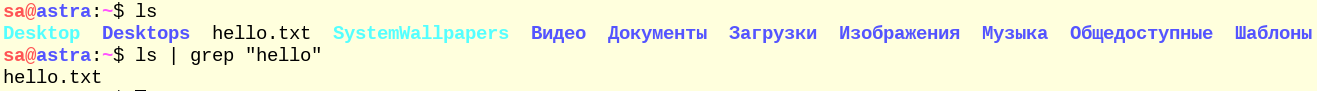
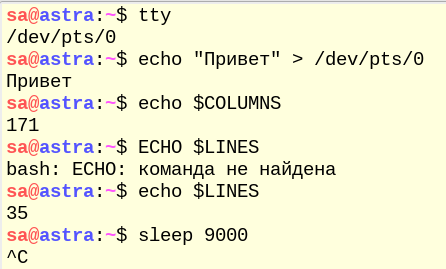


Рисунок 29 – Конвейер

**Практика 1**. **В пункте 6**, в ответ на вопрос почему Ctrl + C не закрывает mc, предположительно, данная комбинация внутри данного окна выполняет другой спектр задач. Для выхода я использовал F10

1-4)



5)

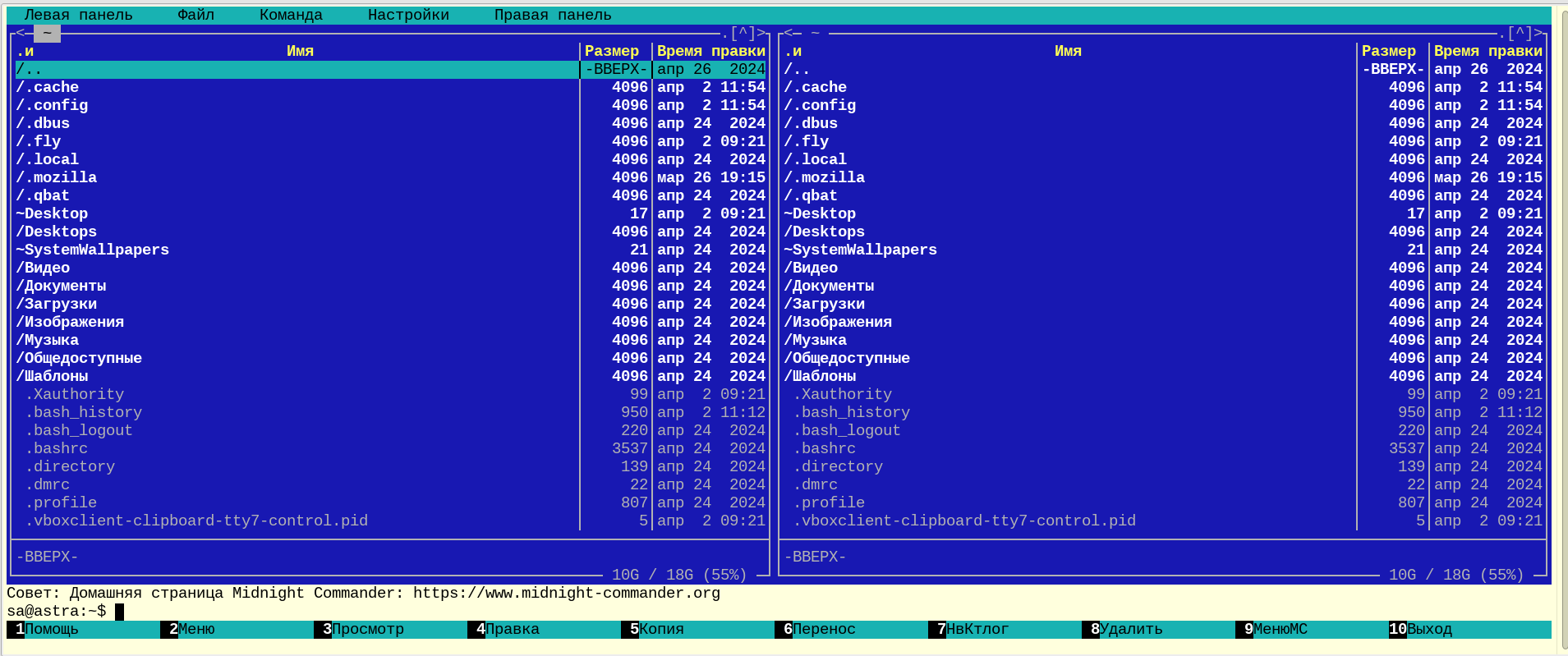


Рисунок 30 – Практика 1

**Практика 2**. **Пункт 5:**

Изменился ли файл после выполнения cat?

Команда cat /etc/hostname предназначена только для просмотра содержимого файла hostname и никак не влияет на сам файл, то есть он не будет изменен.

Куда команда cat вывела содержимое файла?

Вывод команды cat отображается в терминале, так как она по умолчанию направляет содержимое файла на стандартный вывод который связан с экраном.

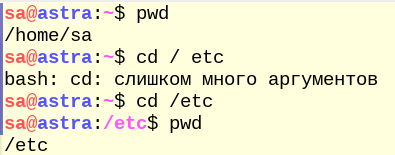
Безопасна ли команда cat?

Сама по себе команда cat не представляет опасности, поскольку её функция сводится к чтению и отображению содержимого файла. Однако, степень безопасности зависит от того, какой именно файл мы просматриваем.

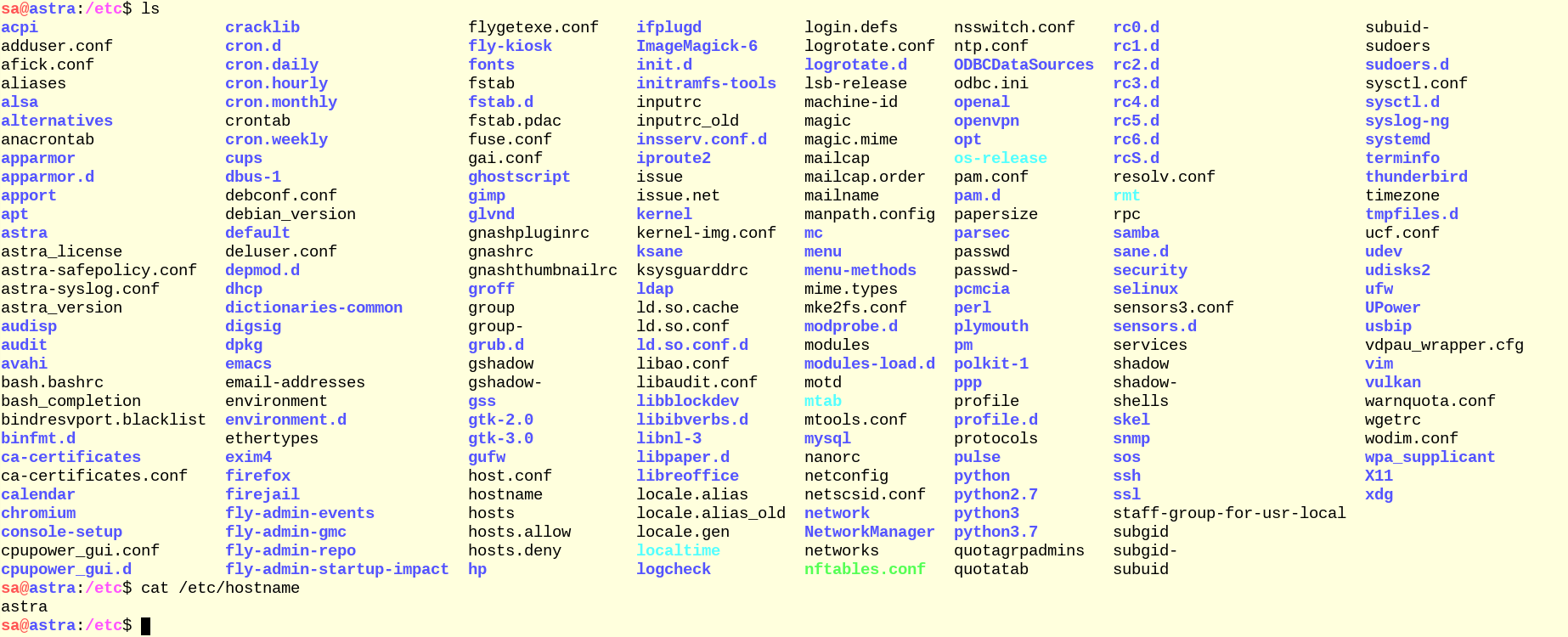
**Пункт 6:** Файл hostname.old будет создан в домашней директории пользователя, выполнившего команду. Если команда выполнена обычным пользователем, файл сохранится в его домашнем каталоге

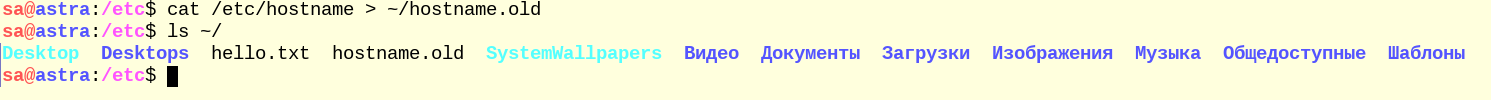
**Пункт 8:** Красное подчеркивание и сообщение "File '/etc/hostname' is unwritable" указывают на то, что вы пытаетесь изменить файл /etc/hostname, не имея необходимых прав доступа. Это означает, что редактор запущен без прав суперпользователя (например, с использованием sudo), и, следовательно, запись в этот файл запрещена.

1-3)



4-5)



6)

7)



8)

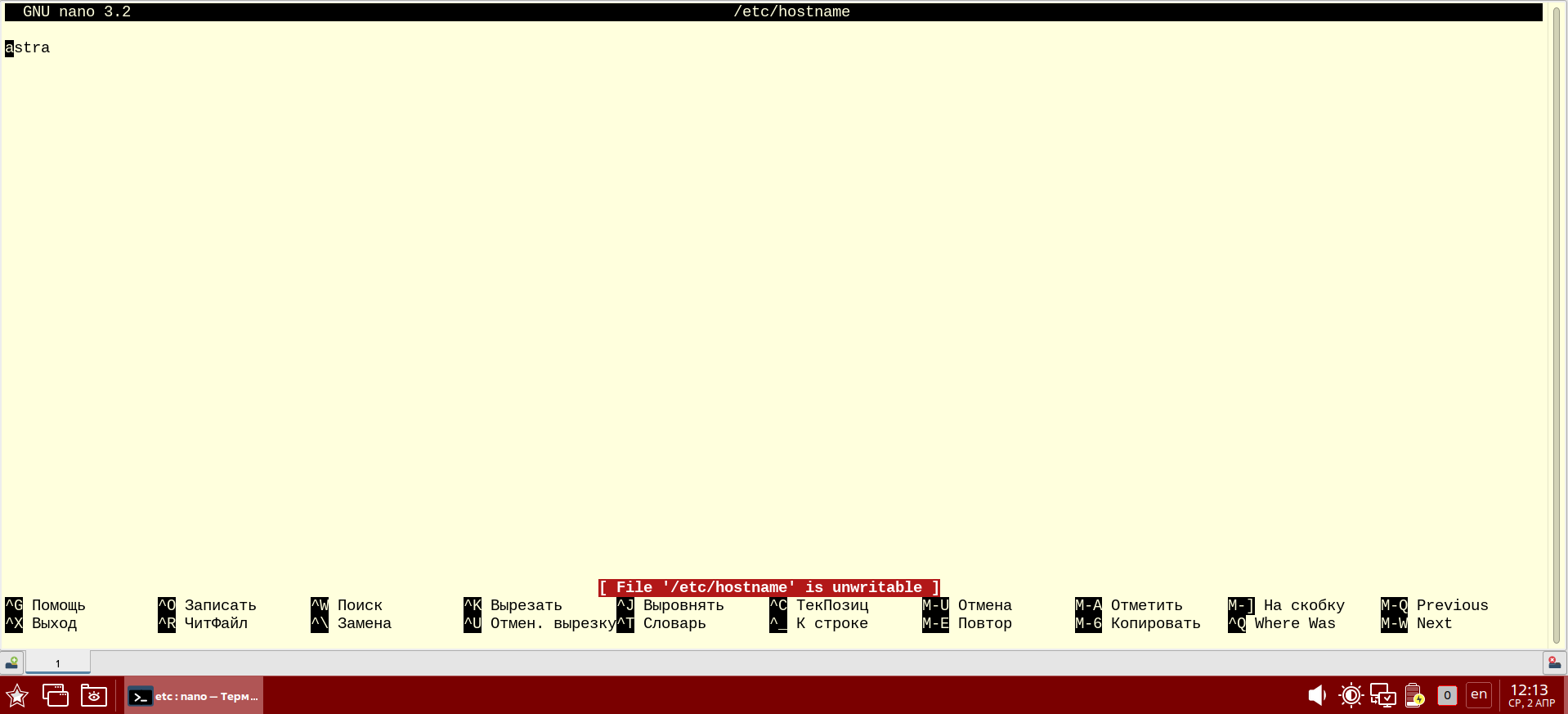


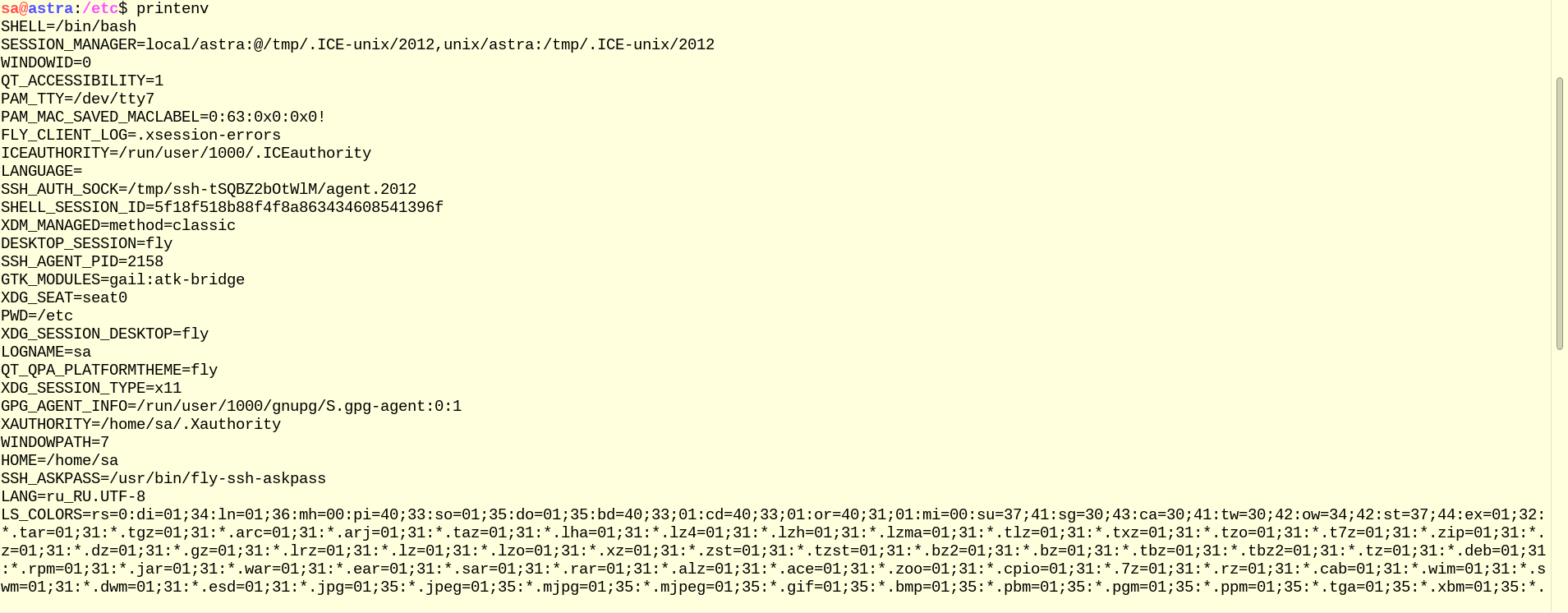
Рисунок 31 – Практика 2

**Практика 3. Пункт 5:** Переменная $PATH сообщает системе, в каких каталогах искать исполняемые файлы команд. Даже если такие команды, как reboot, установлены в системе, они могут находиться в каталогах, доступ к которым ограничен пользователем root.

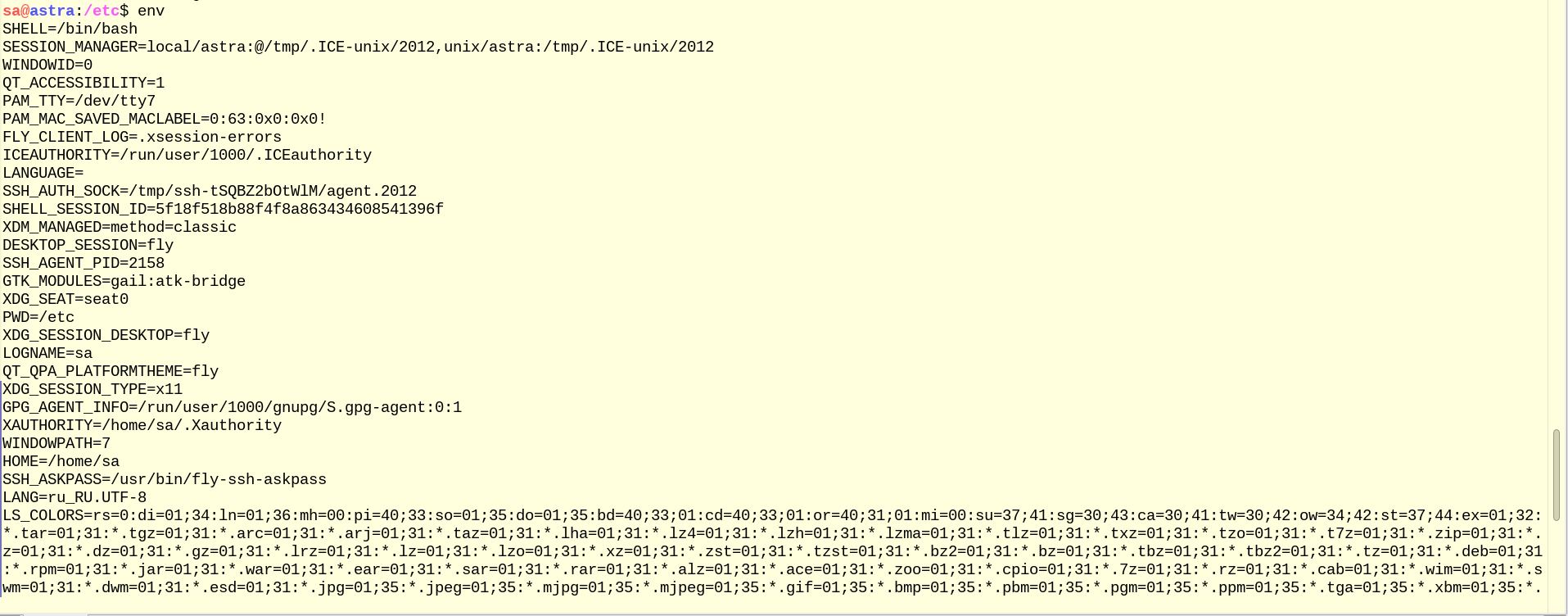
1)



2)



3)



4-5)

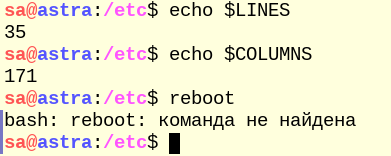


Рисунок 32 – Практика 3

**Вопросы:**

**1. Как называется устройство, которое может отправлять команды ЭВМ и выводит на экран полученный результат?**

Это называется терминалом. Хотя исторически это было отдельное устройство, сейчас чаще всего это эмулятор, работающий в графическом интерфейсе.

**2. Какая папка отвечает за конфигурационные файлы?**

«/etc»

**3. Какая управляющая последовательность завершает операцию?**

Как правило, используется сочетание клавиш Ctrl+C.

**4. Какая команда выводит список файлов и каталогов текущей директории?**

«Ls»

**5. Какими текстовыми редакторами можно редактировать файл?**

«nano»

**6. Какой командой можно получить справку на любую команду?**

**7. Какой командой можно перенаправить стандартный вывод в файл hosts.bak?**

Чтобы перенаправить вывод, используйте команду <команда> > hosts.bak. Например, чтобы перенаправить вывод cat /etc/hosts в файл hosts.bak, введите cat /etc/hosts > hosts.bak. Имейте в виду, что если hosts.bak уже существует, он будет перезаписан. Для добавления вывода в конец файла без перезаписи используйте <команда> >> hosts.bak.

**8. В какой переменной хранится список каталогов для запуска исполняемых файлов?**

В переменной $PATH.

**9. Какой поток данных передается по конвейеру?**

По конвейеру передается стандартный вывод (stdout) от первой команды, который используется в качестве стандартного ввода (stdin) для следующей команды.

**10. Какая команда отображает историю команд?**

«history»

**11. Какой файл содержит профиль текущего пользователя?**

Это зависит от используемой оболочки. В оболочке Bash, с которой мы работали, это могут быть файлы ~/.bashrc, ~/.bash\_profile, ~/.bash\_login или ~/.profile. Оболочка последовательно считывает эти файлы при входе пользователя в систему.

**Вывод:**

Я узнал, чем терминал отличается от консоли, как работает оболочка командной строки и узнал о различных тонкостях.