

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)**

Факультет Информатика и вычислительная техника

(наименование факультета)

Кафедра Кибербезопасность информационных систем

(наименование кафедры)

**ОТЧЕТ**

по лабораторным работам по предмету: Компьютерные сети

Выполнил:

ст. гр. ВКБ31 Котелевец К.А.

Ростов-на-Дону

2025

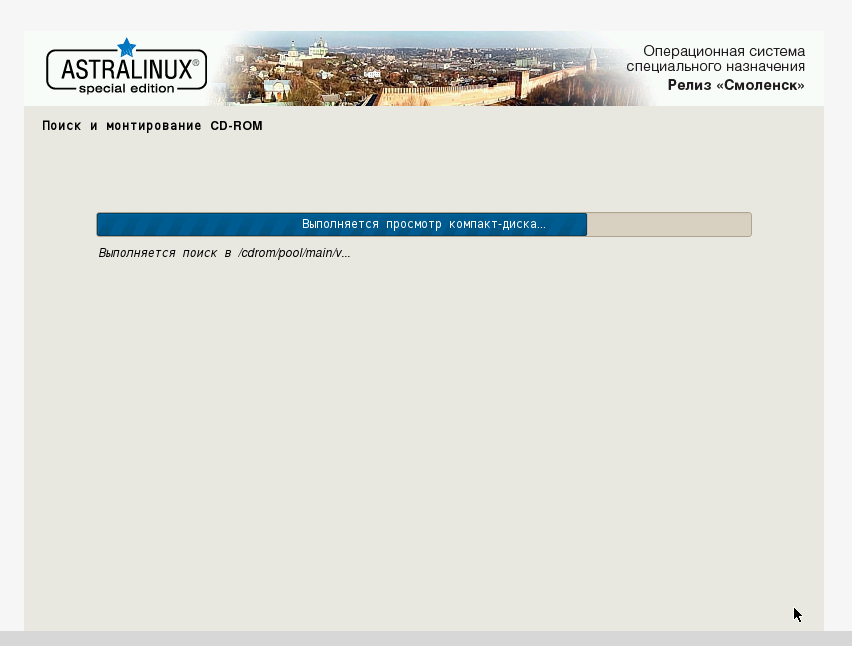
Лабораторная работа 1

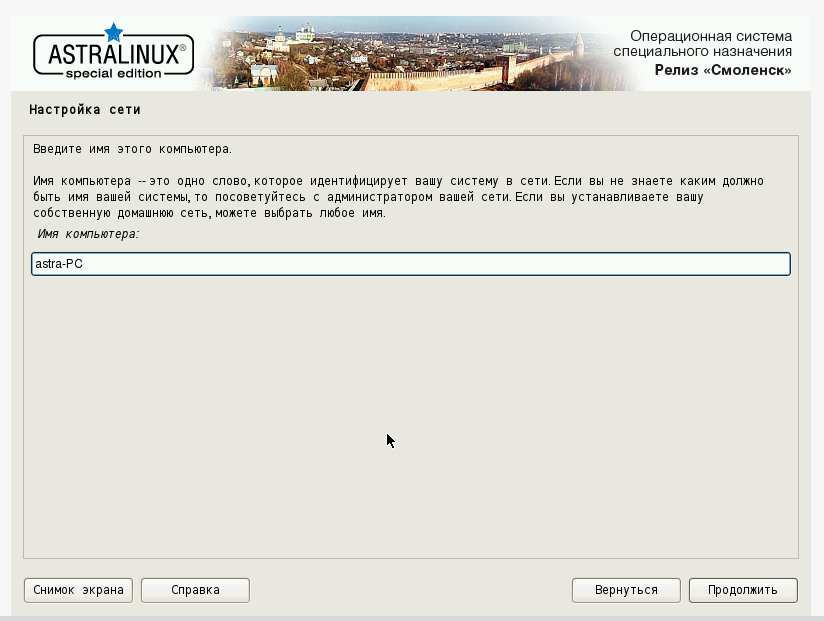
Установка и настройка операционной системы AstraLinuxSE

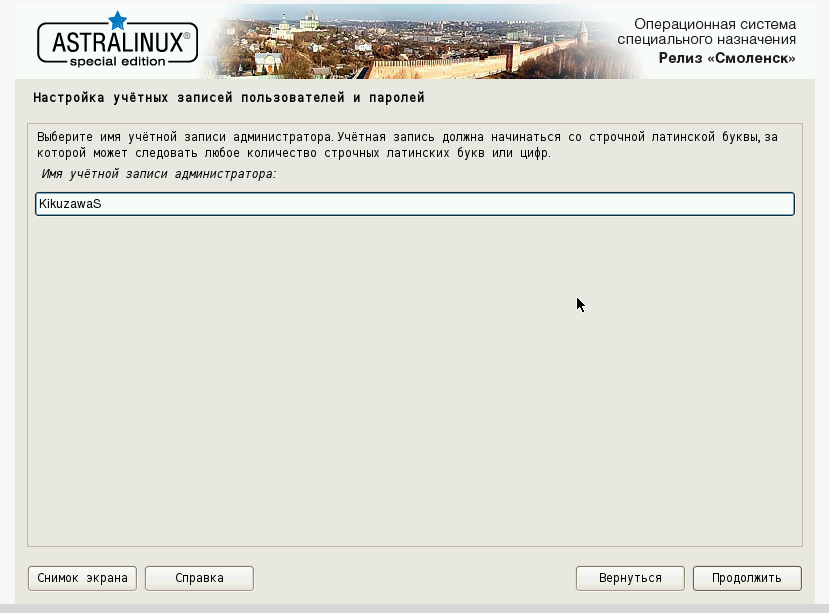
**Цель работы –** изучение требований к целевому компьютеру и подготовка к установке, установка ОС, настройка дополнительных параметров в Astra Linux SE.

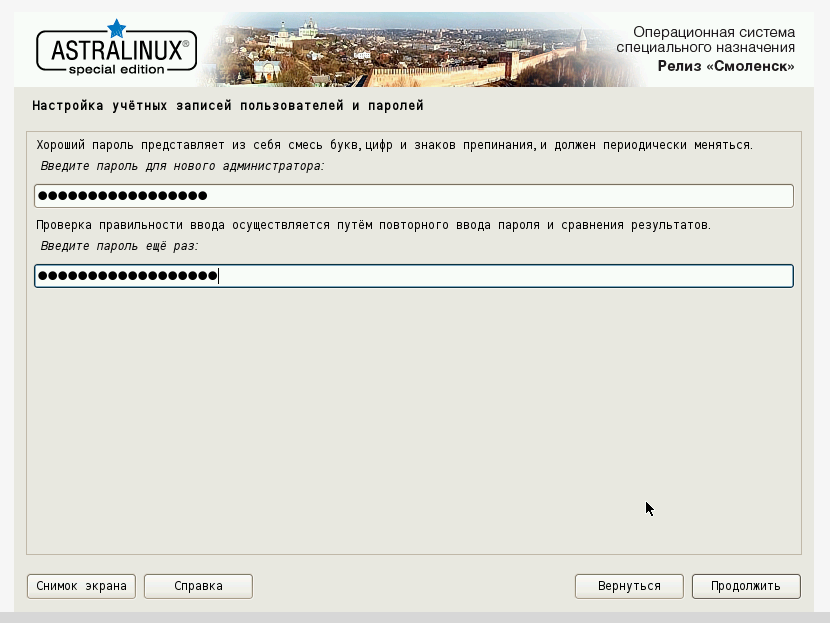
**Ход работы**

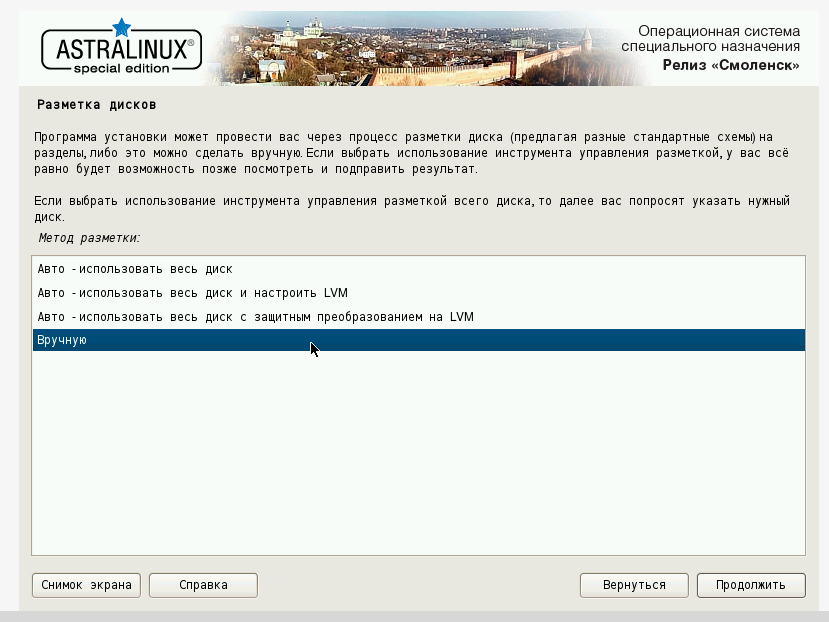
1. Выполнить поэтапную установку Astra Linux SE с **ручной разметкой накопителя!** (уметь рассказать процесс) согласно п. 2 Теоретических сведений.

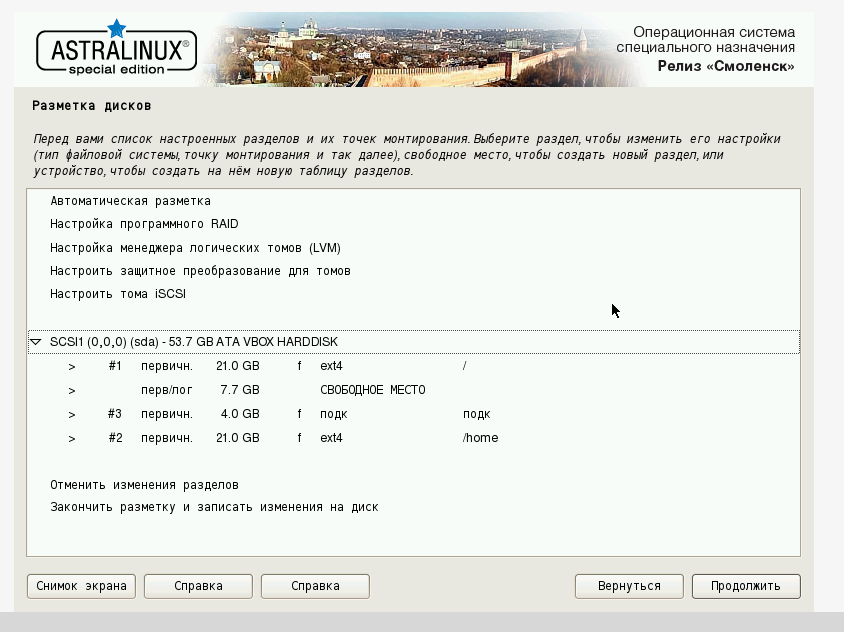




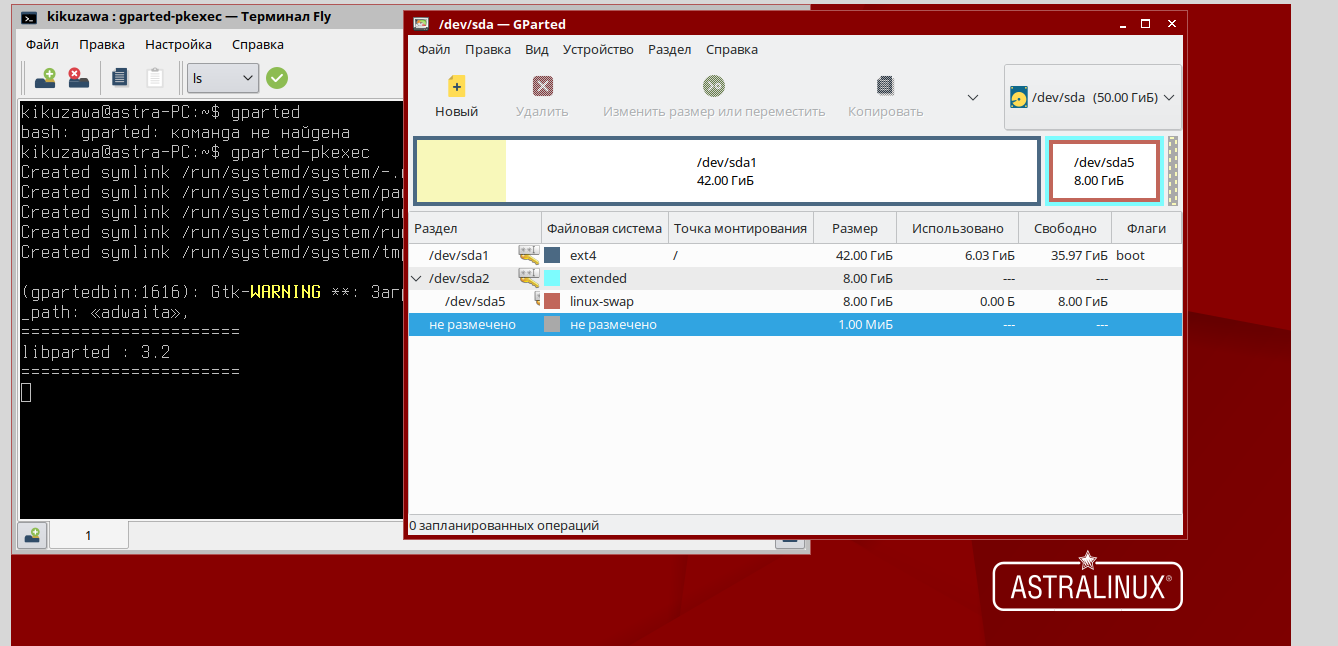








5. Установить и запустить утилиту GParted (сделать скриншот).



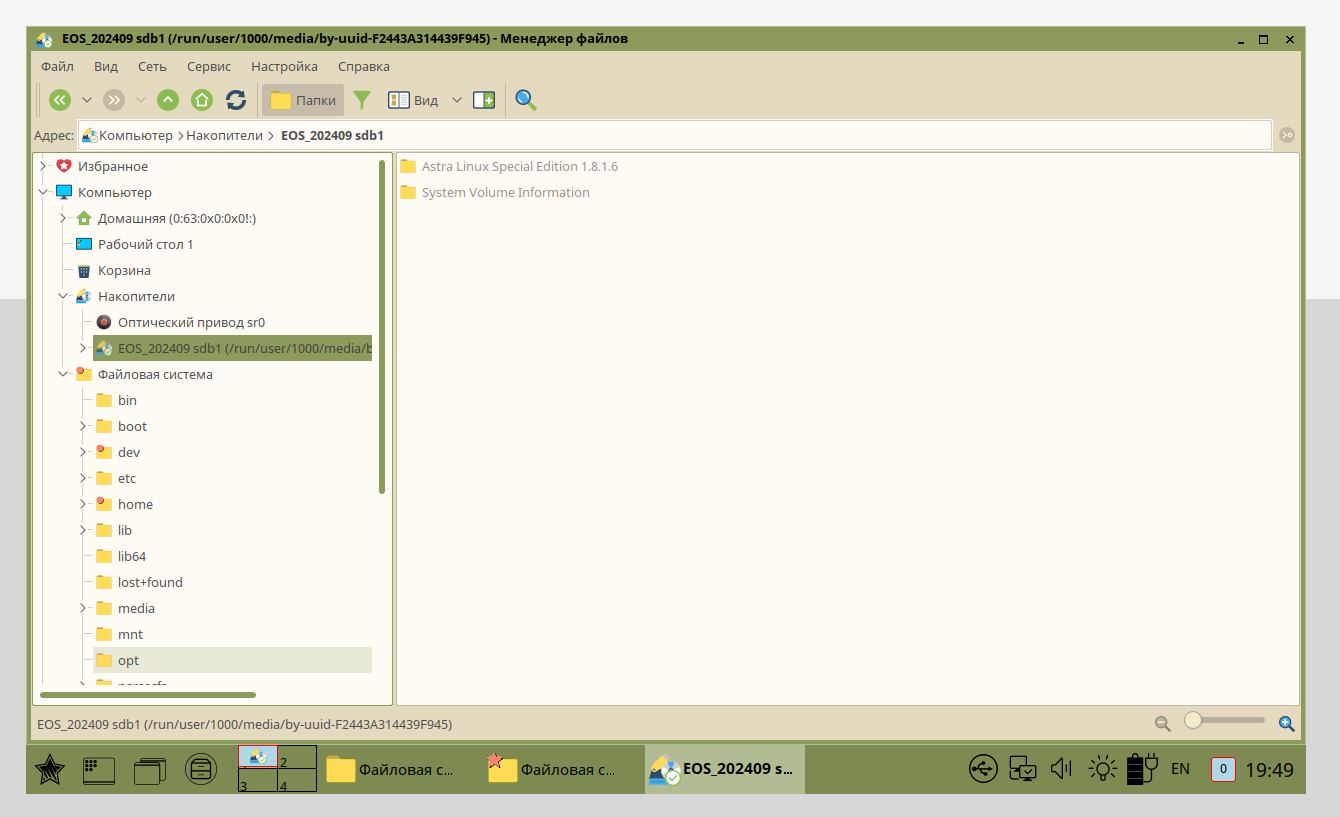
6. Осуществить вход в систему (рассказать типы сессий, меню, все уровни конфидециальности и целостности, категории)

7. Выполнить пользовательские настройки интерфейса работы.

Я ничего не буду менять, меня и так всё устраивает. Поставил для галочки зеленую тему.

8. Показать навыки монтирования/размонтирования сьемных носителей.

На рисунке я показал монтирование флешки



9. Ознакомиться с работой LibreOffice, создать и оформить документы приложений LibreOffice.

Я не буду вставлять сюда скриншоты, я уже этот отчет сделал в LibreOffice

Лабораторная работа №2

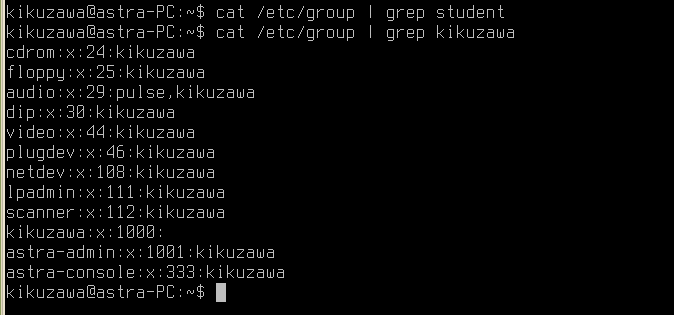
Конвейеры И перенаправление ввода-вывода, архивирование и сжатие

Цель работы – научиться работать с командами поиска и архивами. Рассмотреть практические примеры применения конвейеров и использования перенаправления ввода-вывода. Научиться работать с архивами. Изучить утилиты для сжатия и архивирования файлов. Научиться применять данные утилиты совместно с другими командами.

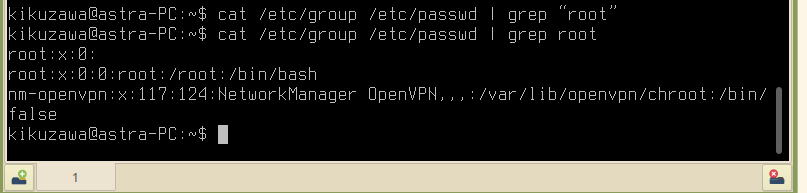
Ход работы:

Задание 1. Найти строку, содержащую слово student в файле /etc/group

Если я введу в терминал команду cat /etc/group | grep “student”, то она ничего не покажет, т.к. просто нет этого слова, однако можно вставить мой никнейм



Задание 2. Отоброзить все строки, содержащие слово root из файлов /etc/passwd и /etc/group



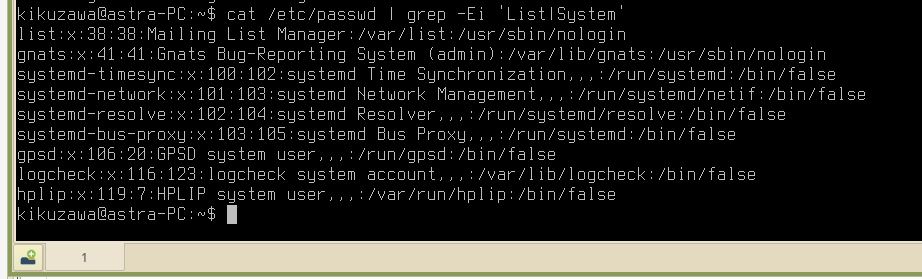
Задание 3. Подсчитать количество строк, в которых содержится пользователь (любой, кроме root) в файле /etc/group



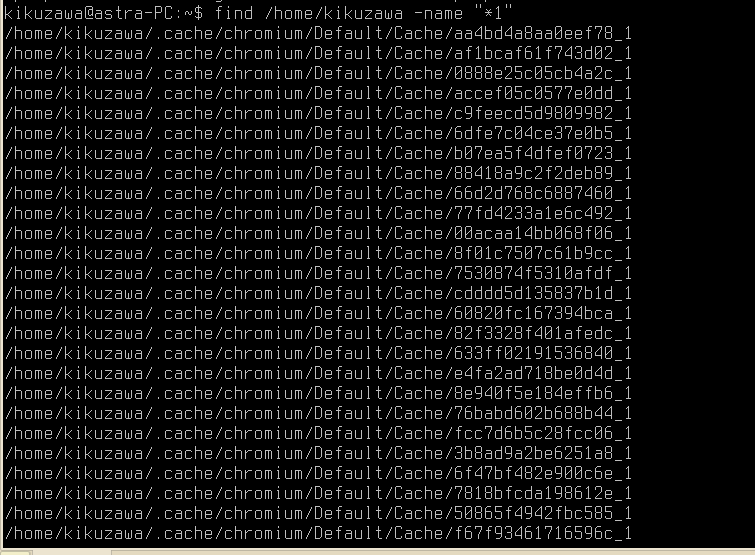
Задание 4. Подсчитать количество строк, в которых не содержатся пользователь в том же файле



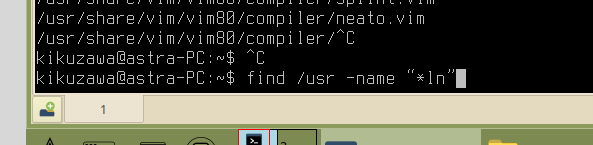
Задание 5. Используя команду grep, отобразить все строки, содержащие слово List или слово System в файле /etc/passwd



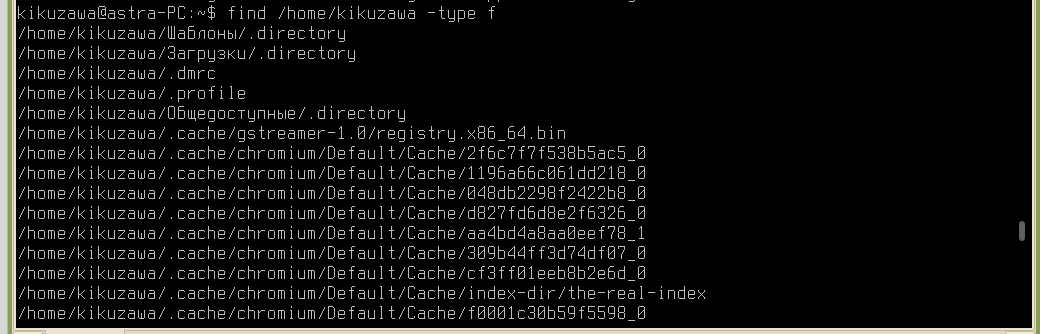
Задание 6. Найти в домашней директории пользователя все файлы и директории, имя которых заканчивается на 1



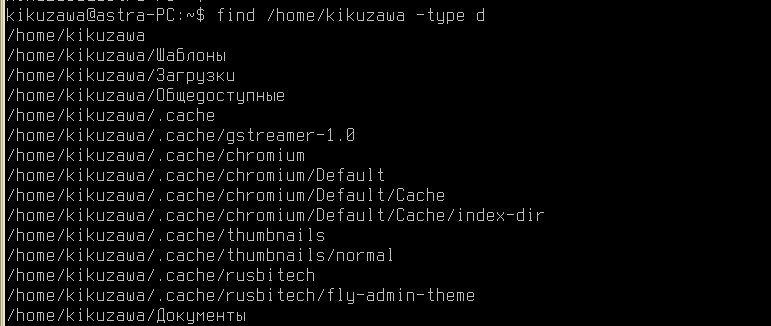
Задание 7. Найдите в директории /usr все файлы и директории ,имя который заканчивается на 1n



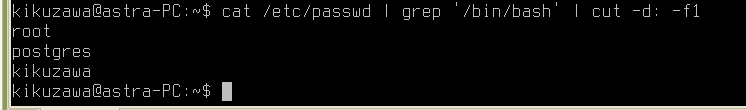
Задание 8. Найдите в домашней директории пользователя student(любого, кроме root) только файлы



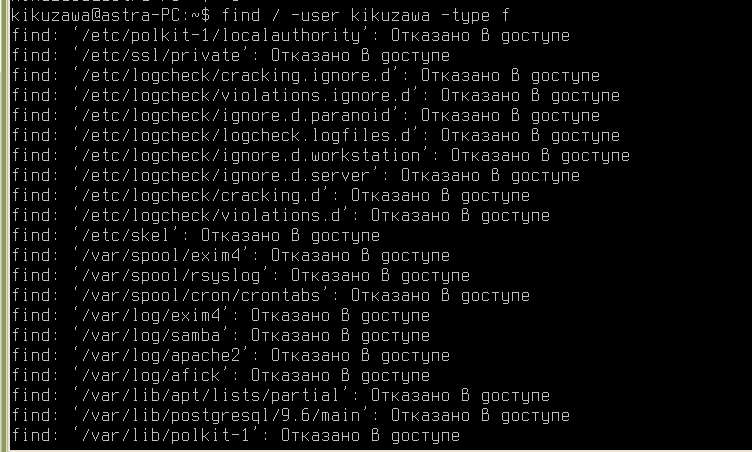
Задание 9. Найдите в домашней директории пользователя только директории



Задание 10. У кого из пользователей в качестве оболочки используется bash

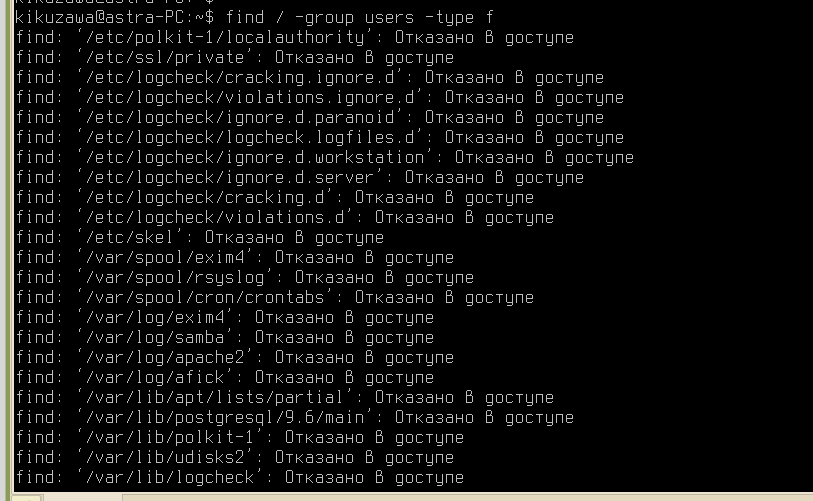


Задание 11. Найдите все файлы, принадлежащие пользователю



Задание 12. Найдите все файлы, принадлежащие любой другой группе,

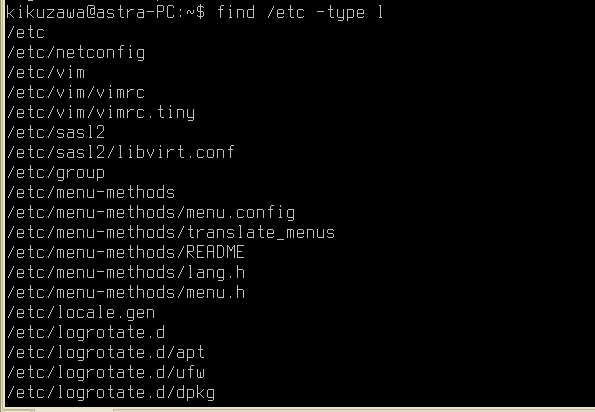
например ftp



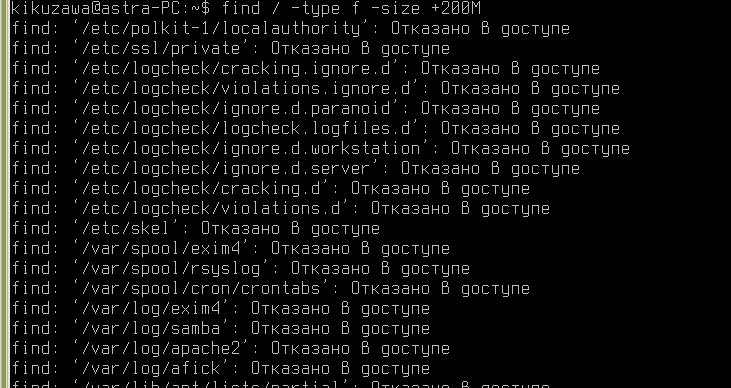
Задание 13. Найдите все файлы несуществующих пользователей или групп.

Задание 14. Найдите все файлы, являющиеся символическими ссылками в

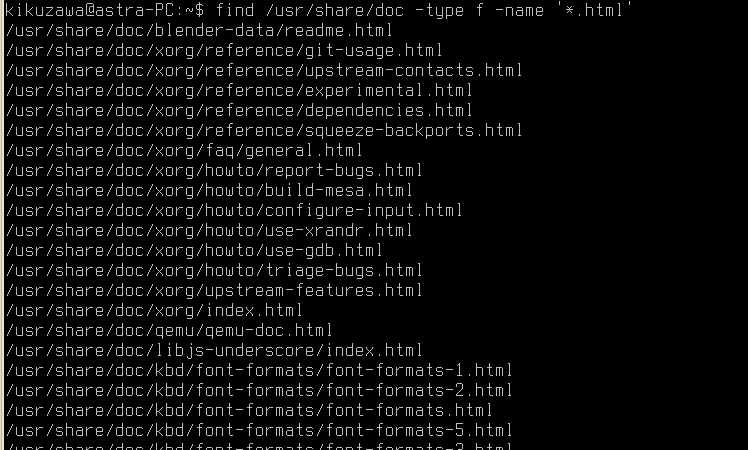
директории /etc



Задание 15. Найдите все файлы в системе размером более 200Mб

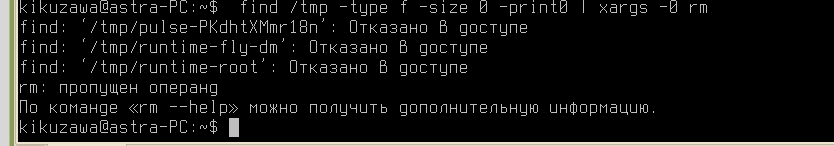


Задание 16. Найдите все файлы с расширением .html в директории /usr/share/doc

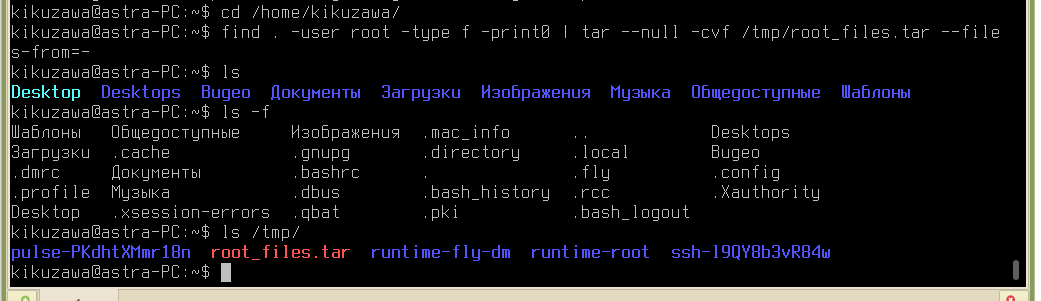


Задание 17. Найдите в директории /tmp файлы нулевого размера, а затем

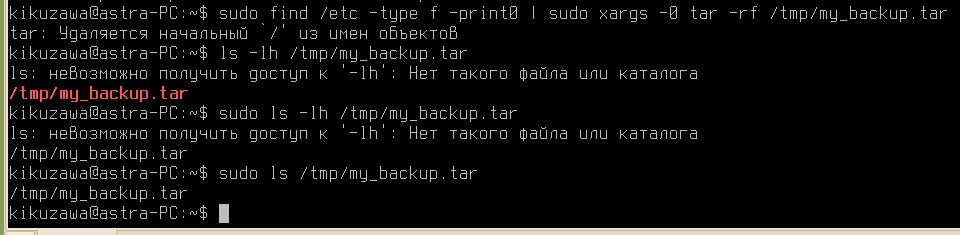
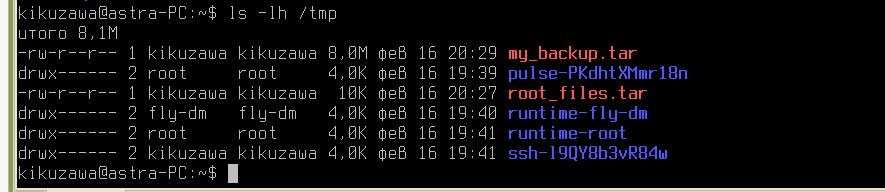
удалите не нужные из них.



Задание 18. . Используйте команду find, чтобы определить местонахождение всех файлов, принадлежащих пользователю root в домашней директории пользователя student, а затем создайте архив с помощью команды tar. Архив должен находиться в каталоге /tmp

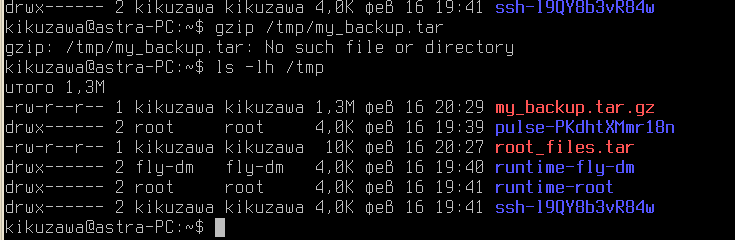


Задание 19. Добавьте содержимое каталога /etc в архив в директорию /tmp. Посмотрите размер полученных файлов (сделать скриншот).

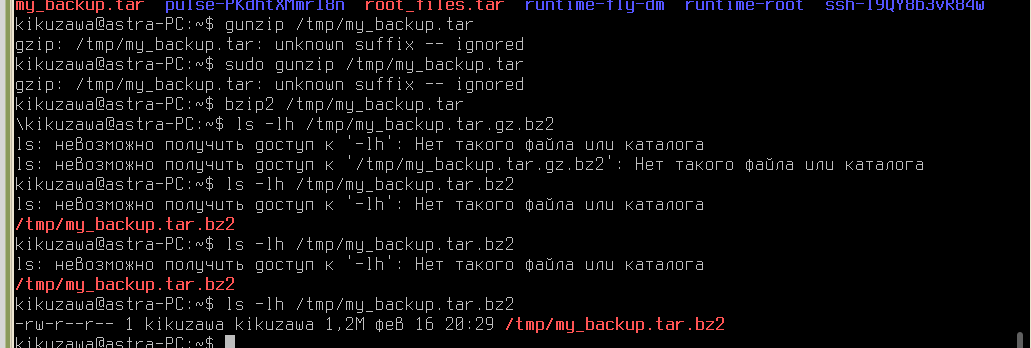
 

Задание 20. Используйте gzip для сжатия ваших архивов и снова посмотрите

получившийся размер файлов.



Задание 21. Отмените сжатие существующих файлов и сожмите их снова с помощью утилиты bzip2, и снова посмотрите размер (сделать скриншот).



Лабораторная работа №3

Управление процессами и планирование

Цель работы – научиться управлять процессами ОС, передавать сигналы процессам. Разобраться в текущем состоянии ОС, насколько система загружена, какие процессы имеют приоритеты. Научиться управлять приоритетами процессов. Рассмотреть разные возможности планирования задач. Научится планировать задания, требующие периодичного выполнения (например, создание бэкапа системы), а также задачи, которые запускаются только один раз в заранее установленное время.

Задание 1.

Из консоли пользователя командой `ps -efl | more` выведите расширенный поэкранный список исполняемых процессов (перечень параметров для расширенного вывода информации можно уточнить с помощью электронного справочника `man`).

Разберитесь с выводимой информацией и объясните преподавателю.

Определите и осуществите вывод процессов согласно выбору преподавателя:

- по типу: системные, демоны, пользовательские (тип процесса определяется по косвенным признакам, в частности, по имени);

- по состоянию S: (исполняющиеся, ожидающие записи на диск, ожидающие событий, приостановленные, зомби);

- по текущему динамическому приоритету `PRI` (наименьшее значение у высокоприоритетных процессов);

- по относительному приоритету `N1`.

Системные процессы (root):

ps -efl | awk 'NR==1{print $0}; $3=="root" {print $0}'

Демоны:

ps -efl | awk 'NR==1{print $0}; $3=="daemon" {print $0}'

Пользовательские процессы:

ps -efl | awk 'NR==1{print $0}; $3=="kikuzawa" {print $0}'

Пример для процессов в состоянии ожидания (D):

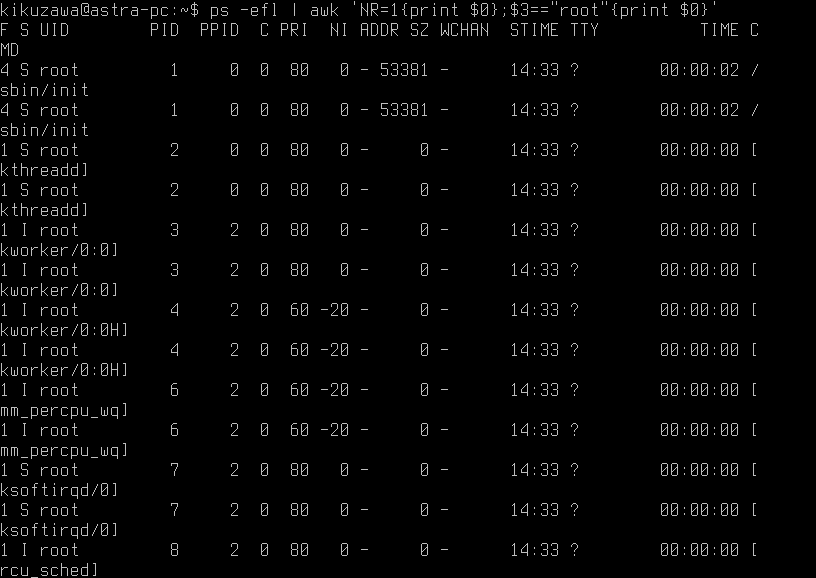
ps -efl | awk 'NR==1{print $0}; $2=="D" {print $0}'

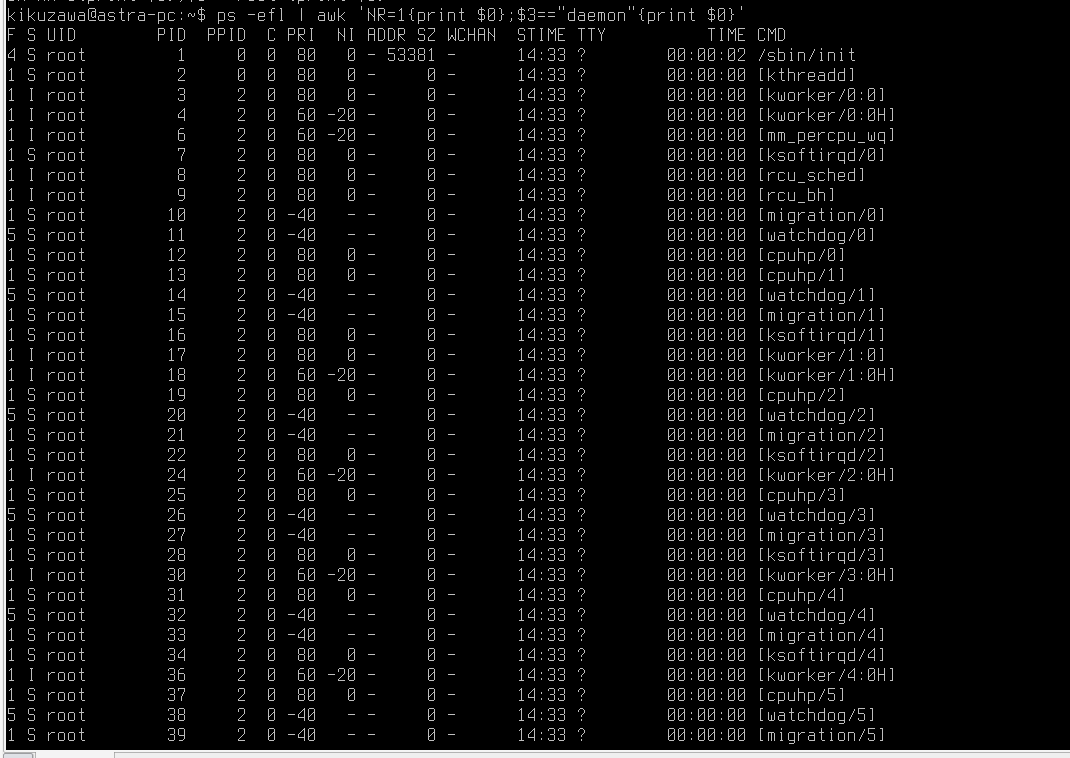
Теперь сделаем вывод по текущему динамическому приоритету PRI.

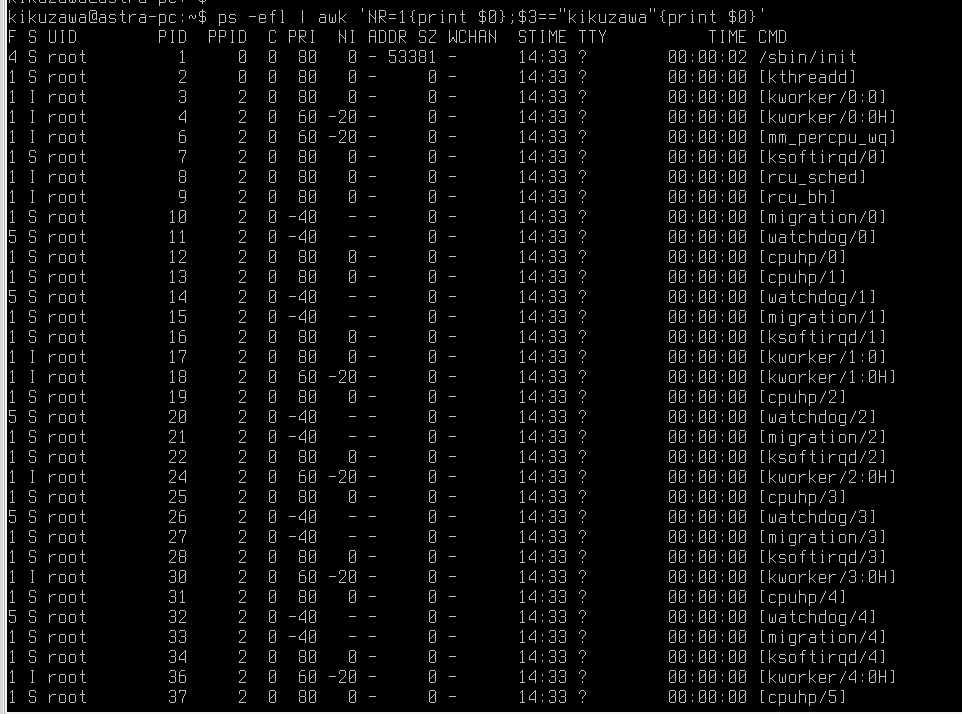
ps -l | sort -k 4

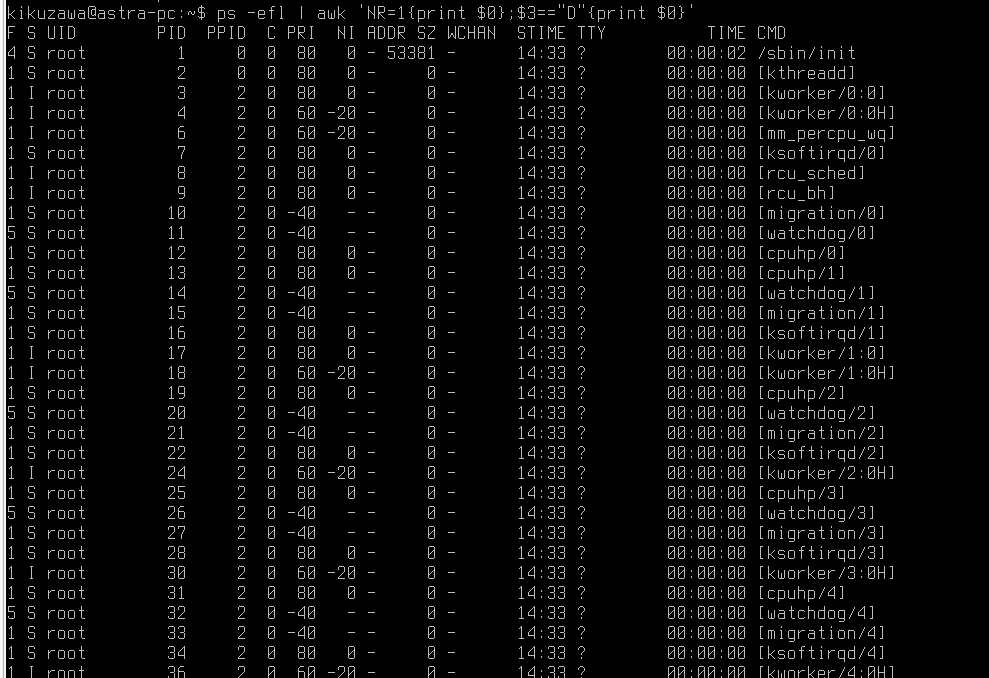
По относительному паритету N1

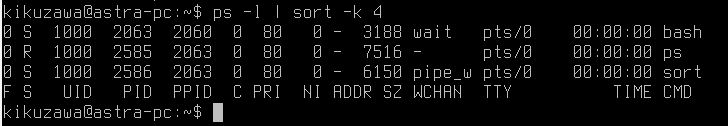
ps -l | sort -k 18

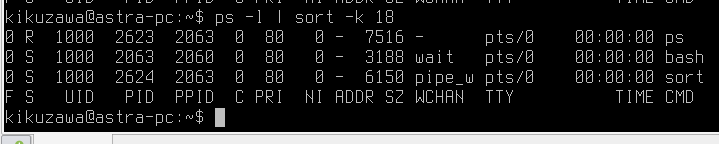












Задание 2.

С помощью `Ctrl+Alt+F2` `(Alt+F2)` откройте второй текстовый терминал и зарегистрируйтесь в нем как суперпользователь, запустите утилиту `top` для текущего контроля процессов.

Чтобы выйти из текстового терминала, нажми `CTRL + ALT + F7`

Задание 3.

Нажатием Ctrl+Alt-Fl (Alt-Fl) вернитесь в первую консоль. Из первой консоли создайте процесс `od /dev/zero > /dev/null`. В соответствии с введенной командой утилита `od` читает и выводит непрерывный поток нулевых байт из «рога изобилия» в нулевое устройство. Переключившись в другую консоль, с помощью команды `top` просмотрите список наиболее активных процессов. Найдите и идентифицируйте запущенный процесс, найдите по идентификатору `PPID` его «родителя», определите его приоритет (возможно это - величина переменная), долю загрузки центрального процессора `%CPU` и оперативной памяти `%МЕМ`.

Команда для удобного копирования: od /dev/zero > /dev/null

Задание 4.

Поочередно из первой и второй консолей с правами администратора и пользователя с помощью команды `od /dev/zero > /dev/null &` создайте по 2-3 одинаковых фоновых процесса. По мере создания новых процессов отслеживайте в третьей консоли их текущий приоритет, загрузку процессора и памяти. Имеются ли различия в приоритете процессов, выполняемых от имени администратора и пользователя?

od /dev/zero > /dev/null &

> `&` - фоновый процесс. Вам нужно 2-3 раза запустить

Чтобы посмотреть появились ли новые процессы, нужно вбить команду ниже:

Top

Чтобы убить процесс, используйте команду ниже:

kill [PID]

> `PID` - уникальный идентификатор процесса

Задание 5.

В консоли пользователя измените приоритет одного из принадлежащих ему процессов. Для этого воспользуйтесь командой `renice -10 PID`. Изменился ли относительный приоритет процесса?

Команда для использования в терминале: renice -10 PID

В `Linux` приоритеты процессов задаются с помощью значений, где более низкие числа означают более высокий приоритет. Таким образом, значение -10 действительно указывает на более высокий приоритет, чем 0. Обычный пользователь не может установить приоритет выше 0.

До этого создайте процесс, используя команду, которая представлена ниже:

od /dev/zero > /dev/null &

У Вас появится снизу `PID`, который вы должны добавить в `renice`. Например, у Вас может быть так:

renice -10 7199

Вы увидете, что ничего не поменялось, потому что не хватает уровня доступа.

Задание 6.

Повторите предыдущий пункт с правами администратора и ответьте на вопрос.

Команда для использования в терминале: sudo renice -10 PID

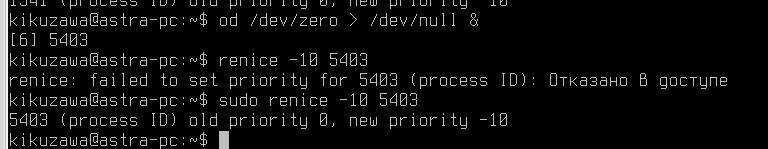
Задание 7.

Переключитесь в консоль пользователя и измените приоритет одного из принадлежащих ему процессов командой `renice 5 PID`. Произошло ли изменение приоритета?

Воспользуйтесь командой ниже: renice 5 -p PID

Здесь произошло изменение приоритета, так как у нас значение не меньше 0. Для пользователя такое вполне допустимо.

Для проверки приоритета можете воспользоваться командой ниже: ps -l -p PID



Задание 8.

Проконтролируйте из третьей консоли изменение приоритетов запущенных процессов. Что произошло?

Тут абсолютно разное поведение могло произойти. Для открытия 3 терминала, воспользуйтесь сочетанием клавиш,

которое представлено ниже:

CTRL + ALT + F3

Задание 9.

Удалите созданные процессы командой `kill`.

Посмотрите `ID` созданных процессов до этого. Теперь можете выполнить удаление.

Теперь перейдем к возможным вариантам удаления. Выбирайте любую команду на ваш вкус:

- `kill [PID1] [PID2] [PID3]`

- `killall od`

- `pkill -f 'od /dev/zero > /dev/null'`

- `killall -9 od`

- `pkill -9 -f 'od /dev/zero > /dev/null'`

> `-9` - сильный сигнал принудительного завершения

Лично я использовал 1 команду, где явно задаются `ID` для удаления процессов.

Задание 10.

С правами пользователя создайте в своей директории сценарий с именем `abсd`. Сценарий можно создать с помощью редактора `nano`

Перейдите в удобную Вам директорию, используя команду `cd`, пример представлен ниже:

cd /path/to/your/directory

Теперь создайте файл `abcd` в редакторе `nano`. Для этого в терминале просто вбейте команду: nano abcd

После этого вставьте код, который представлен ниже:

```bash

#! /bin/bash

while :

do

echo HELLO!

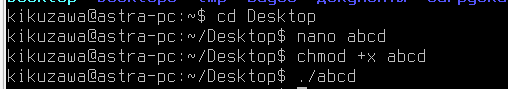
done

```

Теперь нам нужно сделать файл выполняемым, для этого вбейте команду, которая представлена ниже:

chmod +x abcd

Ну и для запуска воспользуйтесь командой, которая представлена ниже: ./abcd



Задание 11.

Используя команду `chmod`, присвойте пользователю полные права на чтение, запись и исполнение данного сценария. Запустите сценарий на исполнение (на экран должны непрерывно выводиться приветствия `HELLO!`)

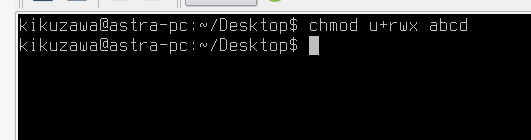
Мы уже до этого создали файл, нам нужно выдать полные права на выполнение. Для этого воспользуйтесь командой ниже:

chmod u+rwx abcd

> Здесь `u` обозначает владельца (`user`), `+rwx` добавляет права на чтение (`r`), запись (`w`) и исполнение (`x`).

Теперь для запуска скрипта можете воспользоваться командой, которая представлена ниже:

./abcd



Задание 12.

Перейдите в третью консоль, с помощью команды top просмотрите список процессов и найдите в нем «зависший» процесс, запущенный пользователем (на самом деле это только имитация зависания, которое пользователь легко может прекратить сам). Прочитайте идентификатор процесса PID.

Опять-таки все описано в условии. Команда, которой Вам нужно воспользоваться: top

Задание 13.

Нажатием Ctrl+C из второй консоли остановите процесс. Как изменилось при этом состояние процесса?

Он перестает окончательно работать, падает полностью нагрузка, после этого процесс пропадает с экрана.

Как же это происходит?

Когда вы нажимаете `CTRL + C`, то ОС отправляет сигнал `SIGINT`, который по умолчанию предназначен для прерывания сигнала.

Бывает такое, что процесс может переопределить данный сигнал для своего поведения.

Задание 14.

Повторно запустите из второй консоли процесс, перейдите в первую консоль и отправьте "зависшему" процессу сигнал на останов (команда kill -15 PID\_process).

Перейдем во 2 консоль. Для этого используйте комбинацию клавиш, которая представлена ниже: CTRL + ALT + F2

Как только перешли во вторую консоль, набираем команду в терминале для запуска процесса.

Это будет тот же наш скрипт, который до этого писали: ./abcd

Теперь перейдем в 1 консоль. Для этого используйте комбинацию клавиш, которая представлена ниже: CTRL + ALT + F1

Теперь остается Вам только убить процесс, узнав заранее его идентификатор.

kill -15 PID

> `PID` - номер процесса, его можно узнать через `top`

> `kill - 15 PID` - стандартное

Задание 15.

Перейдите в другую консоль и отправьте процессу сигнал kill -20 PID. Как реагирует процесс на данный сигнал? Что означает данный сигнал?

Перейдем во 2 консоль. Для этого используйте комбинацию клавиш, которая представлена ниже: CTRL + ALT + F2

Как только перешли во вторую консоль, набираем команду в терминале для запуска процесса.

Это будет тот же наш скрипт, который до этого писали: ./abcd

Теперь перейдем в 1 консоль. Для этого используйте комбинацию клавиш, которая представлена ниже: CTRL + ALT + F1

Теперь остается Вам только убить процесс, узнав заранее его идентификатор. kill -20 PID

> `PID` - номер процесса, его можно узнать через `top`

> `kill -20 PID` - сигнал временного завершения (заморозка процесса). Обычно такое можно повторить с помощью `CTRL + Z`.

Задание 16.

С помощью команды kill -9 PID отправьте этому процессу сигнал принудительного завершения. С другой консоли проконтролируйте выполнение команды. Остановился ли процесс? Остался ли он в списке процессов? Какая программа на самом деле перехватывает и исполняет команду kill -9 PID?

Перейдем во 2 консоль. Для этого используйте комбинацию клавиш, которая представлена ниже: CTRL + ALT + F2

Как только перешли во вторую консоль, набираем команду в терминале для запуска процесса.

Это будет тот же наш скрипт, который до этого писали: ./abcd

Теперь перейдем в 1 консоль. Для этого используйте комбинацию клавиш, которая представлена ниже: CTRL + ALT + F1

Теперь остается Вам только убить процесс, узнав заранее его идентификатор.

kill -9 PID

> `PID` - номер процесса, его можно узнать через `top`

> `kill -9 PID` - принудительное завершение без перехвата.

Задание 17

С помощью команды `echo $PATH` поочередно из консоли администратора и пользователя `userl` выведите список директорий, в которых производится поиск исполняемых файлов, заданных только по имени. В чем заключается различие выведенных списков? Почему в списке `PATH` администратора отсутствует текущий каталог (.)? Почему в списке `PATH` пользователя отсутствует каталог `/sbin`? Имеет ли пользователь возможность изменить порядок проверки каталогов для администратора?

Я, так понимаю, нового пользователя создавать не надо. Работаем дальше под текущим.

Введите команду для вывода директорий в консоли: echo $PATH

Почему в списке `PATH` у администратора отсутствует текущий каталог?

Исключение текущего каталога из `PATH` для администратора является мерой безопасности.

Путь администратора обычно включает директории, содержащие системные и административные утилиты, такие как `/sbin`, `/usr/sbin`.

Эти каталоги содержат программы для управления системой, которые не предназначены для запуска не привилегированными пользователями.

Почему в списке `PATH` пользователя отсутствует каталог `sbin`?

Каталог `/sbin` и подобные ему (`/usr/sbin`, `/usr/local/sbin`) содержат программы для администрирования системы,

которые обычно не предназначены для использования обычными пользователями.

Пользователь не может напрямую изменить порядок проверки каталогов для администратора без соответствующих привилегий.

Однако любой пользователь может изменить свой собственный `PATH`, используя команды типа `export PATH=...` в своем

`.bashrc`, `.bash\_profile` или других конфигурационных файлах оболочки.

Задание 18.

Попробуйте запустить несколько утилит из второй консоли с правами пользователя (например, `renice -10 PID`, `date -s O`). Как реагирует система на ваши попытки? Опишите.

Чтобы сменить приоритет на высокий, нужно пользоваться командой: renice -10 PID

> `PID`- номер процесса.

> Команда `renice` изменяет приоритет запущенных процессов.

Чтобы сменить время на начало эпохи `Unix`, нужно: date -s O

> Вместо '0' может быть любое значение в секундах, начиная с эпохи `Unix`

Попытка обычного пользователя изменить данные параметры приводит к ошибке. Для этого нужно `sudo`.

Задание 19.

Перейдите в административную консоль и повторите запуск утилит с правами суперпользователя. Что произошло?

Чтобы сменить приоритет на высокий, нужно пользоваться командой: sudo renice -10 PID

> `PID`- номер процесса.

> Команда `renice` изменяет приоритет запущенных процессов.

Чтобы сменить время на начало эпохи `Unix`, нужно: sudo date -s O

> Вместо '0' может быть любое значение в секундах, начиная с эпохи `Unix`

Всё получилось, в связи с тем, что у нас есть права суперпользователя.

Система не ограничивает попытки.

Задание 20.

Убедитесь в том, что пользователю разрешен запуск указанных утилит. Объясните, почему пользователь не может запустить утилиты с некоторыми «критичными» параметрами? Где, по вашему мнению, расположен механизм контроля за ходом запуска (в ядре операционной системы, в командной оболочке, в самой утилите?). Ответ обоснуйте.

Опять-таки приведу пример с `renice`, который пользователь может запускать, но с определенным параметрами выдает ошибку.

renice 5 PID

> `PID`- номер процесса.

> Команда `renice` изменяет приоритет запущенных процессов.

В итоге почему так происходит? Происходит в целях безопасности, чтобы пользователь просто так систему не навернул.

Где же в итоге тогда защита? На каком слое в ОС? Тут первичным фильтром выступает сама оболочка и программа.

Они предоставляют дополнительные уровни проверки, но это зачастую бывает недостаточно, поэтому ядро выступает главным

контроллером.

Ядро операционной системы является ключевым компонентом в контроле за ходом запуска и выполнения команд,

поскольку именно оно имеет доступ ко всем ресурсам системы и контролирует права доступа на самом низком уровне.

Командная оболочка и отдельные программы могут предоставлять дополнительные уровни проверки, но они действуют в рамках

правил и ограничений, установленных ядром. Это обеспечивает, что даже если внешний уровень (например, оболочка или приложение)

подвергнется компрометации, низкоуровневый контроль защиты, предоставляемый ядром, останется в силе.

Задание 21.

С правами пользователя скопируйте в свой рабочий каталог один из исполняемых файлов с параметром `SUID` каталога `/bin` (исполняемые файлы выделены цветом и символом \*, а параметр `SUID` отмечен символом «s» в правах владельца на исполнение). Как изменились права доступа к файлу после его копирования?

> Бит `SUID` (Set User ID) — это специальный бит в Unix-подобных операционных системах,

> который позволяет исполняемым файлам запускаться с правами владельца файла, а не с правами пользователя,

> который запустил файл.

> Это может быть полезно для выполнения программ, которые требуют повышенных привилегий для выполнения определенных операций.

Как найти файл, где есть SUID? Воспользуйтесь командой ниже: ls /bin | grep '^...s'

Я же лично буду использовать для копирования команду `ping`. Пример команды для копирования представлена ниже: cp /bin/ping ~/my\_directory/

Что же происходит когда мы копируем в домашний каталог? В таком случае сбрасывается `SUID`.

Сделано в целях безопасности.

Задание 22.

Из второй консоли с правами пользователя скопируйте в свой домашний каталог утилиту, которую разрешено запускать только администратору (например, `chattr`). Копирование производите с параметрами, гарантирующими переход копии во владение пользователю. Убедитесь и покажите, что пользователь имеет на скопированный файл все необходимые права.

Перейдем во вторую консоль, для этого используем сочетание клавиш:

CTRL + ALT + F2

Теперь скопируем `chattr` с правами пользователя, для этого воспользуйтесь командой ниже: cp /usr/bin/chattr ~/chattr\_copy

> Если не пропускает, то используйте ещё `sudo` перед командой

Теперь назначим права, для этого воспользуйтесь командами ниже:

```bash

sudo chown $USER:$USER ~/chattr\_copy

chmod +x ~/chattr\_copy

```

Проверьте права доступа, вбив команду ниже: ls -l ~/chattr\_copy

Попробуем воспользоваться, для этого выполните набор команд, который представлен ниже:

```bash

touch ~/test\_file

~/chattr\_copy +i ~/test\_file

```

Вывод: не все утилиты будут функционировать корректно, если они предполагают взаимодействие с системными файлами или

требуют глобальных изменений в системе, что выходит за рамки прав обычного пользователя.

Задание 23.

Настроить на запуск однократно некоторую задачу в определенное время. Вы также должны наметить другую задачу, которая стартовала бы каждые десять минут (между 8:00 и 17:00). Задачи выбрать самостоятельно.

В `Linux` есть две утилиты, которые позволяют запускать задачи на конкретное время.

- `at` - для однократных задач, запланированных на конкретное время. Задача запускается один раз. То же самое, что и таймер.

- `cron` - для периодических задач. То есть задача запускается каждый раз в определенный период.

Давайте создадим скрипты, которые можно будет потом использовать для запуска.

Для однократной задачи:

```bash

#!/bin/bash

echo "One-time script executed at $(date)" > result.txt

```

Для периодической задачи:

```bash

#!/bin/bash

echo "Periodic script executed at $(date)" > result.txt

```

> Перед запуском каждого скрипта установите ему права на запуск, используя команду `chmod +x <name>`, где `name` - это название `sh` скрипта.

Теперь для установки однократной задачи используйте пример ниже:

at 12:00 <<< "[path/to/script]"

> `path/to/script` - это абсолютный путь к скрипту.

Теперь для установки периодической задачи будем использовать команды ниже: crontab -e

В конце файла добавьте строку, как показано ниже: \*/10 8-17 \* \* \* [path/to\_script]

> Это путь к скрипту, который нужно вставить.

> `cron` и `at` на `Astra Linux` 1.8 не работают нормально, они не могут открывать приложения, выводить результат в консоль.

> Методом проб и ошибок нашли, что они нормально могут только записывать в `txt` файл.

Задание 24.

Создать напоминание, например, о встрече с Х в 12:00, сегодня. Войдите в систему как `root` и введите следующие команды: `at noon`; <Нажмите Enter > `echo "Time for meet with Х."` <Нажмите Enter > <После завершения нажмите ctrl-D>. Проверьте очередь заданий, используя команду `atq`.

> Выполняйте последовательно команды, которые я описал ниже.

1. Заходим под основной учетной записью, то есть под администратором в учетную запись.

2. Открываем терминал, используя комбинацию `CTRL + ALT + F1`

3. Вбиваем команду `at noon`

4. Теперь вбиваем команду `echo "Time for meet with X."`

5. Нажимаем комбинацию клавиш `CTRL + D`

6. Проверяем очередь, вбивая в терминал команду: `atq`

Задание 25.

Настройте получение информации о статусе системы каждые десять минут.

Для начала, нам нужно зайти, как суперпользователь. Для этого вбейте в терминале команду ниже: su root

Теперь откроем планировщик задач в терминале, для этого воспользуйтесь командой, которая представлена ниже: crontab -e

Укажите строку, которая представлена ниже: \*/10 8-17 \* \* \* /usr/bin/free >> /root/free.txt

> Данная команда выводит информацию о доступной и использованной памяти,

> а потом перенаправляет ее вывод в файл `/root/free.txt`

После этого нажмите `CTRL + D`

Результат можно увидеть теперь в файле, для этого в терминале вбейте команду, которая представлена ниже: cat /root/free.txt

Задание 26.

Выполните задачи: убедиться, что каталог, где сохраняются задания at, пуст; проверить время и дату в терминале; запланировать создание файла в корневом каталоге через две минуты;

Произведем настройку задания с использованием `at`

> Убедитесь, что каталог, где сохраняются задания `at`, пуст.

> Это можно проверить командой `atq`.

> Если есть задания, их можно удалить с помощью `atrm` <номер задания>.

Теперь проверьте текущее время и дату в терминале. Вы можете использовать команду ниже: date

Зачем мы проверяли дату и время? Неизвестно. Ведь можно, не зная текущее время, поставить на выполнение скрипт: echo "touch /root/testfile.txt" | at now + 2 minutes

> Выше представлен пример команды, которая запланирует создание файла через 2 минуты.

Теперь посмотрите наличие того, что у Вас добавлена задача в очередь. Для этого используйте команду, которая представлена ниже: atq

> Проверьте каталог, где собираются задания в очереди на выполнение.

> Обычно это `/var/spool/at` или `/var/spool/cron/atjobs`, в зависимости от вашей системы.

> Вы должны увидеть новый файл, который представляет задание.

Контрольные вопросы по Astra Linux

1. Имеются ли различия в приоритете процессов, выполняемых от имени администратора и пользователя?

Да, в Astra Linux (как и в других Unix-подобных системах) процессы, выполняемые от имени администратора (root), могут иметь более высокие приоритеты доступа к системным ресурсам по сравнению с процессами, запущенными обычным пользователем. Это связано с тем, что администратору необходимо обеспечить бесперебойную работу критически важных системных процессов.

2. Как изменяется относительный приоритет процесса?

Относительный приоритет процесса можно изменить с помощью команды `nice` и `renice`. Команда `nice` задает начальный приоритет для нового процесса, а `renice` изменяет приоритет уже запущенного процесса. Чем ниже значение nice (от -20 до 19), тем выше приоритет процесса.

3. Почему в списке PATH администратора отсутствует текущий каталог (.)?

Текущий каталог (.) отсутствует в PATH администратора для повышения безопасности. Это предотвращает случайный запуск вредоносных программ из текущего каталога, что может привести к компрометации системы.

4. Почему в списке PATH пользователя отсутствует каталог /sbin?

Каталог `/sbin` содержит системные утилиты, которые обычно используются администратором для управления системой. Обычным пользователям не требуется доступ к этим утилитам, поэтому `/sbin` не включен в их PATH.

5. Имеет ли пользователь возможность изменить порядок проверки каталогов для администратора?

Нет, обычный пользователь не может изменить порядок проверки каталогов в PATH администратора. Это связано с ограничениями прав доступа: только администратор может изменять системные настройки, включая PATH.

6. Где расположен механизм контроля за ходом запуска утилит (в ядре операционной системы, в командной оболочке, в самой утилите?)

Механизм контроля за ходом запуска утилит расположен в \*\*командной оболочке\*\*. Именно оболочка интерпретирует команды, ищет исполняемые файлы в каталогах, указанных в PATH, и запускает их.

7. Покажите, какие основные три процесса наиболее потребляют ресурсы на вашем компьютере?

Для просмотра процессов, потребляющих наибольшее количество ресурсов, можно использовать команду `top` или `htop`. В верхней части вывода будут отображены процессы, потребляющие больше всего CPU и памяти. Пример: top

8. Покажите, присутствуют ли на вашей машине процессы-зомби?

Процессы-зомби можно найти с помощью команды `top` или `ps`. В `top` они обозначаются как "Z" в колонке состояния. Пример: ps aux | grep 'Z'

9. Назначение Cron и Anacron.

- Cron используется для планирования периодических задач, которые должны выполняться в определенное время (например, ежедневно, еженедельно).

- Anacron предназначен для выполнения задач, которые должны запускаться с определенной периодичностью, но не требуют точного времени запуска (например, на машинах, которые не работают 24/7).

10. Crontab и at применение.

- Crontab используется для создания, редактирования и управления расписанием выполнения задач.

- At позволяет запустить задачу однократно в указанное время.

11. С помощью какой команды пользователь может составить расписание для запуска периодических заданий?

Для составления расписания используется команда `crontab -e`. Она открывает редактор для редактирования файла crontab пользователя.

12. Какой командой можно увидеть активные таймеры?

Активные таймеры можно увидеть с помощью команды `systemctl list-timers`. Она покажет все запланированные таймеры systemd, включая те, которые связаны с cron и anacron.

Пример:

systemctl list-timers