## ЛАБОРАТОРНАЯ РАБОТА 1. ИНТЕРАКТИВНАЯ РАБОТА В СИСТЕМЕ. ПОЛЬЗОВАТЕЛЬСКАЯ УЧЕТНАЯ ЗАПИСЬ ............... 2

*Упражнение 1.1. Загрузка операционной системы. Вход в систему.................................................... 2*

*Упражнение 1.2. Виртуальные терминалы ...................................................................................................... 2*

*Упражнение 1.3. Завершение сеанса ................................................................................................................. 3*

*Упражнение 1.4. Команды для получения информации о системе и работающих*

*пользователях ............................................................................................................................................................... 3*

*Упражнение 1.5. Редактирование командной строки командного интерпретатора bash и*

*история команд ............................................................................................................................................................. 4*

*Упражнение 1.6. Свойства пользовательской учетной записи .............................................................. 4*

*Упражнение 1.7. Изменение контекста пользователя ................................................................................ 5*

*Упражнение 1.8. Общение между пользователями вычислительной системы ............................... 6*

## ЛАБОРАТОРНАЯ РАБОТА 2. РАБОТА СО СПРАВОЧНЫМИ СИСТЕМАМИ ............................................................................... 7

*Упражнение 2.1. Страницы электронного руководства UNIX ................................................................. 7* *Упражнение 2.2. Встроенная справка командного интерпретатора bash ......................................... 8*

*Упражнение 2.3. Справочная система GNU Info ........................................................................................... 9*

ЛАБОРАТОРНАЯ РАБОТА 3. ПОЛЬЗОВАТЕЛЬСКОЕ ОКРУЖЕНИЕ. ПОЛЬЗОВАТЕЛЬСКИЙ ПРОФИЛЬ ............................... 10 *Упражнение 3.1. Переменные окружения. Переменные командного интерпретатора ............. 10*

*Упражнение 3.2. Конфигурационные файлы пользовательского окружения.*

*Пользовательский профиль .................................................................................................................................. 11*

## ЛАБОРАТОРНАЯ РАБОТА 4. ФАЙЛЫ И ФАЙЛОВАЯ СИСТЕМА UNIX ................................................................................. 13

*Упражнение 4.1. Изучение структуры файловой системы ..................................................................... 13*

*Упражнение 4.2. Типы файлов ............................................................................................................................ 13*

*Упражнение 4.3. Манипулирование объектами дерева каталогов UNIX ......................................... 14*

*Упражнение 4.4. Поиск файлов .......................................................................................................................... 15*

*Упражнение 4.5. Операции с файловыми системами ............................................................................... 16*

### ЛАБОРАТОРНАЯ РАБОТА 5. ВЛАДЕЛЬЦЫ И ПРАВА ДОСТУПА ОБЪЕКТОВ ФАЙЛОВОЙ СИСТЕМЫ UNIX. ...................... 17

*Упражнение 5.1. Владельцы файлов ................................................................................................................ 17*

*Упражнение 5.2. Права доступа к объектам файловой системы UNIX ............................................. 17*

ЛАБОРАТОРНАЯ РАБОТА 6. ОБРАБОТКА ТЕКСТОВОЙ ИНФОРМАЦИИ. УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ КОМАНД.

## ПОДСТАНОВКИ КОМАНДНОГО ИНТЕРПРЕТАТОРА ............................................................................................................... 19

*Упражнение 6.1. Обработка текстовой информации в UNIX ................................................................. 19*

*Упражнение 6.2. Перенаправление ввода-вывода информации ......................................................... 20*

*Упражнение 6.3. Подстановка вывода команд ............................................................................................ 21*

*Упражнение 6.4. Подстановка шаблонных символов ............................................................................... 21*

## ЛАБОРАТОРНАЯ РАБОТА 7. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА UNIX ............................................................................... 23

*Упражнение 7.1. Упаковщики и архиваторы UNIX ..................................................................................... 23*

*Упражнение 7.2. Текстовые редакторы UNIX ............................................................................................... 24*

*Упражнение 7.3. Файловый менеджер UNIX - Midnight Commander ................................................. 25*

*Упражнение 7.4. Подсистемы отложенных и периодических заданий. ........................................... 25*

## ЛАБОРАТОРНАЯ РАБОТА 8. ПРОЦЕССЫ UNIX .................................................................................................................. 28

*Упражнение 8.1. Типы процессов ...................................................................................................................... 28*

*Упражнение 8.2. Атрибуты процессов ............................................................................................................. 29*

*Упражнение 8.3. Задания и сигналы ................................................................................................................ 29*

## ЛАБОРАТОРНАЯ РАБОТА 9. СЕТЕВОЕ ОКРУЖЕНИЕ UNIX ................................................................................................. 31

*Упражнение 9.1. Удаленный доступ в систему с помощью протокола SSH ................................... 31*

*Упражнение 9.2. Локальная и сетевая почта ............................................................................................... 32*

*Упражнение 9.3. Передача файлов с помощью протокола FTP ........................................................... 32*

*Упражнение 9.4. Передача файлов с помощью протоколов SCP ........................................................ 33*

*Упражнение 9.5. Синхронизация файлов с помощью протокола RSYNC/SSH .............................. 33*

## ЛАБОРАТОРНАЯ РАБОТА 10. СИСТЕМА X WINDOW .......................................................................................................... 34

*Упражнение 10.1. Создание собственного настольного окружения .................................................. 34*

*Упражнение 10.2. Настройка настольного окружения Xfce ................................................................... 35*

*Упражнение 10.3. Графический вход в систему ......................................................................................... 35*

## ЛАБОРАТОРНАЯ РАБОТА 11. ПОДСИСТЕМА ПЕЧАТИ ......................................................................................................... 36

*Упражнение 11.1. Средства печати UNIX ....................................................................................................... 36*

### ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ КОМАНДНОГО ИНТЕРПРЕТАТОРА. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ ............... 38

*Создание скелета сценария на языке командного интерпретатора BASH ...................................... 38*

*Реализация алгоритма сценария на языке командного интерпретатора BASH ............................ 38*

**Лабораторная работа 1. Интерактивная работа в системе. Пользовательская учетная запись**

# Упражнение 1.1. Загрузка операционной системы. Вход в систему

1. Включите компьютер.
2. В приглашении загрузчика выбора операционной системы выберите Linux.
3. Проследите за сообщениями системы, выдаваемыми в процессе загрузки. Появление приглашения к вводу имени пользователя, означает окончание процесса загрузки системы. Примечание 1: для отображения полного лога загрузки ОС требуется убрать параметр ядра *quite*. Примечание 2: если по умолчанию установлен графический вход в ОС, переключитесь на виртуальную консоль сочетанием клавиш “Ctrl-Alt-F1” или “CtrlAlt-F2”.
4. Зафиксируйте приглашение к вводу имени пользователя и расшифруйте его составляющие:

* Ubuntu 22.04.3 LTS – версия ОС;
* ubuntu-edu – название машины;
* tty1 – название/номер виртуального терминала.

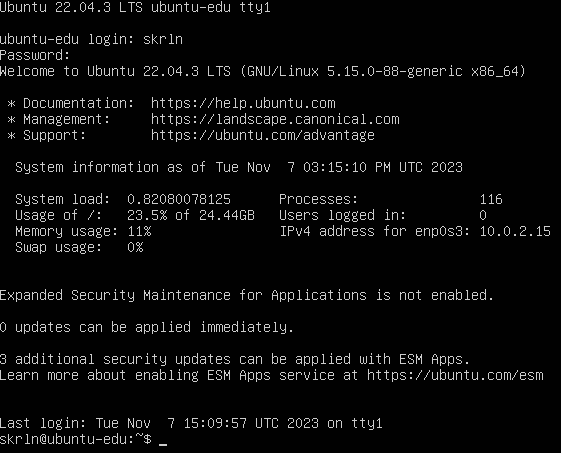


1. В приглашении к вводу имени пользователя введите имя учетной записи *student*или другую доступную учетную запись.
2. Зафиксируйте приглашение к вводу пароля и расшифруйте его составляющие:



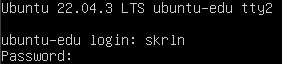
1. В приглашении к вводу пароля введите *password*или другой установленный пароль. В случае правильного ввода имени пользователя и пароля появится приглашение к вводу команд.
2. Зафиксируйте появившиеся сообщения и приглашение к вводу команд и расшифруйте его составляющие:

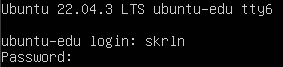
* Ссылки на документы, связанные с ОС;
* Системная информация (использование памяти, кол-во процессов и т.д.);
* Информация об обновлениях.



# Упражнение 1.2. Виртуальные терминалы

1. Используя клавиши **Alt+Fx**, **Alt+←**, **Alt+→** переключитесь и войдите в систему на 2-ом, 4-ом и 6-ом терминале. Примечание: если работаете в графической системе для перехода используйте сочетание “**Ctrl-Alt+Fx**”.
2. Зафиксируйте появляющееся сообщение от системы:



1. Проследите за появившимися приглашениями к вводу команд.
2. При помощи команды **tty**, получите и зафиксируйте информацию о 2-ом, 4-ом и 6-ом виртуальном терминале соответственно:

1. Вернитесь на терминал #1.
2. При помощи команды **clear** очистите терминал, проследите за реакцией системы.

* После выполнения **clear** осталось только новое приглашение ко вводу команд.

Примечание: если данное упражнение невыполнимо с использованием виртуальных терминалов, например, при использовании MacOS X или WSL, то выполните **tty** для эмулятора терминала.

# Упражнение 1.3. Завершение сеанса

1. Последовательно переключаясь между виртуальными терминалами, завершите сеанс работы с системой на терминалах, на которых вы входили в систему, при помощи команд **exit**, **logout**, либо используя комбинацию клавиш **^D**.
2. Проследите за соответствующей реакцией системы.

* Система возвращается к приглашению ко вводу имени пользователя

# Упражнение 1.4. Команды для получения информации о системе и работающих пользователях

При помощи следующих команд: **whoami**, **id**, **users**, **who**, **w**, **date**, **cal**, **uname**, **uptime** получите информацию о системе и пользователях и зафиксируйте ее:



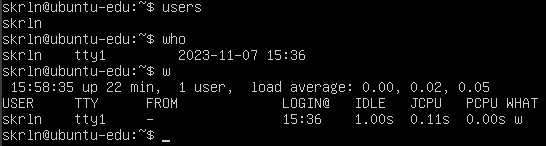
1. имя текущего пользователя:



1. его идентификаторы UID, GID и идентификаторы вторичных групп:



1. количество и имена пользователей, работающих в системе, их терминалы, время работы в системе и время входа в систему, выполняемые команды:

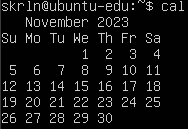


1. системная дата и время:



1. календарь текущего месяца:

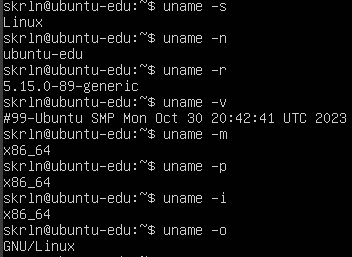




1. версия операционной системы компьютера:



* **-a, --all** Вывести всю информацию
* **-s, --kernel-name** Вывести имя ядра
* **-n, --nodename** Вывести имя машины в сети (имя хоста)
* **-r, --kernel-release** Вывести информацию о выпуске ядра
* **-v, --kernel-version** Вывести версию ядра
* **-m, --machine** Вывести тип оборудования машины
* **-p, --processor** Вывести тип процессора
* **-i, --hardware-platform** Вывести тип аппаратной платформы
* **-o, --operating-system** Вывести название операционной системы

****

1. время работы операционной системы от момента запуска и нагрузку на операционную систему:



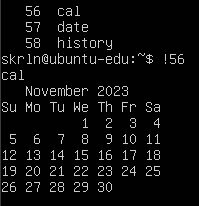
* Последние числа – нагрузка за последние 1, 5, 15 мин.

# Упражнение 1.5. Редактирование командной строки командного интерпретатора bash и история команд

1. Используя клавиши **↑**, **↓** просмотрите историю вводимых ранее команд.
2. Используя клавиши **←**, **→**, **Del**, **BackSpace**, **^H**, **^?**, **^U** потренируйтесь редактировать вводимую команду.
3. Начните набирать любую известную вам команду (первые несколько букв), воспользуйтесь клавишей **Tab** для автоматического завершения ввода команды.
4. При помощи команды: **history** получите историю ранее вводимых комманд и зафиксируйте три последние:

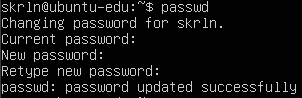


1. Повторите любую ранее вводимую команду при помощи ссылки на ее номер: **!n**, где **n** - номер введенной ранее команды.



# Упражнение 1.6. Свойства пользовательской учетной записи

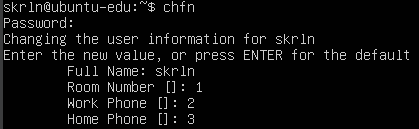
1. При помощи команды: **passwd** измените пароль учетной записи. Зафиксируйте появляющиеся сообщения:



1. При помощи команды: **finger** посмотрите и зафиксируйте информацию о свойствах учетных записей student и суперпользователя соответственно:



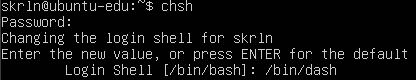
1. При помощи команды: **chfn** измените содержание информационного свойства (GECOS) учетной записи student. Зафиксируйте появляющиеся сообщения:



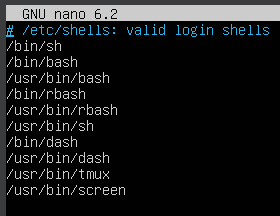
1. При помощи команды: **chsh** измените начальный командный интерпретатор учетной записи student. Зафиксируйте появляющиеся сообщения:



1. Посмотрите и зафиксируйте изменения в свойствах учетной записи student:



* Интепретатор должен быть в /etc/shells (редачит только рут)!

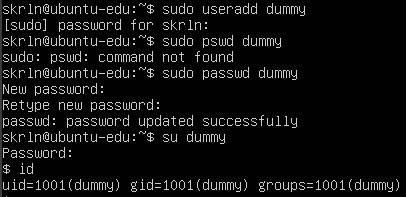


1. Завершите сеанс работы пользователя в системе и заново войдите в систему, проследите за изменениями, произошедшими вследствие смены начального командного интерпретатора:

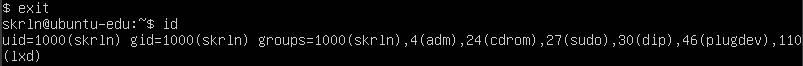


# Упражнение 1.7. Изменение контекста пользователя

1. При помощи команды: **su** измените контекст текущего пользователя *student* на контекст пользователя *vinnie* (или другого пользователя), и зафиксируйте его идентификаторы UID, GID и идентификаторы вторичных групп. Примечание: если второго пользователя в системе не существует, то добавьте его с помощью команд **useradd** и **passwd**:



1. При помощи команды: **exit** вернитесь в контекст текущего пользователя *student*, убедитесь в этом, проверив его идентификаторы UID, GID и идентификаторы вторичных групп:



# Упражнение 1.8. Общение между пользователями вычислительной системы

1. Переключитесь и войдите в систему на свободном виртуальном терминале, воспользовавашись учетной записью *vinnie*. Вернитесь на терминал пользователя *student*.
2. При помощи команды **write** пошлите различные сообщения (завершая ввод каждого сообщения управляющим символом **^D** - символом завершения ввода) пользователю *vinnie*, проследите за появляющимися на терминале пользователя *vinnie* сообщениями:

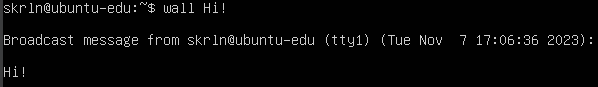
 





1. При помощи команды **mesg** отключите возможность приема сообщений пользователем *vinnie*. Попробуйте послать сообщения пользователю *vinnie*, проследите за появляющимися сообщениями: write: (*dummy has messages disabled* с первого скриншота пред. задания)

При помощи команды **wall** пошлите сообщение всем пользователям сразу (завершая ввод каждого сообщения управляющим символом **^D** - символом завершения ввода), проследите за появляющимися сообщениями:





Лабораторная работа зачтена:

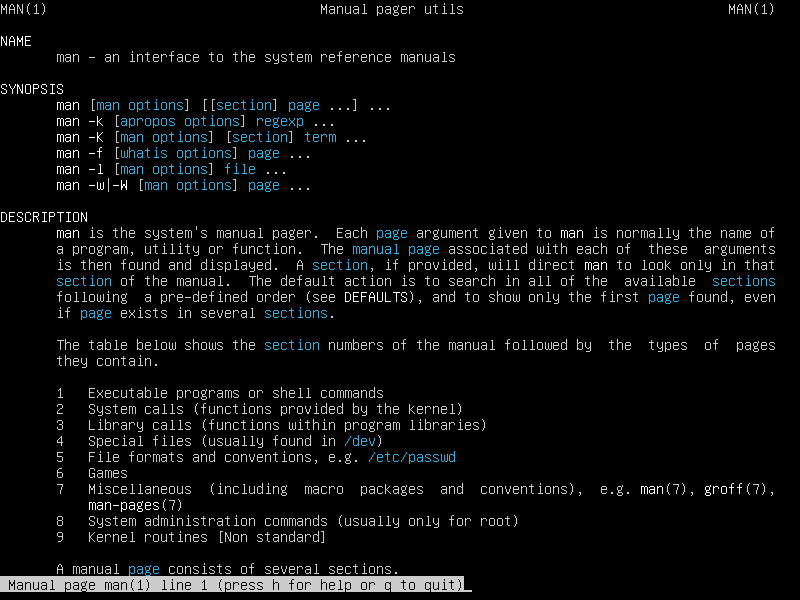
Дата:

Подпись преподавателя:

**Лабораторная работа 2. Работа со справочными системами**

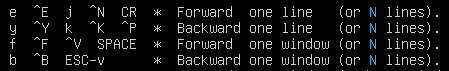
# Упражнение 2.1. Страницы электронного руководства UNIX

1. При помощи команды: **man** man ознакомтесь со справкой по команде **man**. Зафиксируйте назначение команды **man**:



* man – отображает страницы справочника переданного исп. файла \ команды

1. Страницы руководства выводятся на терминал постранично при помощи программы постраничной разбивки (PAGER). Вызовите встроенную справку по программе постраничной разбивки с помощью клавиши **h**. Ознакомьтесь с основами работы со справочной системой и зафиксируйте основные комбинации клавиш, используемые для навигации и поиска в справочной системе, а именно:
   1. клавиши построчного и постраничного листания вверх и вниз:



* 1. клавишу выхода из справочной системы:



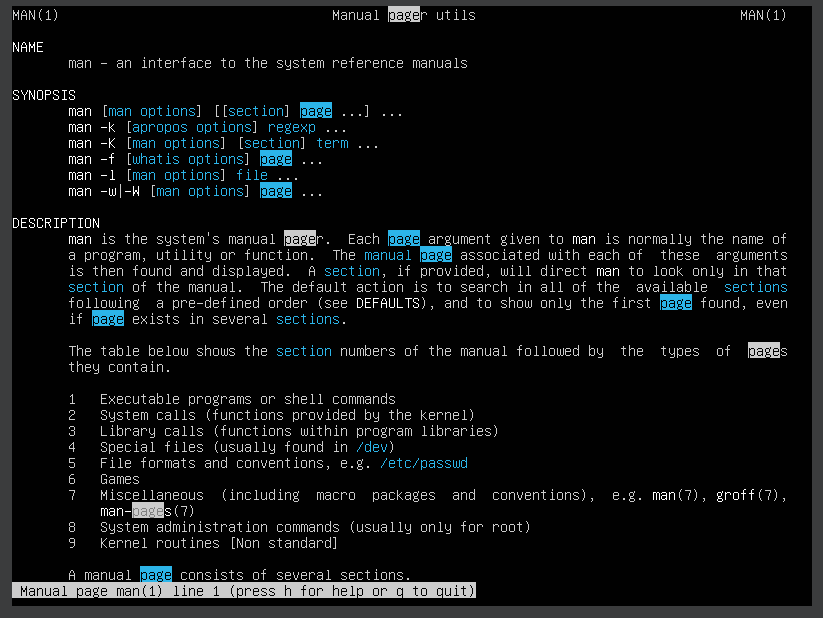
* 1. клавиши поиска информации в странице руководства:



* n N !!!
  1. клавиши повторения предыдущего поиска:



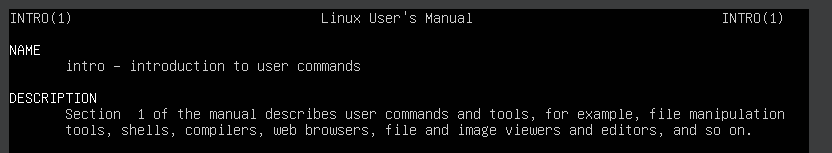
1. При помощи клавиш поиска и повторения поиска найдите на текущей странице все появления слова **page**, зафиксируйте использованные сочетания клавиш:

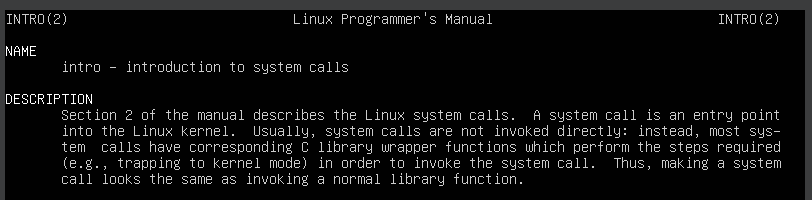


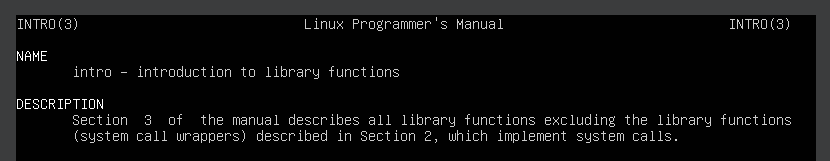
* \page

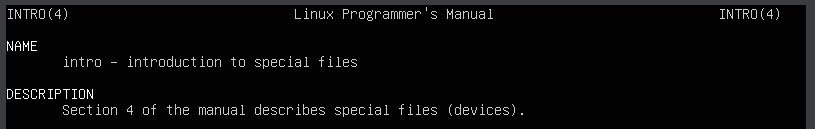
1. При помощи команды: **man** <номер раздела> **intro** ознакомьтесь с назначением всех восьми разделов справочной системы. Зафиксируйте их назначение:

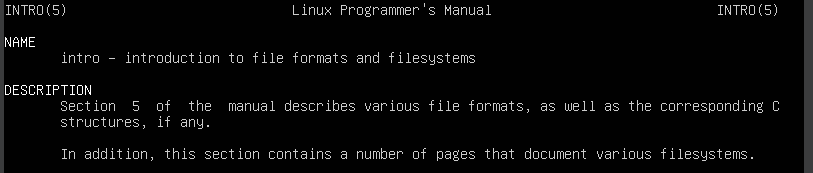
* 1 - Исполняемые программы или команды оболочки (shell)
* 2 - Системные вызовы (функции, предоставляемые ядром)
* 3 - Библиотечные вызовы (функции, предоставляемые программными библиотеками)
* 4 - Специальные файлы (обычно находящиеся в каталоге /dev)
* 5 - Форматы файлов и соглашения, например о /etc/passwd
* 6 - Игры
* 7 - Разное
* 8 - Команды администрирования системы (обычно, запускаемые только суперпользователем)
* 9 - Процедуры ядра [нестандартный раздел]

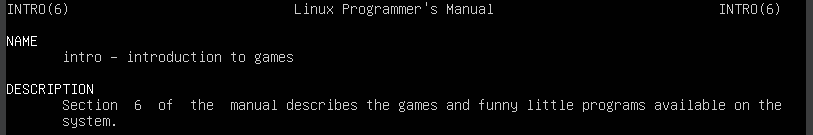


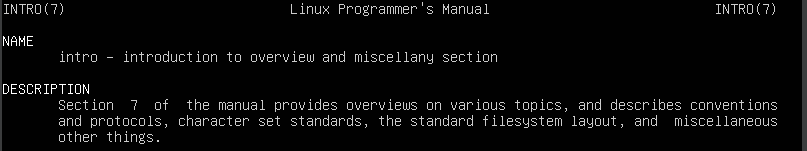


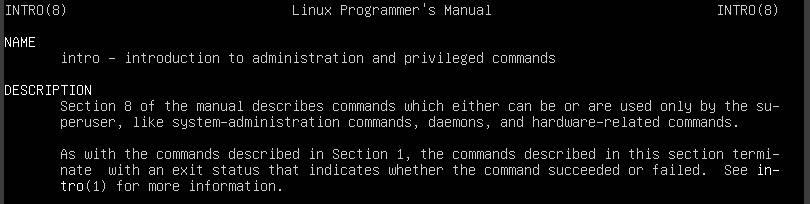






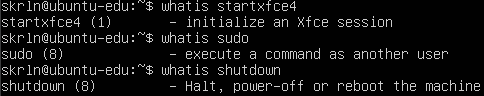








1. При помощи команды: **whatis** получите краткую справку по всем известным вам командам. Зафиксируйте и расшифруйте составляющие нескольких описаний команд:



!!!

1. При помощи команды: **apropos** воспользуйтесь контекстным поиском справочной информации по любому ключевому слову. Зафиксируйте и расшифруйте составляющие нескольких найденных ссылок:



!!!

* Поиск переданной пользователем строки в заголовках страниц руководств.

# Упражнение 2.2. Встроенная справка командного интерпретатора bash

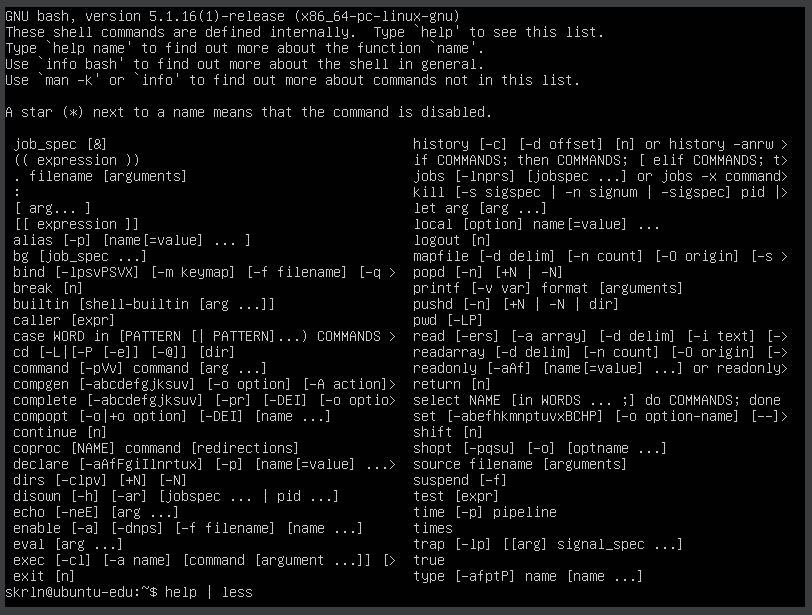
1. При помощи команды: **help** help ознакомьтесь со справкой по команде **help**. Зафиксируйте назначение команды **help**:



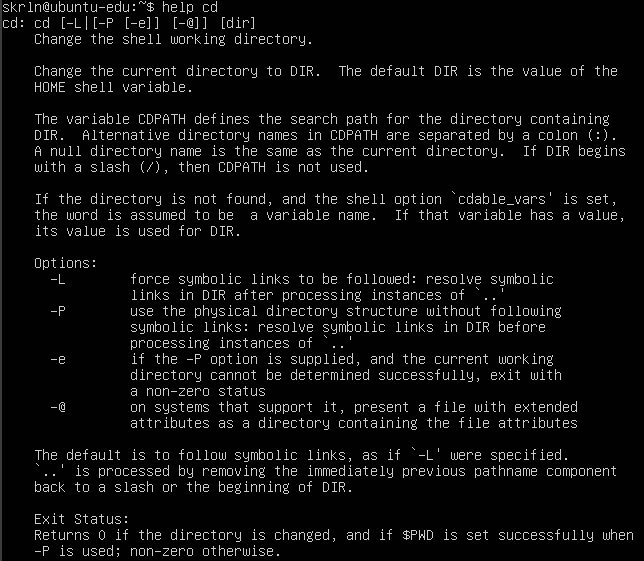
help — о встроенных в интерпретатор командах!

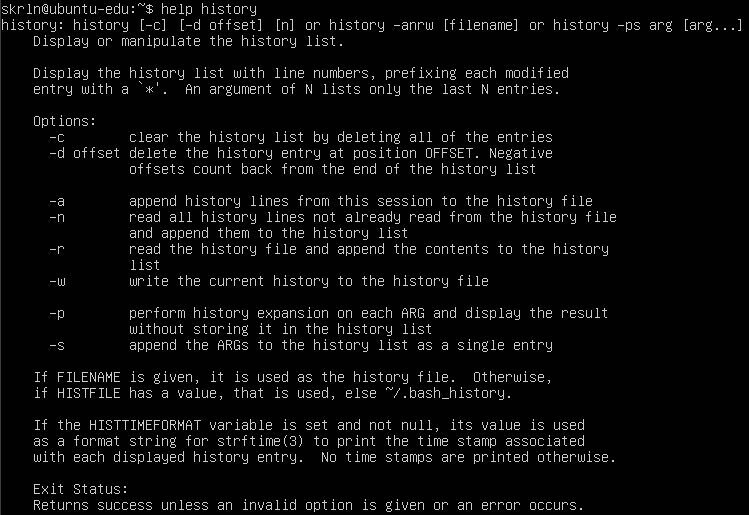
1. При помощи команды: **help** выведите список встроенных команд **bash**.

* (потерялся скрин)

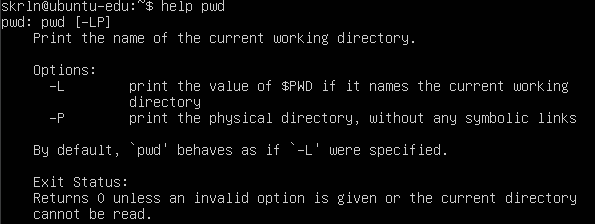


1. Посмотрите встроенную справку по командам: **cd**, **history**, **logout**, **pwd**. Зафиксируйте назначение этих команд:





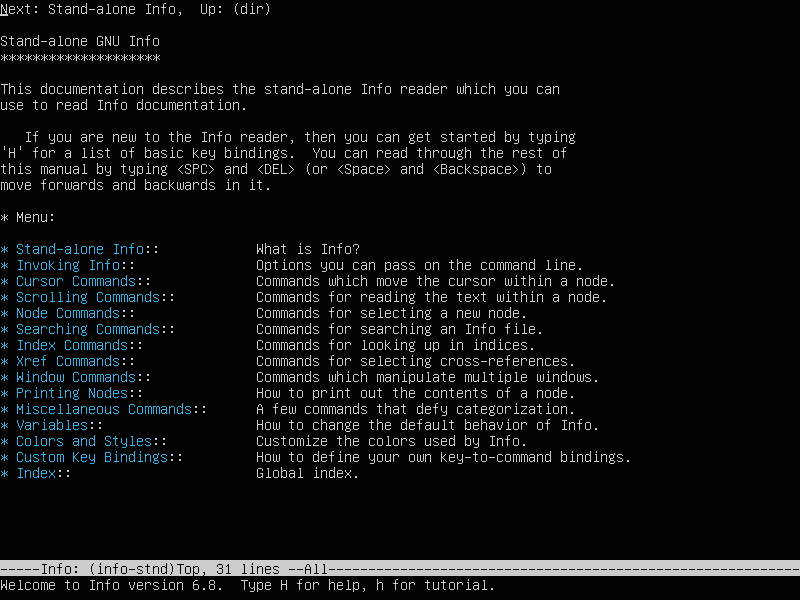




# Упражнение 2.3. Справочная система GNU Info

1. При помощи команды: **info** info ознакомьтесь со справкой по команде **info**. Примечание: в системах отличных от GNU данная справочная система может остсутствовать.

Зафиксируйте назначение команды **info**:



* Справочник с гиперссылками
* **INFOPATH** – переменная окружения, содержащая список каталогов, разделенных двоеточиями где будет производится поиск файлов **info.** Используется, если не задана опция **--directory.**
* texinfo?

1. Вызовите встроенную справку по программе GNU Info с помощью клавиши **h**. Ознакомьтесь с основами работы со справочной системой и зафиксируйте основные комбинации клавиш, используемые для навигации и поиска в справочной системе, а именно:
   1. клавиши построчного и постраничного листания вверх и вниз:



* 1. клавиши навигации по страницам справочной системы:



* 1. клавиши навигации по пунктам меню справочной системы:



* 1. клавиши навигации по перекрестным ссылкам справочной системы:



* 1. клавишу выхода из справочной системы:



* 1. клавиши поиска информации на странице справочной системы:



Лабораторная работа зачтена:

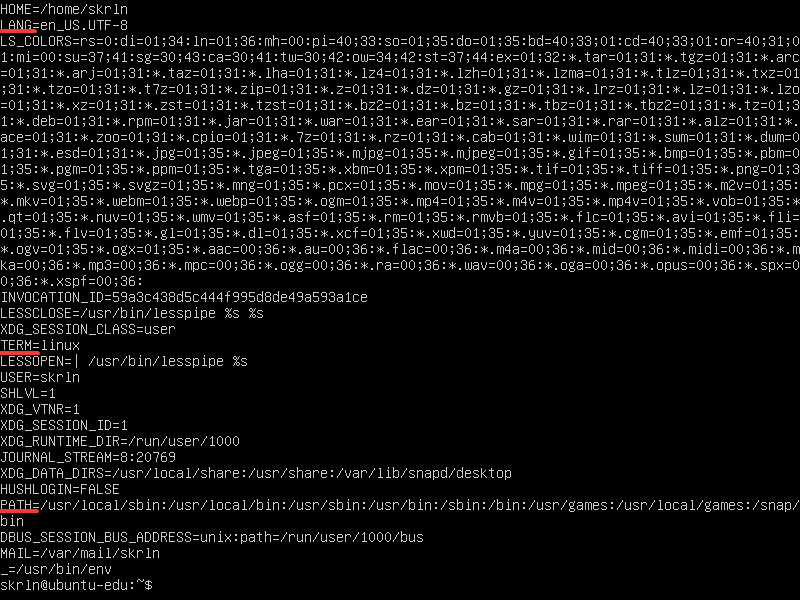
Дата:

Подпись преподавателя:

**Лабораторная работа 3. Пользовательское окружение. Пользовательский профиль**

# Упражнение 3.1. Переменные окружения. Переменные командного интерпретатора

1. Посмотрите значение переменных окружения с помощью команды **env**. Зафиксируйте значения переменных окружения PATH, LANG, TERM:

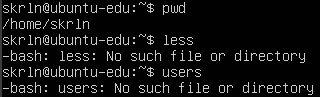


1. Установите в переменную окружения PATH пустое значение.



* Работают встроенные в интерпретатор!

1. Проследите за реакцией различных команд (**date**, **man**, ...). Зафиксируйте произошедшие изменения:

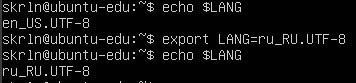


1. Восстановите старое значение переменной окружения PATH.

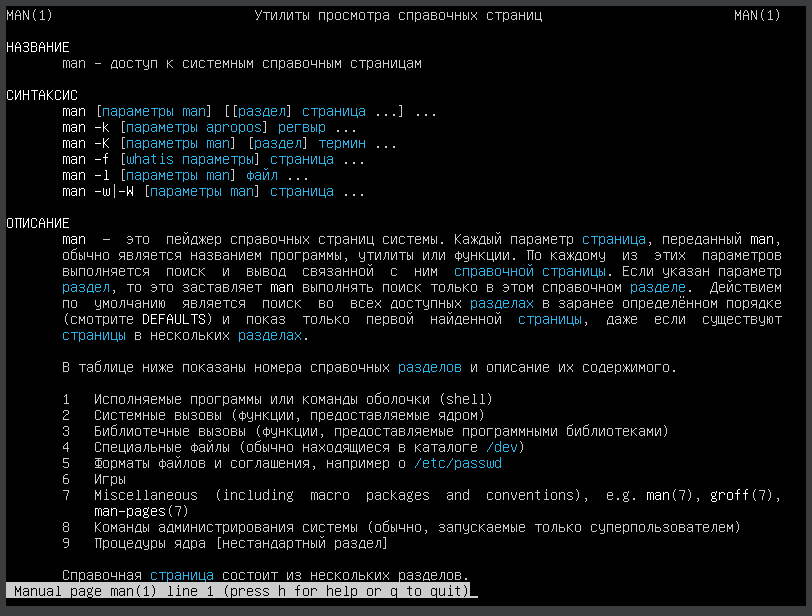
* logout

1. Установите в переменную окружения LANG значение любого языка в формате **язык\_[СТРАНА.[кодировка]]**.

* Раскомментировал ru\_RU.UTF-8 в /etc/locale.gen
* sudo locale-gen



1. Проследите за реакцией различных команд (**date**, **man**, ...). Зафиксируйте произошедшие изменения:





1. Восстановите старое значение переменной окружения LANG.
2. Установите в переменную окружения TERM значение **vt100**.
3. Проследите за реакцией различных команд (**mcedit**, **man**, ...). Зафиксируйте произошедшие изменения:

* mcedit поменял цвет фона! и f10 не работает ☹

1. Восстановите старое значение переменной окружения TERM.
2. Посмотрите значение внутренних переменных командного интерпретатора **bash** с помощью команды **set**. Зафиксируйте значение переменной PS1:



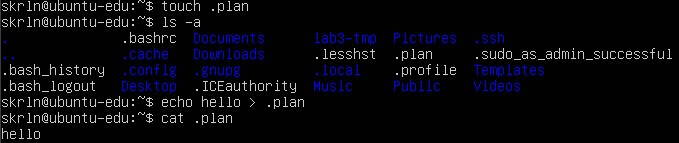


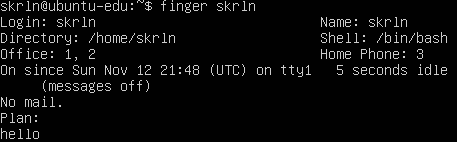
1. Установите во внутреннюю переменную PS1 любое значение.
2. Проследите за реакцией командного интерпретатора. Зафиксируйте произошедшие изменения:



# Упражнение 3.2. Конфигурационные файлы пользовательского окружения. Пользовательский профиль

1. Создайте файл .plan в домашнем каталоге пользователя. Запишите в него любую информацию.
2. При помощи команды **finger** посмотрите и зафиксируйте изменения в свойствах пользователя student:





1. Отредактируйте файл .bashrc или .bash\_profile в домашнем каталоге пользователя. Примечание: если используется интерпретатор отличный от bash, то данные файлы не будут считаны, используйте .profile или специфичный для вашего интерпретатора файл конфигурации.

Запишите в файл команду присвоения в переменной окружения LANG значение любого языка в формате **язык\_[СТРАНА.[кодировка]]**:



1. Завершите сеанс пользователя. Войдите в систему с использованием учетной записи student.
2. Проследите за реакцией различных команд (**date**, **man**, ...). Зафиксируйте произошедшие изменения:

* Теперь язык сохранятся после логаута!

1. Отредактируйте файл .bash\_profile в домашнем каталоге пользователя. Запишите в него команду установки в переменную PS1 собственного cтрокового значения для приглашения командного интерпретатора:
2. Завершите сеанс пользователя. Войдите в систему с использованием учетной записи student.
3. Проследите за реакцией командного интерпретатора. Зафиксируйте произошедшие изменения:



Лабораторная работа зачтена:

Дата:

Подпись преподавателя:

**Лабораторная работа 4. Файлы и файловая система UNIX**

# Упражнение 4.1. Изучение структуры файловой системы

При помощи следующих команд: **pwd**, **cd**, **ls** отработайте основные операции навигации в файловой системе:

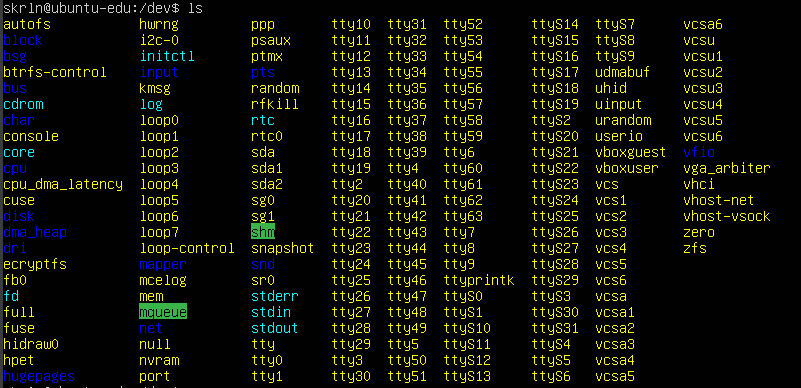


1. Перейдите в корневой каталог.



1. Спускаясь по дереву каталогов, пройдитесь по основным каталогам и подкаталогам, и ознакомьтесь с их содержанием.





1. Зафиксируйте структуру иерархии файловой системы и назначение основных каталогов:
2. bin - бинарные версии файлов (в том числе для командной оболочки)
3. dev - псевдофайлы, представляющие собой аппаратные средства, подключённые к устройству
4. boot – файлы для загрузки системы
5. cdrom – временное место для подключенных устройств
6. etc - большая часть конфигурационных файлов
7. home – домашние директории пользователя
8. lib(32/64) – библиотеки для системных приложений.
9. lost+found – файлы, восстановленные после сбоя работы системы, т.е. файлы после проверки диска
10. media – каталог для съемных накопителей
11. mnt – каталог для съемных накопителей
12. proc – виртуальная файловая система, информация о запущенных процессах
13. root – домашняя директория суперпользователя
14. run – информация о системе с момента ее загрузки
15. sbin – утилиты, которые нужны для настройки и администрирования системы суперпользователя (нужны администратору)
16. snap
17. srv – server, информация о запущенных серверах
18. sys – виртуальная файловая система, содержит информацию об устройствах
19. opt - необязательные компоненты системы или приложения.
20. tmp - временные файлы
21. usr - пользовательские компоненты
22. var - файлы приложений (в том числе системные журналы, кэши и т. д.)

# Упражнение 4.2. Типы файлов

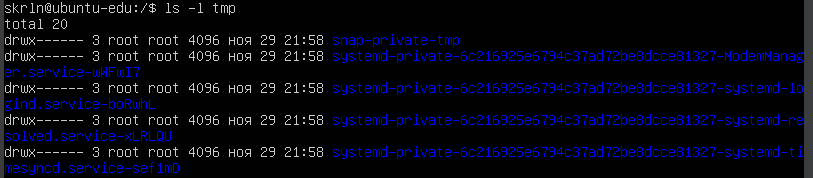
1. Для экземпляров различных типов файлов:
   1. обычный файл (/etc/fstab, /bin/bash, /usr/bin/apropos)



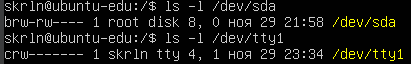




* 1. каталог (/tmp)



* 1. специальный файл устройства (/dev/sda, /dev/tty1)



* 1. именованный канал (/run/systemd/initctl/fifo)



* 1. сокет (/run/systemd/journal/dev-log)



* 1. символическая ссылка (/dev/cdrom)



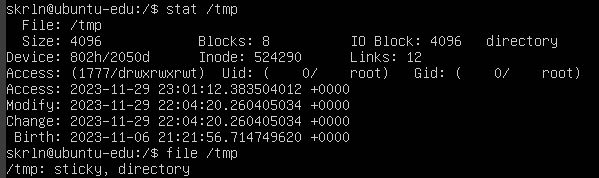
Примечание: при отсутствии файлов из перечня найдите аналог соответствующего типа.

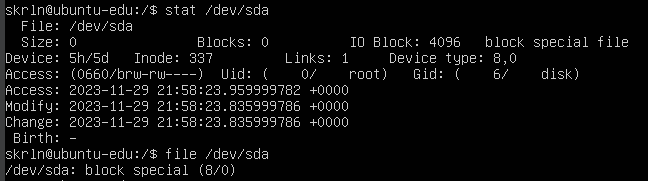
1. Запишите особенности отображения различных типов файлов командой **ls** -l:

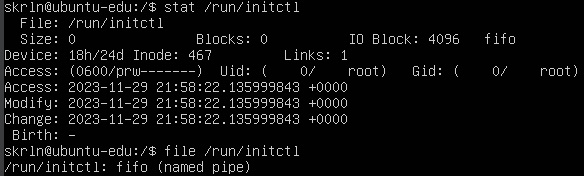
* обычный файл -
* каталог d
* специальный файл устройства b (блочные)
* специальный файл устройства c (символьные)
* именованный канал p
* сокет s
* символическая ссылка l

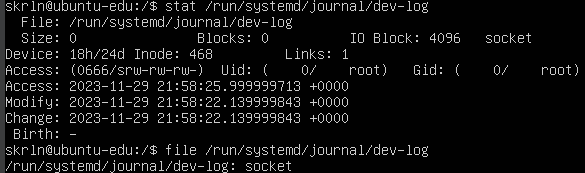
1. Используя команды: **stat**, **file** исследуйте метаданные и содержимое различных типов файлов (из Шаг 1):

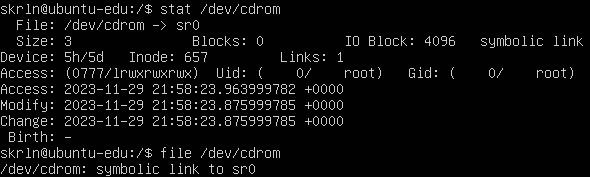












# Упражнение 4.3. Манипулирование объектами дерева каталогов UNIX

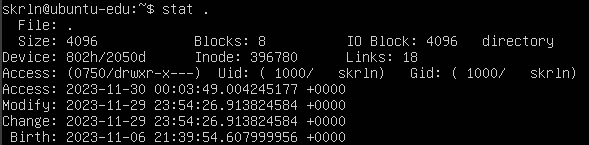
При помощи следующих команд: **touch**, **mkdir**, **rmdir**, **cp**, **mv**, **rm**, **ln**, **mkfifo**, **mknod** отработайте основные операции работы с файловой системой, зафиксируйте полученные результаты:



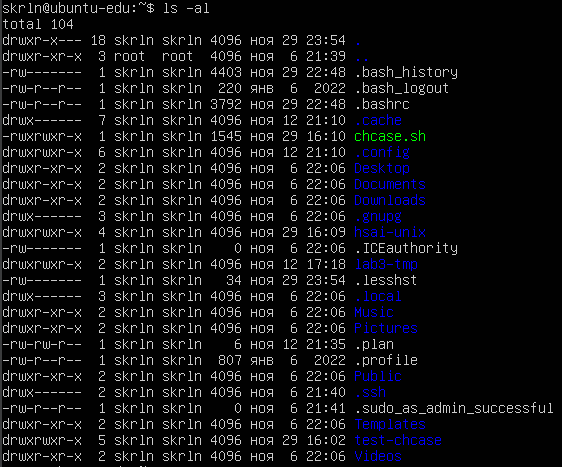
1. перейдите в домашний каталог пользователя:



1. получите информацию о текущем рабочем каталоге:



1. получите информацию о содержимом текущего каталога:



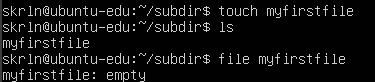
1. создайте новый подкаталог subdir в текущем каталоге:



e. перейдите в него:

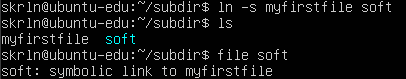


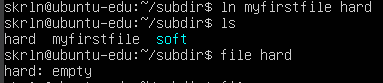
1. создайте в данном подкаталоге новый файл myfirstfile:



1. создайте мягкую symblink и жесткую hardlink ссылки на него:

* The inode and file data are permanently deleted when the number of hard links is zero.
* if the original file gets deleted, the hard link still refers to the location of the data in the hard drive, therefore the information is still saved.
* If the symbolic link file gets deleted, nothing happens to the original file. If the original file gets deleted, the symbolic link file will then point to a file that does not exist.





1. переместите и/или скопируйте файл myfirstfile в вышележащий каталог:

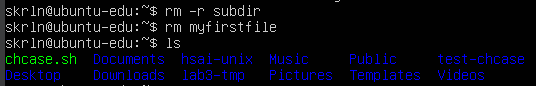




i. удалите оставшиеся файлы:



1. перейдите в вышележащий каталог и удалите созданный вами подкаталог subdir и файл myfirstfile:



1. создайте именованный канал myfifo:



1. создайте образ оптического диска, находящийся в накопителе /dev/cdrom:



# Упражнение 4.4. Поиск файлов

При помощи следующих команд: **find**, **whereis**, **which** отработайте основные операции поиска файлов, зафиксируйте полученные результаты:



1. найдите в каталоге /usr все файлы размером больше 800Кb (пропал скрин с командой)

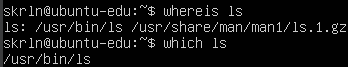




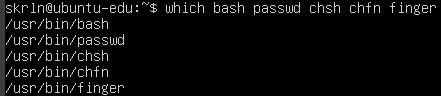
!!!

1. найдите информацию о всех исполняемых файлах, страницах руководства и исходных текстах программы **ls**

* whereis ищет бинарники и заголовки руководства
* which – только бинарники
* The which command performs searches in the PATH variable while the whereis command searches standard Linux directories, including $PATH and $MANPATH.

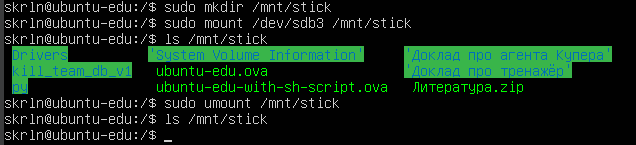


1. найдите информацию о местоположении программ **bash**, **passwd**, **chsh**, **chfn**, **finger**



# Упражнение 4.5. Операции с файловыми системами

1. При помощи следующих команд: **mount**, **umount**. Примечание: может потребоваться повышение привелегий до суперпользователя:
   1. примонтируйте файловую систему находящуюся на оптическом диске или флешнакопителе, в каталог /mnt, проследите за сообщениями:



b. размонтируйте находящуюся на оптическом диске или флеш-накопителе файловую систему, примонированную в каталог /mnt, и проследите за сообщениями:

1. При помощи следующих команд: **du**, **df** подсчитайте:

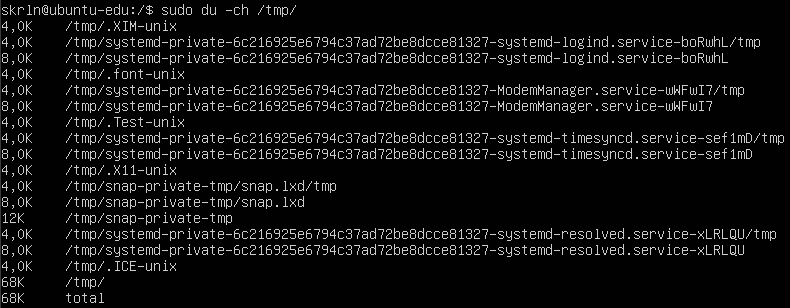
***du*** *- позволяет вывести размер всех файлов в определённой папке в байтах или в более удобном формате*

***df*** *- команда df позволяет задействовать одноименную утилиту, предназначенную для вывода информации об объеме свободного пространства в каждом из смонтированных разделов*

* 1. место, занимаемое на диске домашнего каталога пользователя:



* 1. место, занимаемое на диске каталогом /tmp



* 1. размер корневой файловой системы (было через du)



* 1. размер файловой системы оптического диска или флеш-накопителя



* 1. свободное место, оставшееся на корневой файловой системе



Лабораторная работа зачтена:

Дата:

Подпись преподавателя:

**Лабораторная работа 5. Владельцы и права доступа объектов файловой системы UNIX.**

# Упражнение 5.1. Владельцы файлов

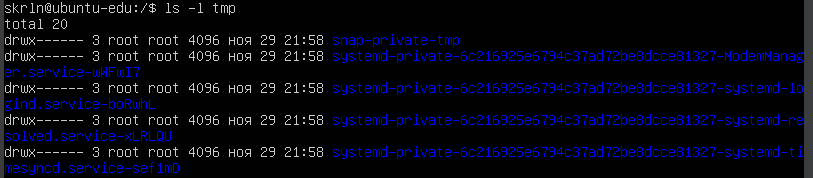
1. Зафиксируйте права доступа к файлу и владельцев файлов из Шаг 1 упр. 4.2, лаб. 4:
   1. обычный файл (/etc/fstab, /bin/bash, /usr/bin/apropos)



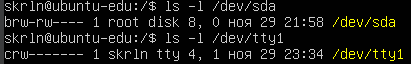




* 1. каталог (/tmp)



* 1. специальный файл устройства (/dev/sda, /dev/tty1)



* 1. именованный канал (/run/systemd/initctl/fifo)



* 1. сокет (/run/systemd/journal/dev-log)



* 1. символическая ссылка (/dev/cdrom)



1. При помощи команд: **chown**, **chgrp** попробуйте подарить файлы из домашнего каталога пользователя *student* пользователю *vinnie*, группе *root*. Зафиксируйте реакцию системы:







 (от skrln)

# Упражнение 5.2. Права доступа к объектам файловой системы UNIX

1. Создайте новый файл newfile. Зафиксируйте права доступа к файлу и владельцев полученного файла:



1. При помощи команды: **umask**:

a. просмотрите и зафиксируйте значение реверсной (не забыть вычесть из 666) маски прав доступа по умолчанию для вновь создаваемых файлов



b. установите значение реверсной маски прав доступа по умолчанию в 000. Создайте файл defaultfile. Зафиксируйте права доступа к файлу и владельцев полученного файла:

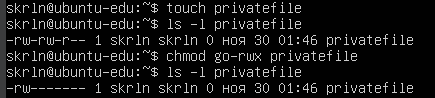


c. установите значение реверсной маски прав доступа по умолчанию в 777. Создайте файл maskedfile. Зафиксируйте права доступа к файлу и владельцев полученного файла:

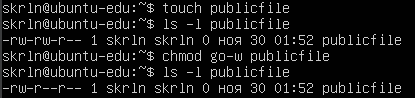


1. При помощи команды: **chmod** отработайте основные операции изменения владельцев и прав доступа к файлам:
   1. создайте в домашнем каталоге файл privatefile и установите права доступа rw- ---

---:



* 1. создайте в домашнем каталоге файл publicfile и установите права доступа rw- r- r--:



* 1. создайте в домашнем каталоге новый каталог groupsubdir и установите права доступа rw- r-- ---:





d. создайте в домашнем каталоге новый каталог publicdir и установите права доступа rw- rw- rw-:





Лабораторная работа зачтена:

Дата:

Подпись преподавателя:

**Лабораторная работа 6. Обработка текстовой информации. Управление вводом-выводом команд. Подстановки командного интерпретатора**

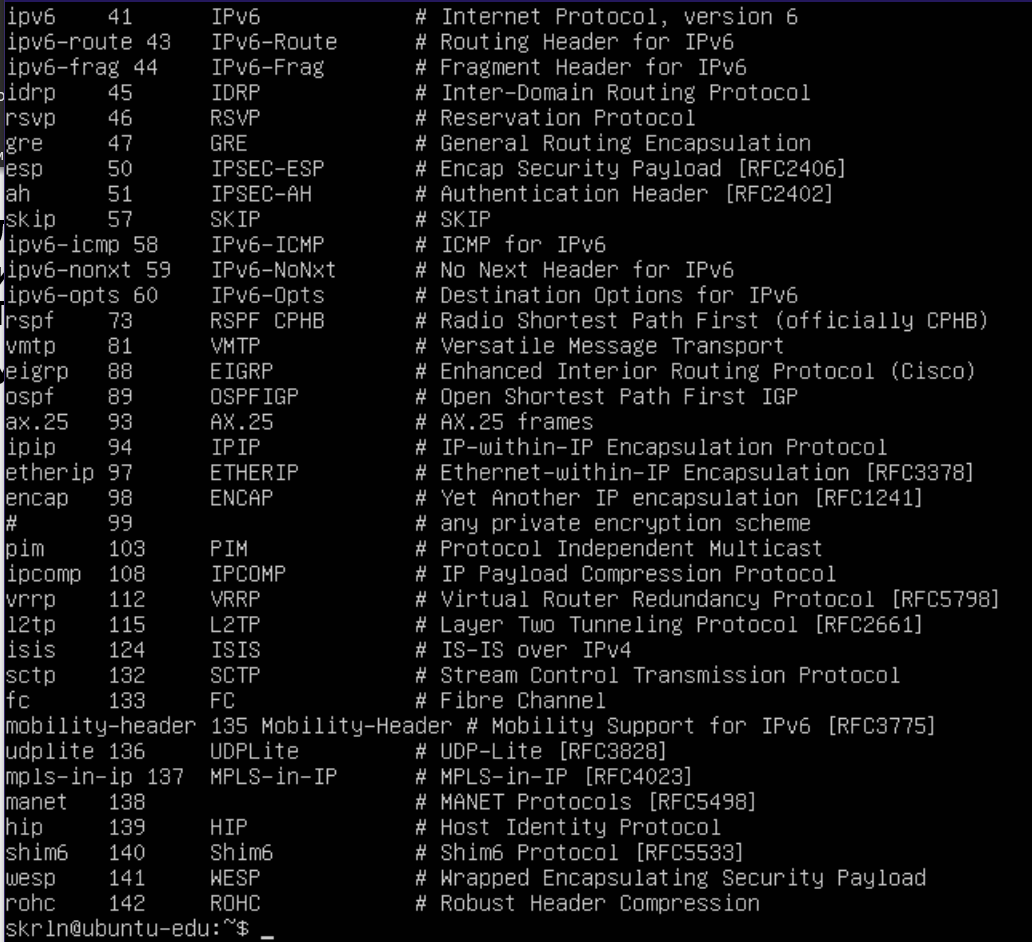
# Упражнение 6.1. Обработка текстовой информации в UNIX

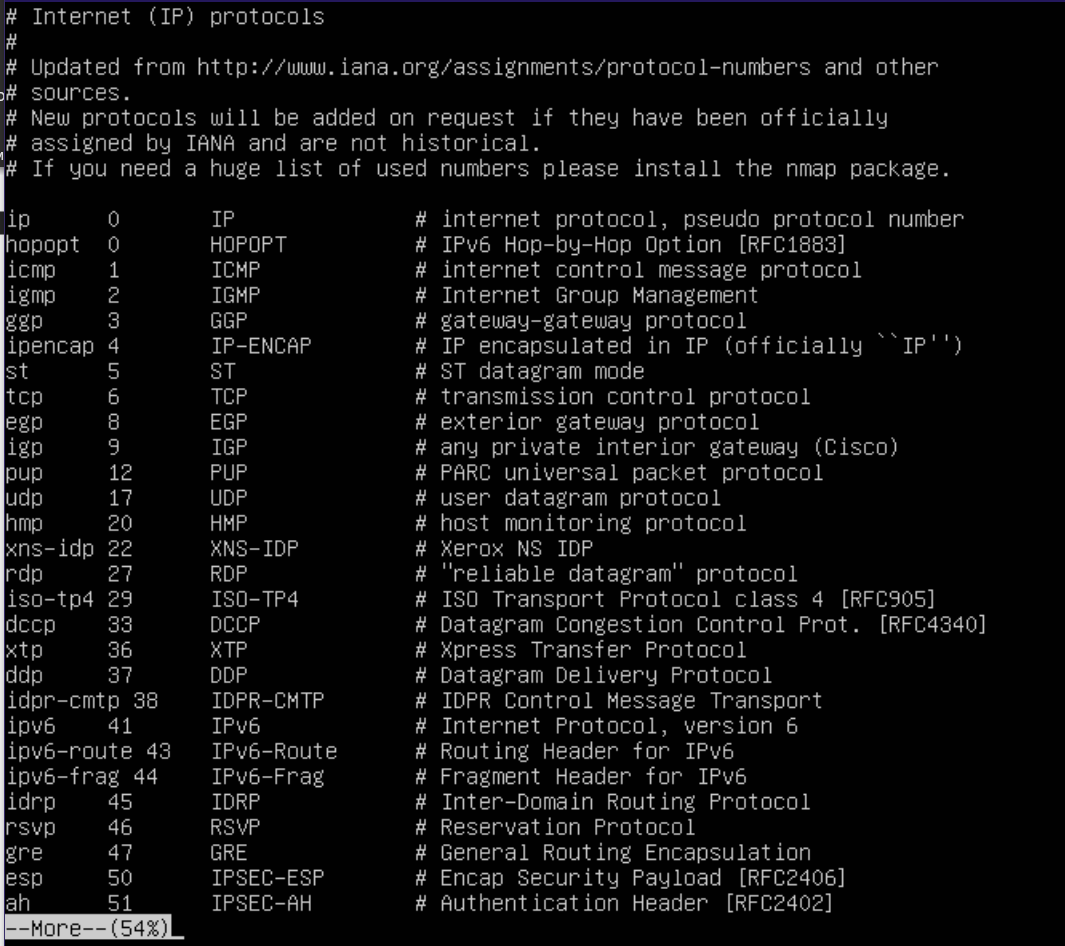
При помощи следующих команд: **cat**, **less**, **more**, **tail**, **head**, **od**, **hexdump**, **cmp**, **comm**, **diff**, **cut**, **grep**, **sort**, **wc**, **tr** отработайте основные операции работы с содержимым файлов, зафиксируйте полученные результаты:



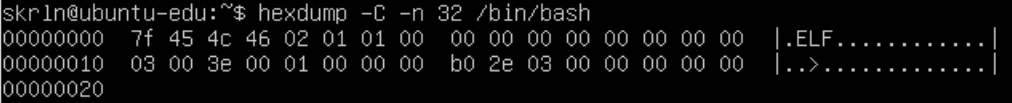
1. выведите текстовый файл (/etc/protocols) на экран полностью и по частям:

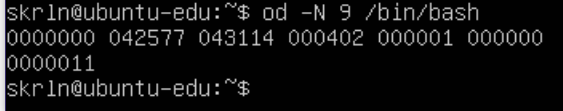
* cat /etc/protocols - полностью
* less/more/tail/head - по частям



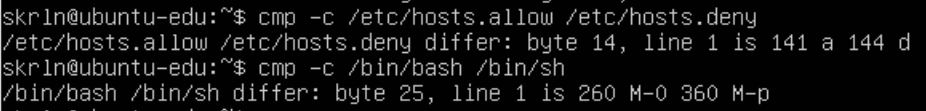


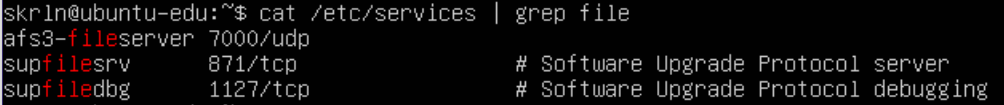
1. выведите двоичный файл (/bin/bash) в шестнадцатеричном и восьмеричном виде:





c. сравните текстовые (/etc/hosts.allow и /etc/hosts.deny) и двоичные (/bin/bash и /bin/sh) файлы:



d. найдите все строки файла (/etc/services), содержащие слово **file**, затем строки содержащие четырехзначные числа, затем строки не начинающиеся со знака **#**: 



A screenshot of a computer

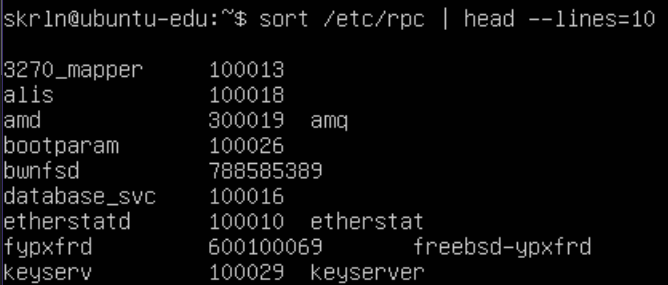
Description automatically generated



1. подсчитайте количество строк/слов/букв в файле (/etc/protocols):

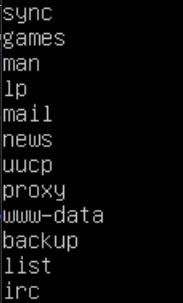


1. отсортируйте содержимое файла (/etc/rpc):



1. извлеките из структурированного файла (/etc/passwd) первое поле всех строк:





# Упражнение 6.2. Перенаправление ввода-вывода информации

При помощи конструкций перенаправления: **>**, **<**, **>>**, **|** и соответствующих команд, отработайте основные операции конвейерной обработки информации:



1. вырежьте из файла **/etc/passwd** пятое поле, отсортируйте в обратном порядке и сохраните в файл **users**

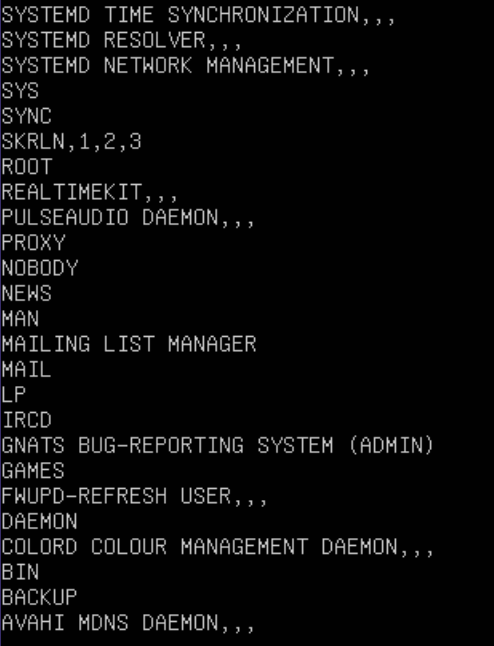




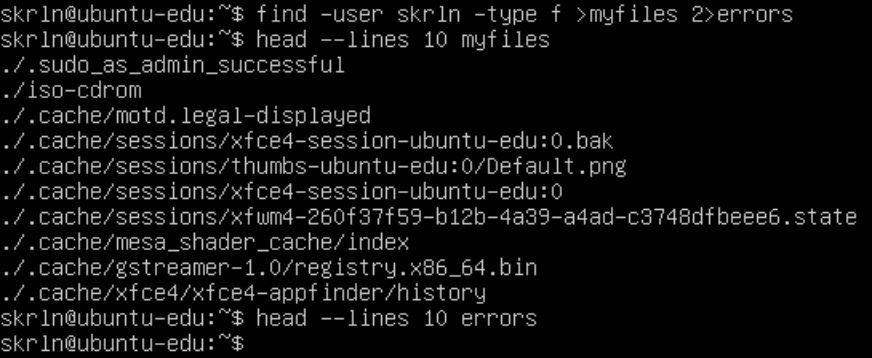
* Полное имя пользователя (GECOS) (или другая доп. информация)

1. преобразуйте содержимое файла **users** к верхнему регистру

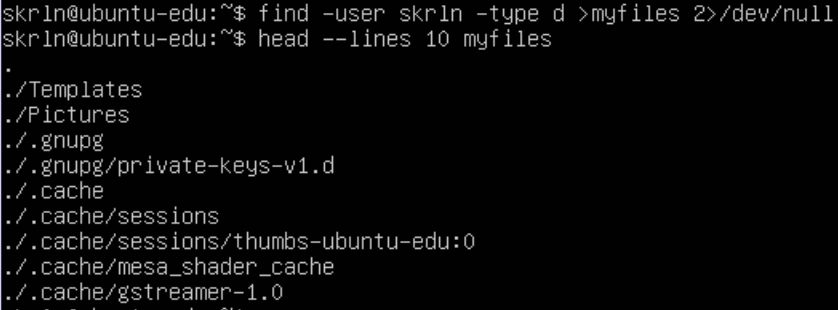




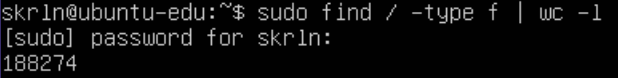
c. найдите все регулярные файлы в принадлежащие пользователю **student** и сохраните найденный список в файле **myfiles**, ошибки поиска сохраните в файле **errors**



d. найдите все каталоги принадлежащие пользователю **student** и добавьте полученный список в файл **myfiles**, подавите вывод сообщений об ошибках поиска



e. подсчитайте суммарное количество обычных файлов в дереве каталогов UNIX



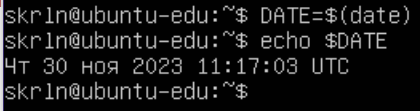
# Упражнение 6.3. Подстановка вывода команд

При помощи конструкций подстановки вывода команд **``**, **$()** отработайте основные операции подстановки вывода команд:

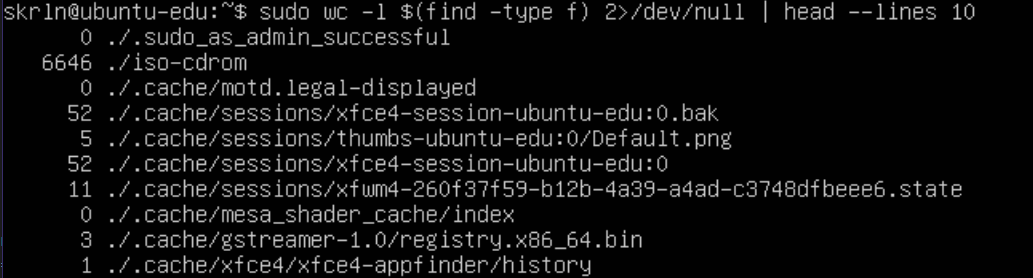


When the old-style backquoted form of substitution is used, backslash retains its literal meaning except when followed by "$", "`", or "\". The first backticks not preceded by a backslash terminates the command substitution. When using the "$(COMMAND)" form, all characters between the parentheses make up the command; none are treated specially.

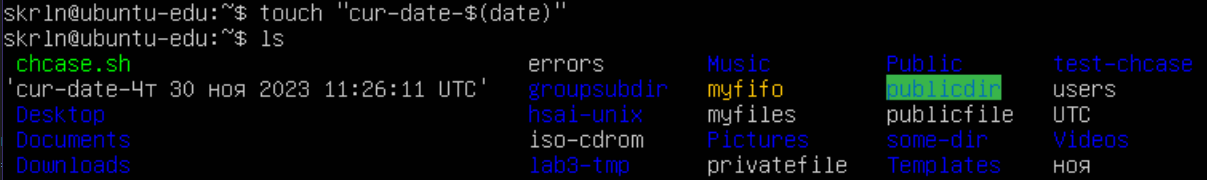
a. Сохраните в переменной DATE текущую дату и время



b. Получите информацию о суммарном количестве строчек во всех регулярных файлах домашнего каталога пользователя:



1. создайте файл, в имени которого присутствует текущая дата и время.

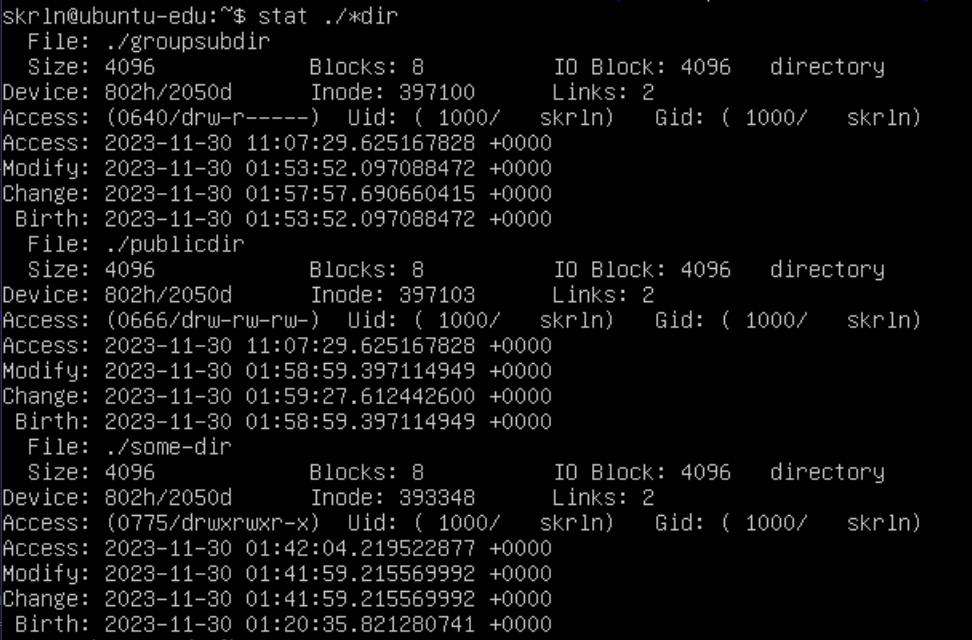


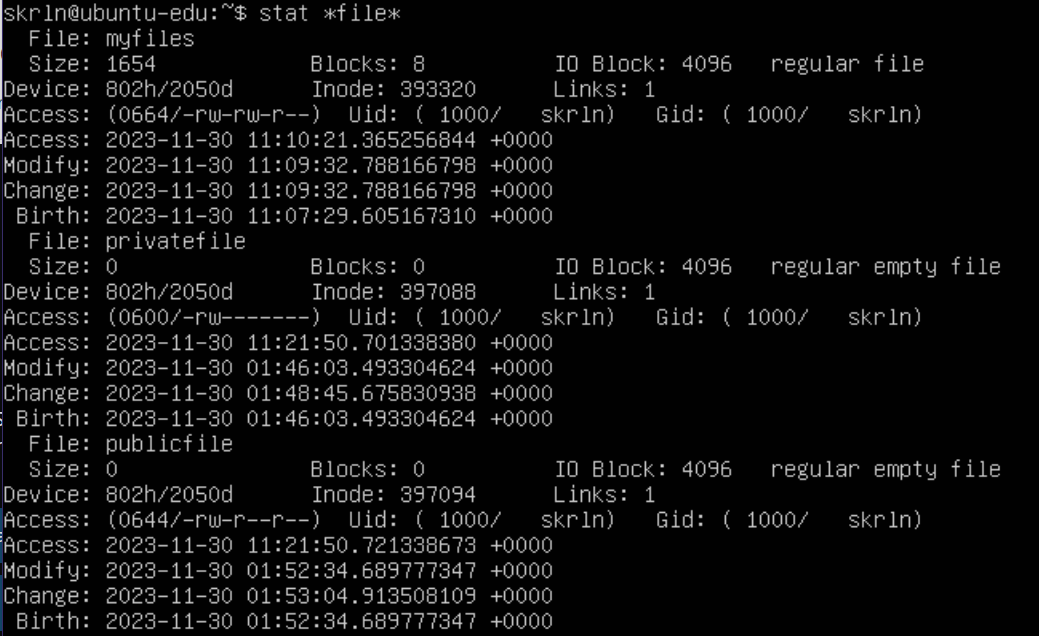
# Упражнение 6.4. Подстановка шаблонных символов

При помощи шаблонных символов: **\***, **?** и соответствующих команд:

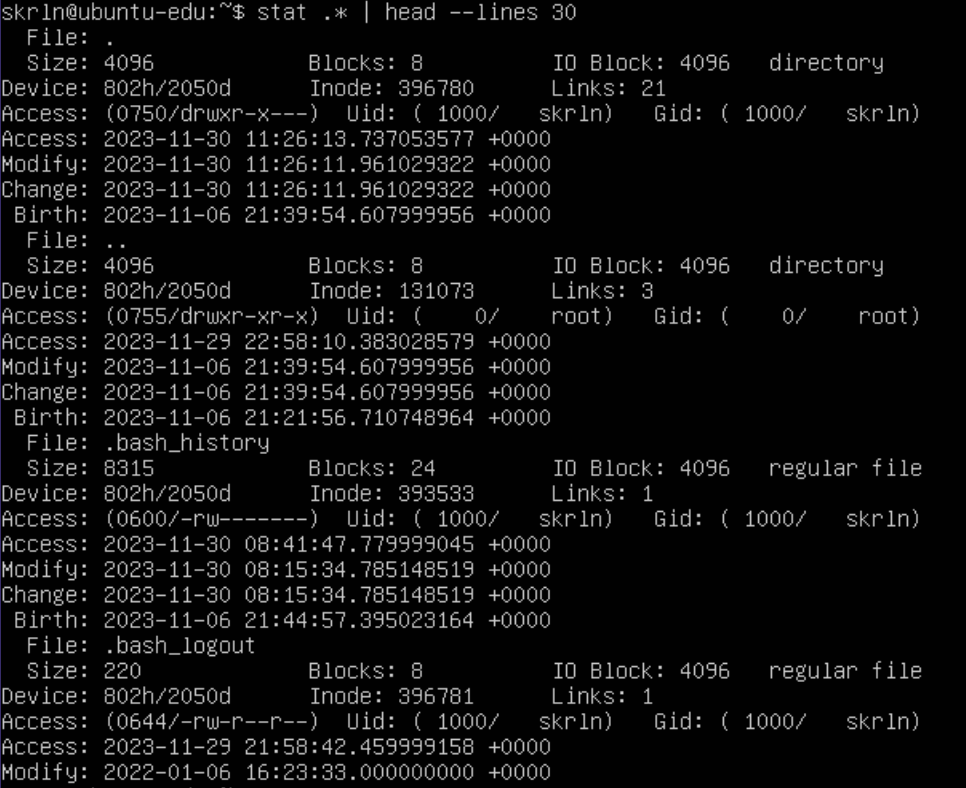


1. выведите полную информацию о подкаталогах домашнего каталога пользователя, заканчивающихся символами **dir** и о файлах содержащих в своем названии подстроку **file**:

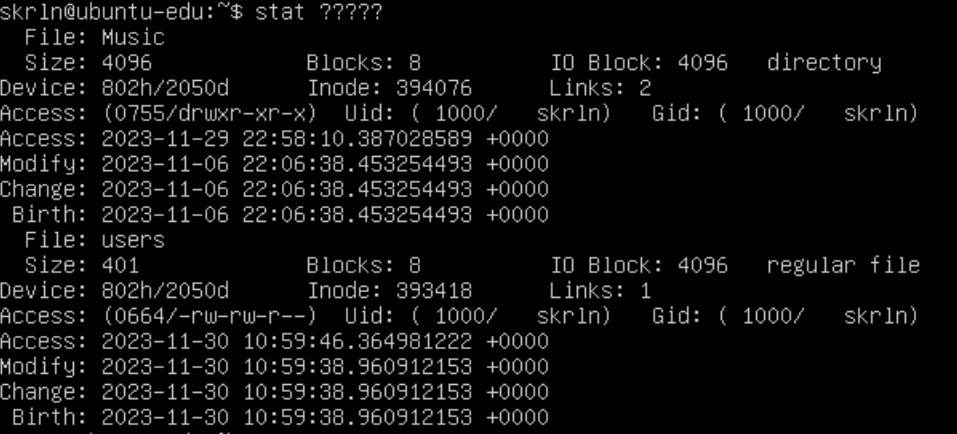




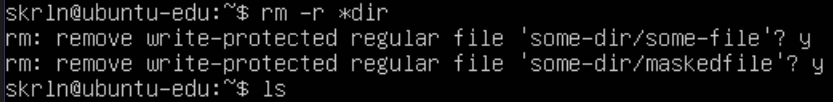
1. выведите полную информацию о объектах домашнего каталога пользователя, имена которых начинаются с точки.



1. выведите полную информацию о объектах домашнего каталога пользователя, имеющих длину имени ровно пять символов:



1. удалите подкаталоги домашнего каталога пользователя, заканчивающихся символами **dir** и файлы содержащие в своем названии подстроку **file**:





Лабораторная работа зачтена:

Дата:

Подпись преподавателя:

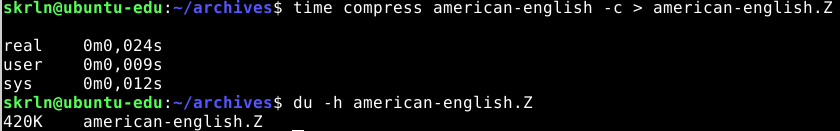
**Лабораторная работа 7. Инструментальные средства UNIX**

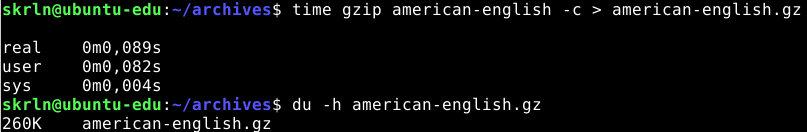
# Упражнение 7.1. Упаковщики и архиваторы UNIX

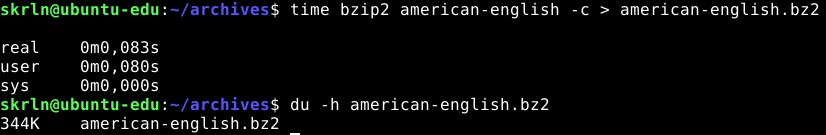
1. При помощи команд упаковки **compress**, **uncompress**, **gzip**, **gunzip**, **bzip2, lzma** упакуйте один и тот же файл (например, **/usr/share/dict/american-english**, скопировав предварительно в домашний каталог) несколько раз. Примечание 1: если файл для сжатия отсутствует, то скачайте любой словарь для проверки орфографии из сети интернет. Примечание 2: если отсутствуют команды упаковки установите их дополнительно через менеджер управления пакетами.

Сравните скорость и качество упаковки:







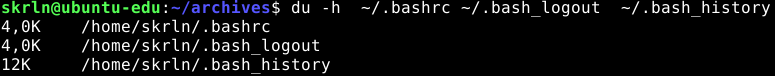


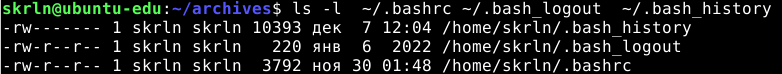


|  |  |  |  |
| --- | --- | --- | --- |
| Архиватор | Время, с | Получившийся размер, Кб | Сжатие |
| compress | 0,024 | 420 | 2,3 |
| gzip | 0,089 | 260 | 3,7 |
| bzip2 | 0,083 | 344 | 2,8 |
| lzma | 0,286 | 204 | 4,7 |

1. При помощи команды архивации **tar** выполните несколько операций архивирования:

a. создайте архив dotfiles.tar содержащий файлы .bashrc, .bash\_profile, .bash\_history, сравните размер архива и суммарный размер файлов:



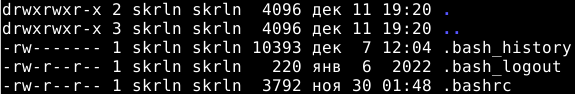






1. разархивируйте полученный архив dotfiles.tar в каталог dotfiles, сравните атрибуты (разрешения, владельцев, время создания) исходных файлов, и файлов изархива:





c. создайте упакованный архив (непосредственно командой **tar**) dotfiles.tar.gz, сравните его размер с размером неупакованного архива dotfiles.tar и суммарным размером файлов архива:



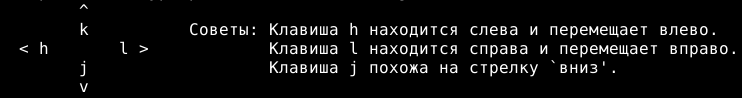
1. разархивируйте упакованный архив dotfiles.tar.gz в каталог dotfiles-:





# Упражнение 7.2. Текстовые редакторы UNIX

1. При помощи команды **vimtutor** **ru** ознакомьтесь с принципами работы в редакторе **vim** и зафиксируйте основные команды:
   1. команды посимвольного перемещения курсора:



* 1. команды пословного перемещения курсора:
* <n>w – к началу n-ного слова от нынешнего
* <n>e – к концу -//-
* 0 (или ^) – в начало строки
* $ -- в конец строки

Можно по строкам файла: <n> G (ctrl g – текущая строка)

* 1. команды постраничного листания:
* ctrl f ->
* ctrl b <-

d. команды поиска текста:

* / <слово>
* <- ->

N n

* ? -- поиск сразу в обратном направлении
* Ctrl O – в начало поиска
* % -- поиск парной строки

e. команды повторения предыдущего поиска:

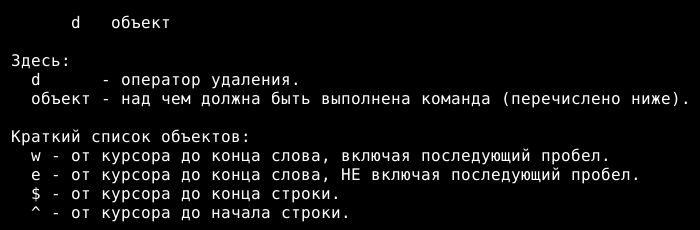
* <- ->

N n

f. команды вставки текста:

* i – вставка перед курсором
* a – вставка после курсора
* A – вставка в конец строки
* p – вставка после удаления
* r <s> -- вставка <s> вместо символа под курсором; R -- несколько
* ce – удаляет от курсора до конца слова и переводит в режим вставки
* с<n><o> -- удаляет n объектов и переводит в режим вставки

g. команды удаления текста:

* x – символ под курсором
* dw – от курсора до конца слова
* d$ -- от курсора до конца строки
* 
* 

!!!

* dd -- строку

1. команды захвата текста:

* визуальный режим v -> y
* y<n>w

i. команды вставки захваченного текста:

* p

j. команды записи текста в файл:



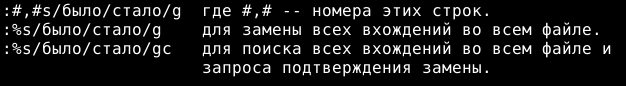
k. команды чтения файла:



l. команды выхода из редактора:

* :q! – без сохранения
* :wq – с сохранением
* u – откат, U – откат строки, CTRL-R – откат отката
* Замена:

 -- первое вхождение (g – во всей строке)



* 
* o/O – пустая строка под/над курсором

При помощи редактора **vim** отредактируйте файл .plan, куда внесите информацию о себе.



# Упражнение 7.3. Файловый менеджер UNIX - Midnight Commander

При помощи файлового менеджера **mc** выполните пункты лаб. 4 и лаб. 5.

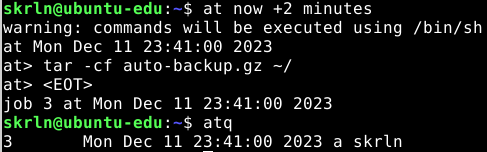


A screenshot of a computer

Description automatically generated

# Упражнение 7.4. Подсистемы отложенных и периодических заданий.

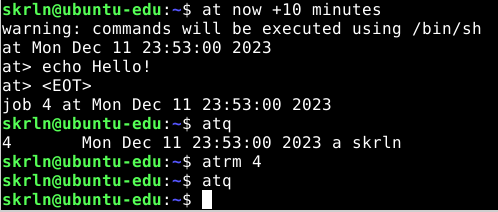
1. При помощи команд **at**, **atq**, **atrm**:
   1. установите отложенное (на пять минут вперед) задание для создания упакованного архива в домашнем каталоге с именем autoarchive-backup.tgz, содержимого которого включает весь домашний каталог пользователя:



* 1. удостоверьтесь, что отложенное задание создано, проследите за изменениями:
  2. дождитесь времени выполнения отложенного задания, удостоверьтесь в корректности его выполнения:

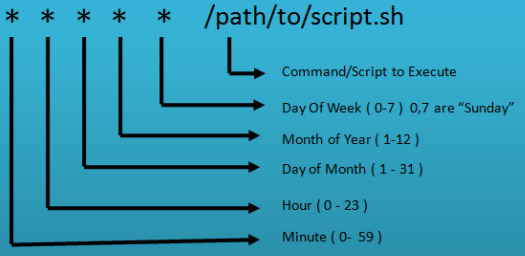


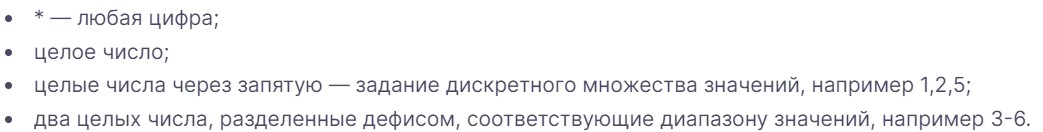
d. создайте любое отложенное задание, затем удалите его, проследите за сообщениями:



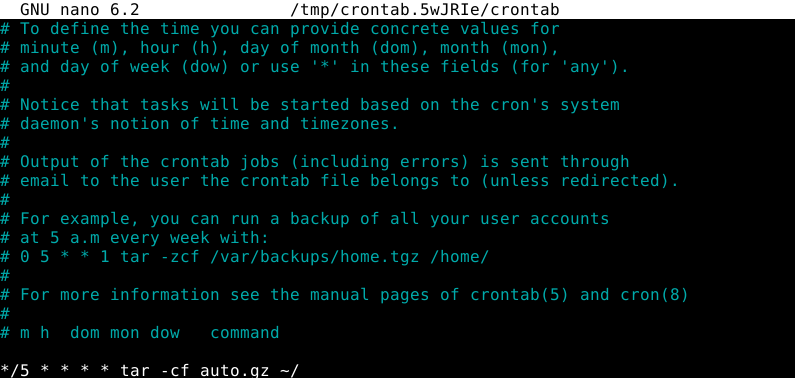
e. удостоверьтесь, что отложенное задание удалено, проследите за изменениями:

1. При помощи команды **crontab**:
   1. установите периодическое (с шаблоном времени выполнения каждые пять минут) задание для создания упакованного архива в домашнем каталоге с именем autoarchive-backup.tgz, содержимого которого включает весь домашний каталог пользователя:

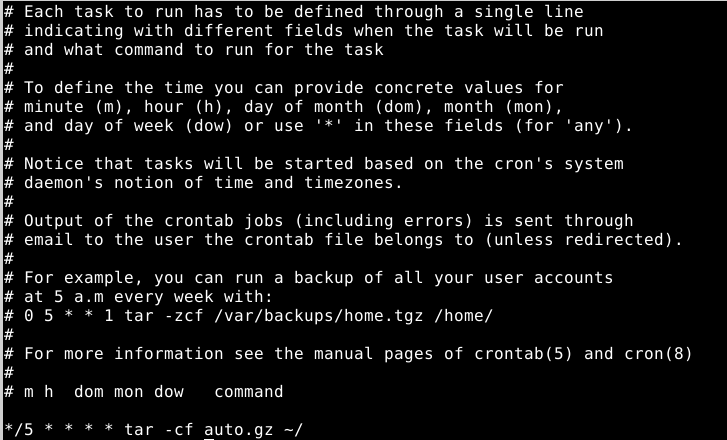




* crontab -e



* 1. удостоверьтесь, что периодическое задание создано, проследите за сообщениями:
* crontab -l



1. дождитесь времени выполнения периодического задания, удостоверьтесь в корректности его периодического выполнения:



1. удалите установленное периодическое задание, проследите за сообщениями:



1. удостоверьтесь, что периодическое задание удалено, проследите за изменениями:

Лабораторная работа зачтена:

Дата:

Подпись преподавателя:

**Лабораторная работа 8. Процессы UNIX**

# Упражнение 8.1. Типы процессов

1. При помощи команды **ps** aux найдите среди всех процессов операционной системы:

* a – для всех пользователей
* u – формат для пользователя
* x – включая без tty

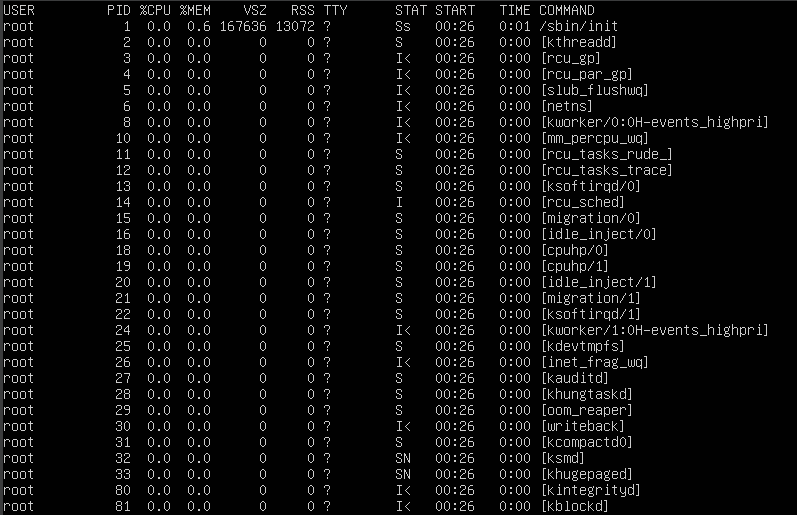
a. системные процессы

**VSZ** is Virtual Memory Size. This is the size of memory that Linux has given to a process, but it doesn’t necessarily mean that the process is using all of that memory. For example, many applications have functions to carry out certain tasks, but may not load them into memory until they are needed. Linux utilizes demand paging, which only loads pages into memory once the application makes an attempt to use them.

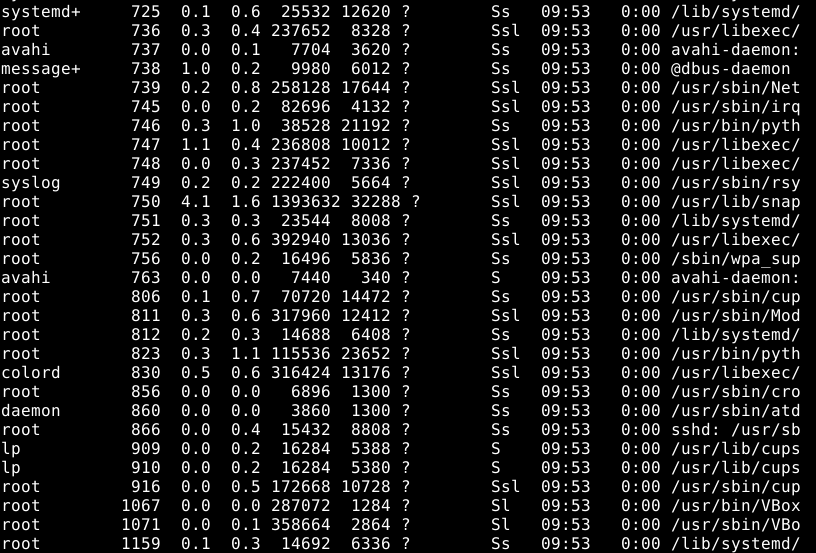
The VSZ size you see has taken all of these pages into consideration, but it doesn’t mean they’ve been loaded into physical memory. The VSZ size is therefore not usually an accurate measurement of how much memory a process is using, but rather an indication of the maximum amount of memory a process can use if it loads all of its functions and libraries into physical memory.

**RSS** is Resident Set Size. This is the size of memory that a process has currently used to load all of its pages. At first glance, it may seem like the RSS number is the real amount of physical memory that a system process is using. However, shared libraries are counted for each process, making the reported amount of physical memory usage less accurate.

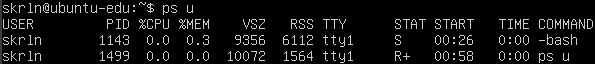
Память == 0



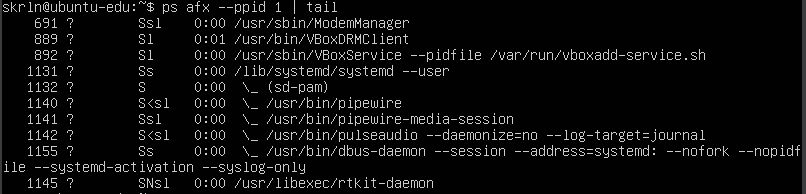
b. демоны TTY == ? и память != 0

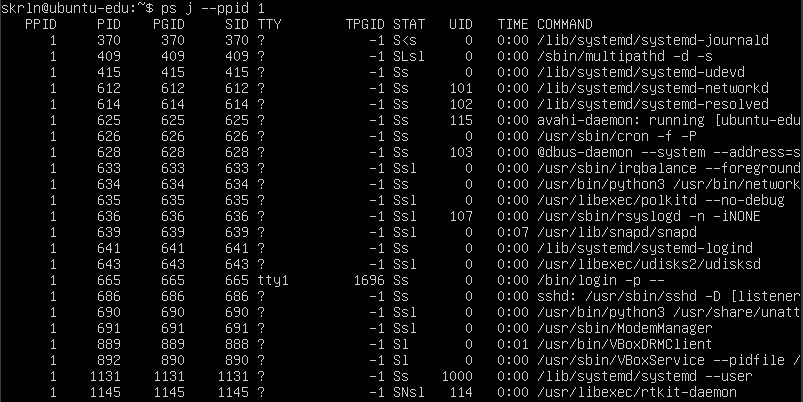


c. пользовательские процессы

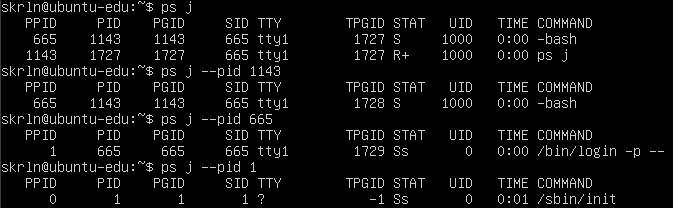


1. При помощи команды **ps** **afx** найдите и зафиксируйте все процессы, порожденные от процесса init:



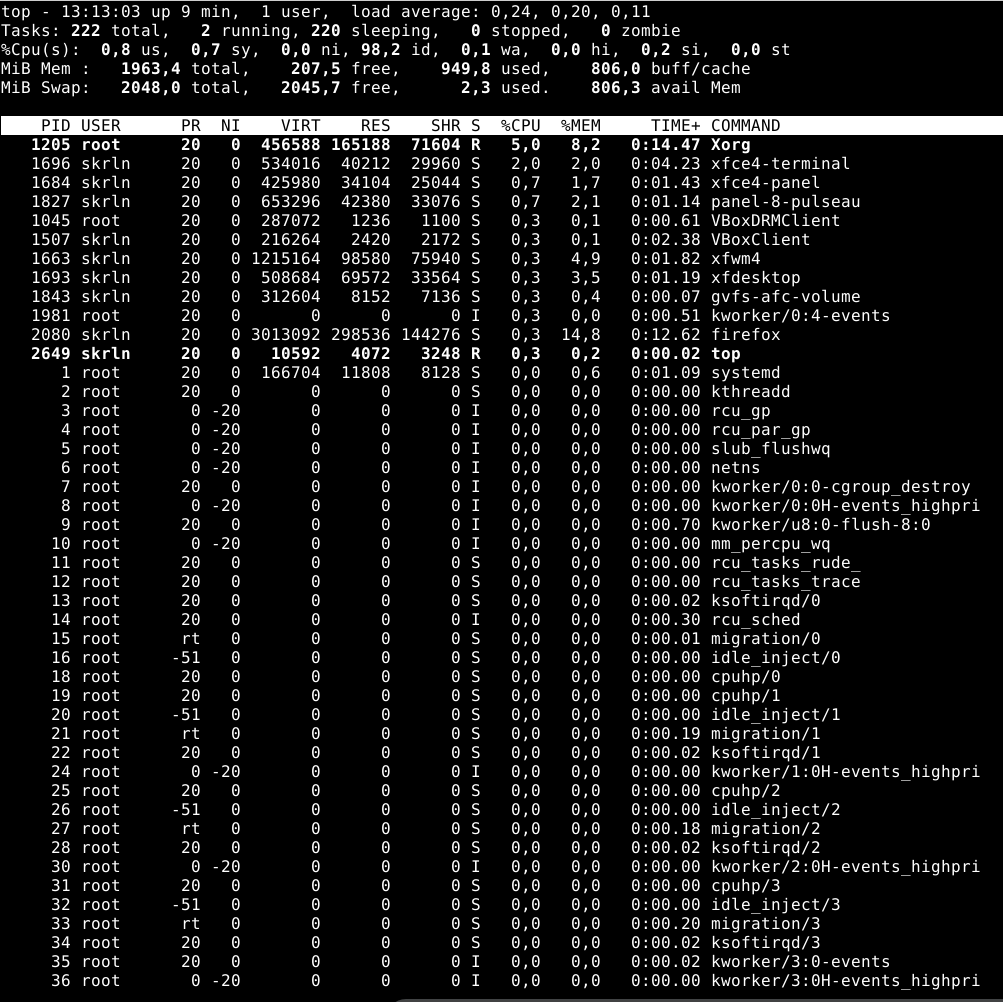


1. Проследите и отметьте всю цепочку порождаемых процессов от процесса init до процесса ps, c выводом которого вы работаете:

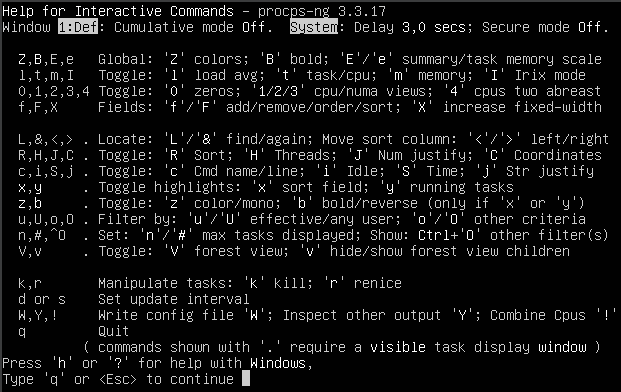


# Упражнение 8.2. Атрибуты процессов

1. Запустите программу **top**. Нажмите клавишу **h** с основными принципами использования этой программы:

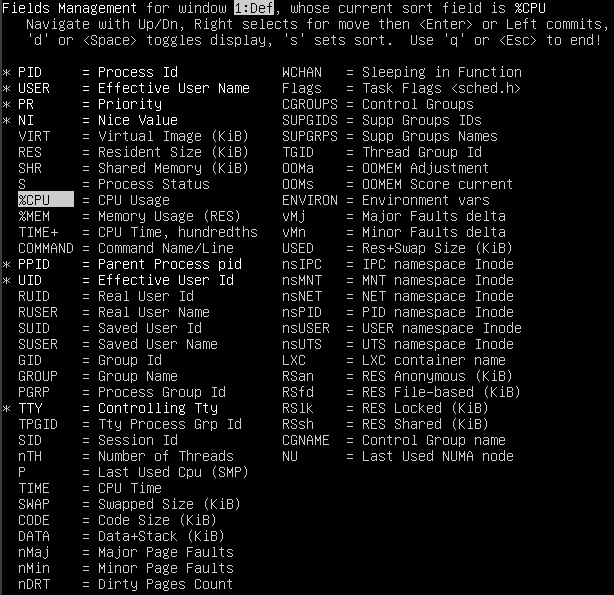


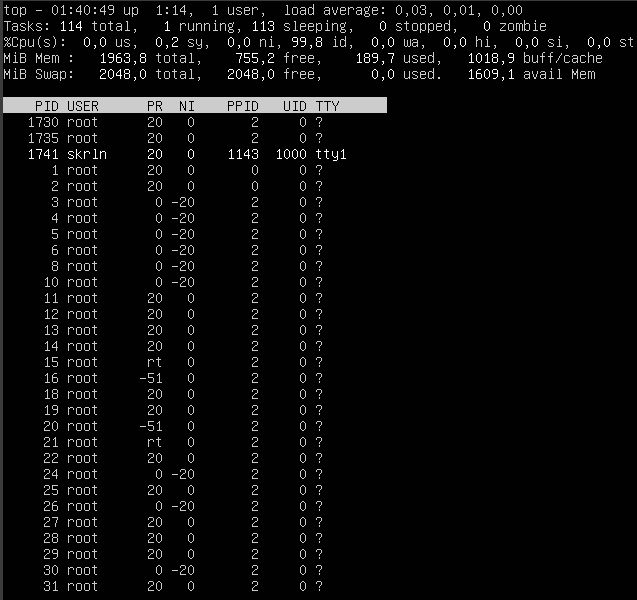
!!!



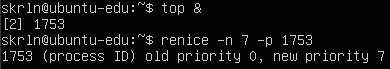
1. Заставьте программу отображать следующие поля: **PID**, **PPID**, **NI**, **UID**, **PRI**, **TTY**, **USER**

f

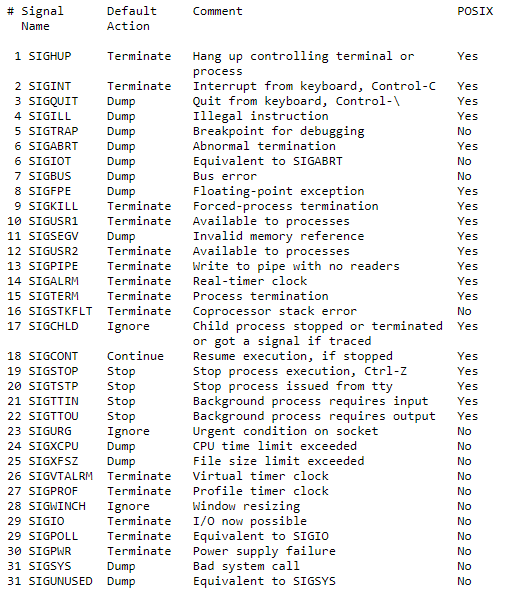




1. Измените атрибут **NICE** процесса top, проследите за изменившемся приоритетом процесса.



# Упражнение 8.3. Задания и сигналы

* 
* 

1. Запустите программу просмотра любой известной страницы руководства **man** на выполнение на переднем фоне.
2. Завершите полученный в предыдущем пункте процесс нажатием комбинации клавиш **^C** (**SIGINT**). Проследите за реакцией системы:

Не завершен

1. Запустите программу просмотра любой известной страницы руководства **man** на выполнение на переднем фоне.
2. Завершите полученный в предыдущем пункте процесс нажатием комбинации клавиш **^\** (**SIGQUIT**). Проследите за реакцией системы:

Не завершен

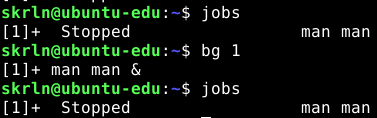
1. Запустите программу просмотра любой известной страницы руководства **man** на выполнение на переднем фоне.
2. Приостановите полученный в предыдущем процесс нажатием комбинации клавиш **^Z** (**SIGTSTP**). Проследите за сообщениями системы:

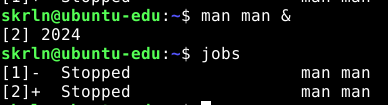


1. При помощи команд **fg**, **bg** поместите приостановленное задание на передний (задний) фон соответственно:

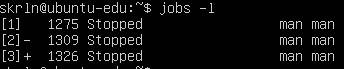


fg/bg %№

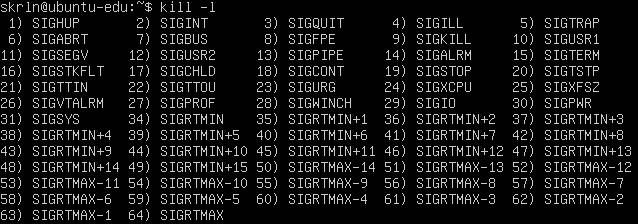




1. Последовательно запустите несколько программ просмотра страниц руководства **man** на выполнение, перемещая получаемы процессы на задний фон. Проследите за присвоенными идентификаторами процессов **PID** и заданий **JOBID**:



1. При помощи команды **jobs** просмотрите и зафиксируйте список выполняющихся заданий:
2. При помощи команды **kill** посылайте различным заданиям сигналы **SIGINT**, **SIGTERM**, **SIGQUIT**, **SIGKILL**, **SIGSTOP**, **SIGCONT**, проследите за результатом:





Лабораторная работа зачтена:

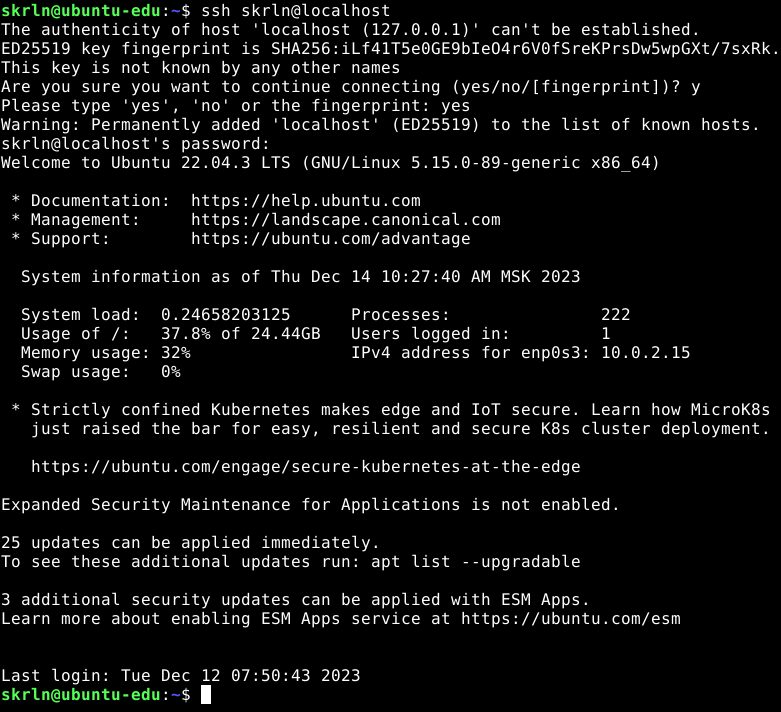
Дата:

Подпись преподавателя:

**Лабораторная работа 9. Сетевое окружение UNIX**

# Упражнение 9.1. Удаленный доступ в систему с помощью протокола SSH

1. При помощи команды **ssh** подключитесь к операционной системе соседа:



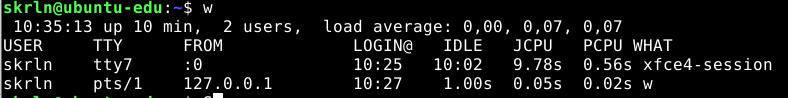
1. При помощи следующих команд: **whoami**, **w**, **who**, **users**, **id**, **date**, **uname**, **tty**, **uptime** получите информацию о **УДАЛЕННОЙ** системе и пользователях и зафиксируйте ее:
   1. имя текущего пользователя:



* 1. его идентификаторы UID, GID и идентификаторы вторичных групп:



* 1. количество и имена пользователей, работающих в системе:



* 1. имя текущего виртуального терминала:



e. системная дата и время:



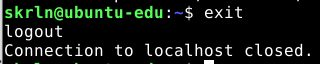
* 1. версия операционной системы компьютера:



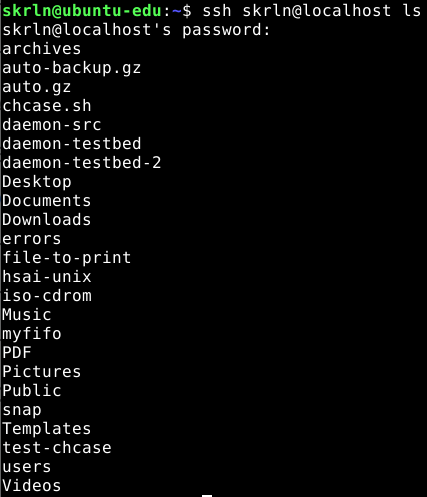
* 1. время работы операционной системы от момента запуска и нагрузку на операционную систему:



1. Закончите сеанс работы в операционной системе на машине соседа.



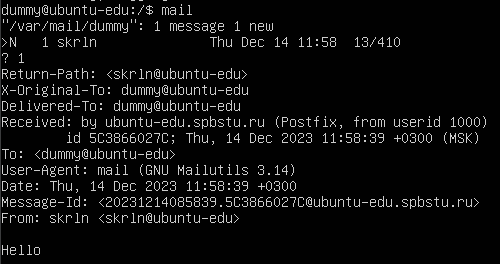
1. При помощи команды **ssh** запустите команду вывода полной информации о файлах домашнего каталога на **УДАЛЕННОЙ** операционной системе машины соседа, проследите за появляющимися сообщениями:



# Упражнение 9.2. Локальная и сетевая почта

1. При помощи команды **mail** пошлите несколько писем различного содержания пользователю **vinnie**, проследите за появляющимися сообщениями:



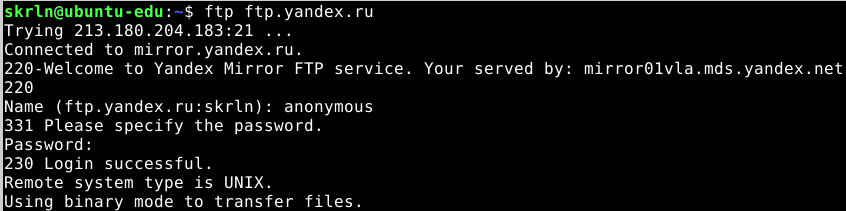


1. Прочитайте и удалите прочитанные письма пользователя **vinnie**, проследите за появляющимися сообщениями:

# 

# Упражнение 9.3. Передача файлов с помощью протокола FTP

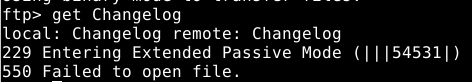
1. При помощи команды **ftp** подключитесь к серверу ftp.yandex.ru (анонимно - учетная запись anonymous), проследите за появляющимися сообщениями:



1. Перейдите в каталог /debian/dists/jessie, проследите за появляющимися сообщениями:



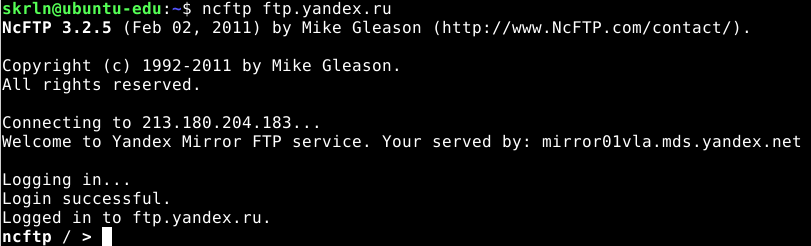
1. Получите с удаленного сервера файл ChangeLog, проследите за появляющимися сообщениями:



1. Закончите сеанс работы с командой **ftp**, проследите за появляющимися сообщениями:

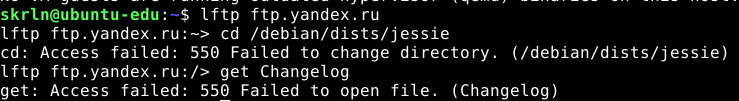


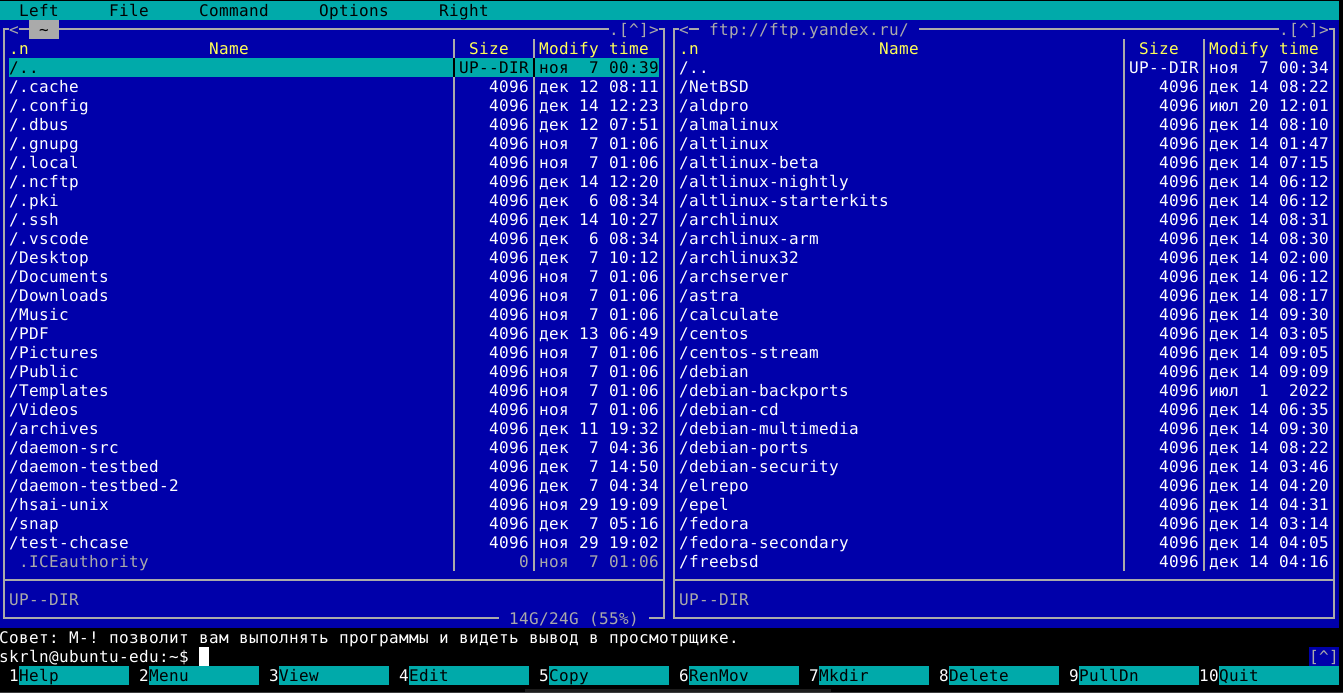
1. При помощи команд **ncftp**, **lftp**, **mc** научитесь пользоваться интерактивными клиентами передачи файлов - повторите, выполненные процедуры из упр. 10.3.





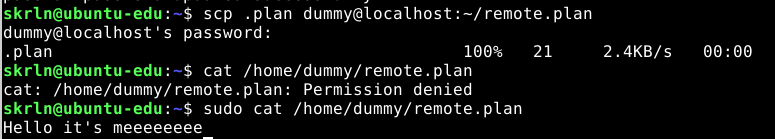






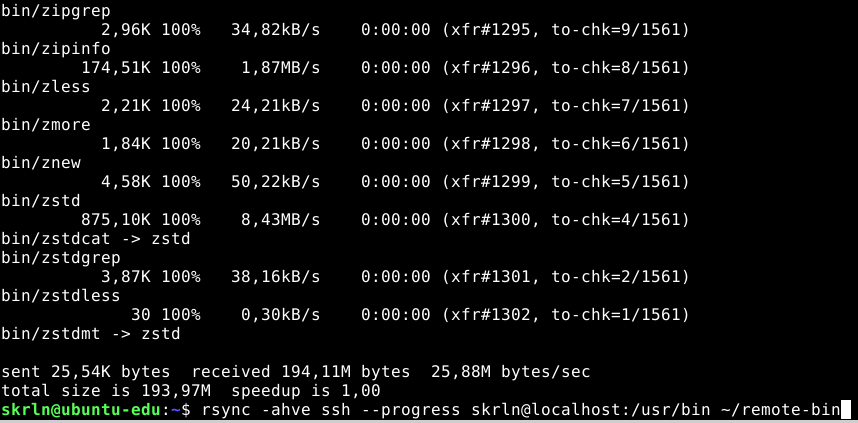
# Упражнение 9.4. Передача файлов с помощью протоколов SCP

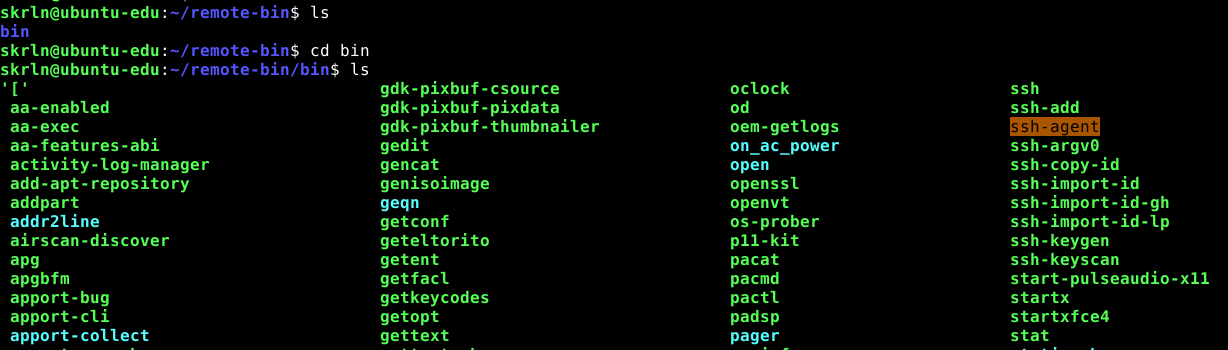
1. При помощи команды **scp** скопируйте файл .plan из домашнего каталога пользователя student машины соседа в домашний каталог пользователя student, с именем remote.plan, проследите за появляющимися сообщениям:
2. Просмотрите содержимое файла remote.plan



# Упражнение 9.5. Синхронизация файлов с помощью протокола RSYNC/SSH

1. При помощи команды **rsync** синхронизуйте содержимое каталога bin из домашнего каталога пользователя student машины соседа, используя защищенный транспорт SSH, в домашний каталог пользователя student, с именем remote.bin, проследите за появляющимися сообщениям:



1. Просмотрите содержимое полученного каталога remote.bin 

Лабораторная работа зачтена:

Дата:

Подпись преподавателя:

**Лабораторная работа 10. Система X Window**

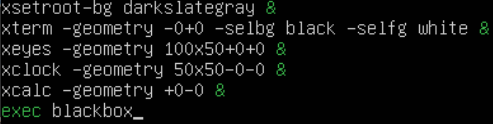
Примечание: при выполнении лабораторной в среде MacOS X или WSL требуется установить дополнительно X server.

# Упражнение 10.1. Создание собственного настольного окружения

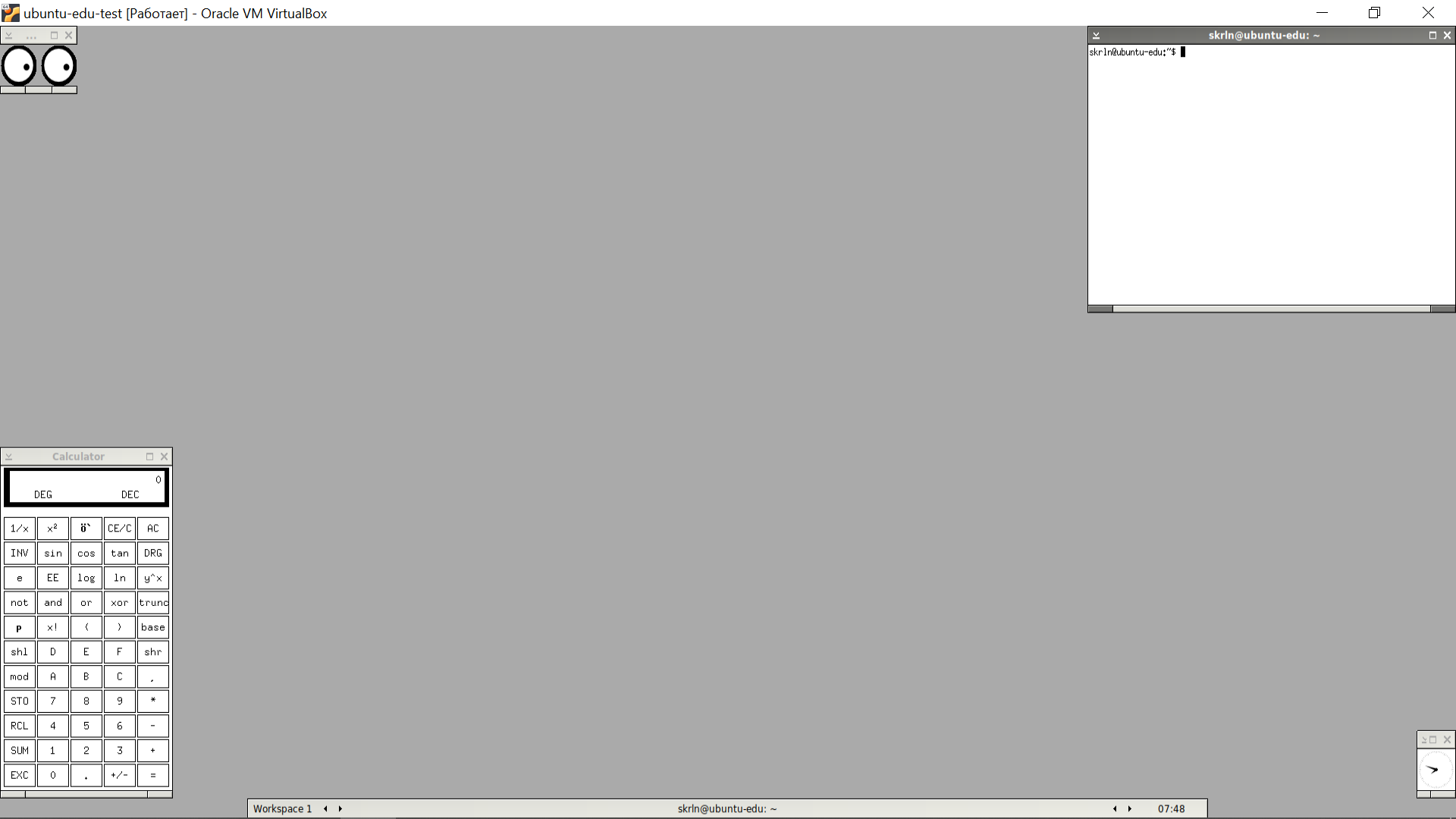
1. Установите оконный менеджер **blackbox**:



1. Отредактируйте сценарий командного интерпретатора .xinitrc в домашней директории пользователя *student*, сформировать собственное окружение, путем задания в данном сценарии команд запуска клиентов:
   1. **xsetroot**, для установки цвета **darkslategray** корневого окна:
   2. **xterm** в правом верхнем углу экрана, цвет символов **white** на фоне **black**:
   3. **xeyes** в левом верхнем углу экрана, размером 100 x 50 пикселов:
   4. **xclock** в правом нижнем углу экрана, размером 50 x 50 пикселов:
   5. **xcalc** в левом нижнем углу экрана:



1. Запустите графическую подсистему X Window, отметьте какое настольное окружение пользователя активизировалось:

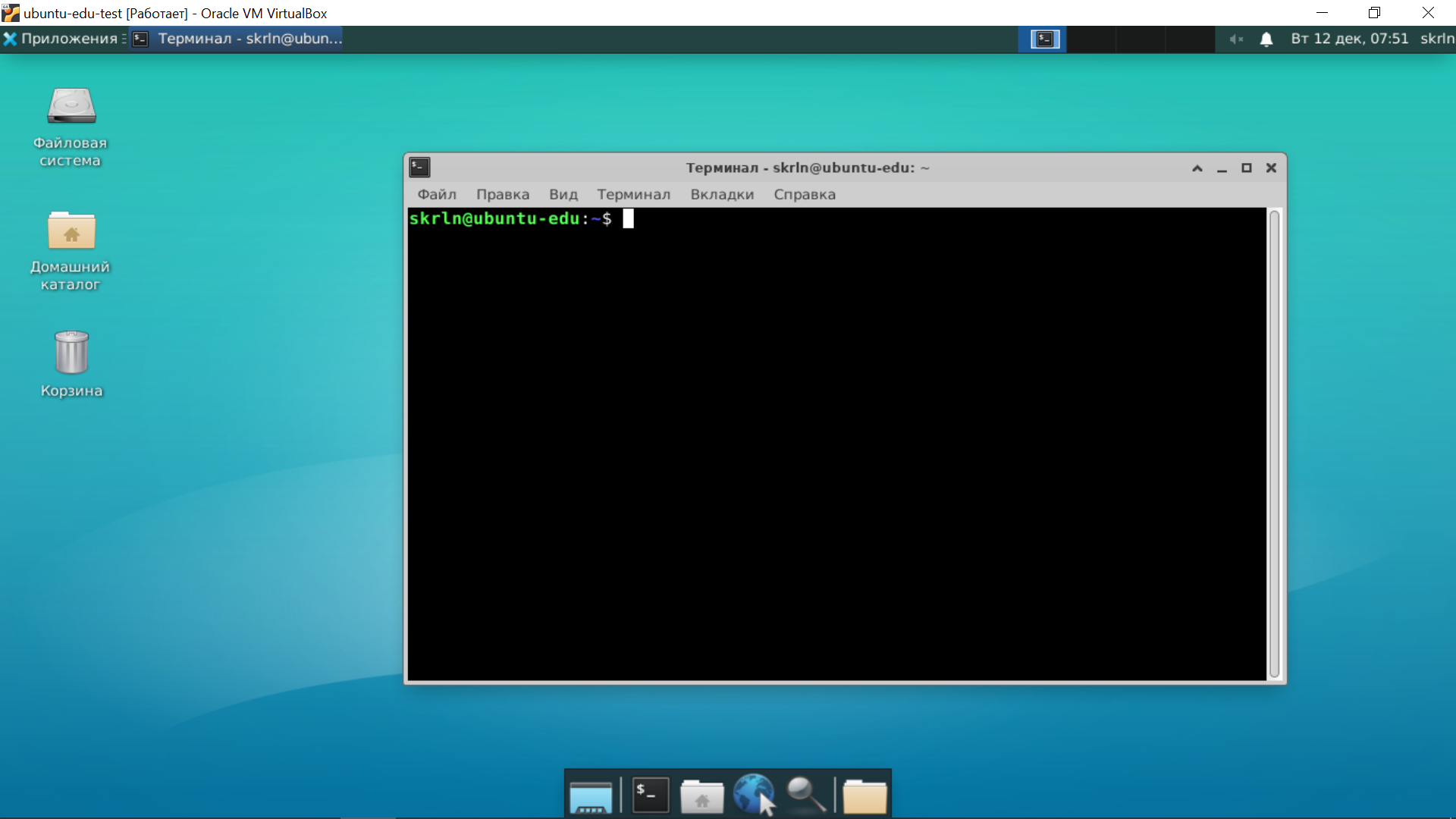


# Упражнение 10.2. Настройка настольного окружения Xfce

1. Создайте сценарий командного интерпретатора .xinitrc в домашней директории пользователя student таким, чтобы при запуске графической системы X Window всегда активизировалось настольное окружение Xfce:



1. Настройте активизировавшееся настольное окружение Xfce по своему вкусу.



1. Закончите графический сеанс работы в операционной системе.

# Упражнение 10.3. Графический вход в систему

Примечание: в случае если в ОС установлен графический вход в систему по умолчанию, то требуется указать какой используется менеджер дисплеев.

1. Настройте автоматический запуск графической среды с использованием менеджера дисплеев:



1. Перезагрузите операционную систему. Убедитесь, что доступен графический вход в систему. Используйте настольное окружение Xfce при входе в систему.



1. Закончите графический сеанс работы в операционной системе.

Лабораторная работа зачтена:

Дата:

Подпись преподавателя:

**Лабораторная работа 11. Подсистема печати**

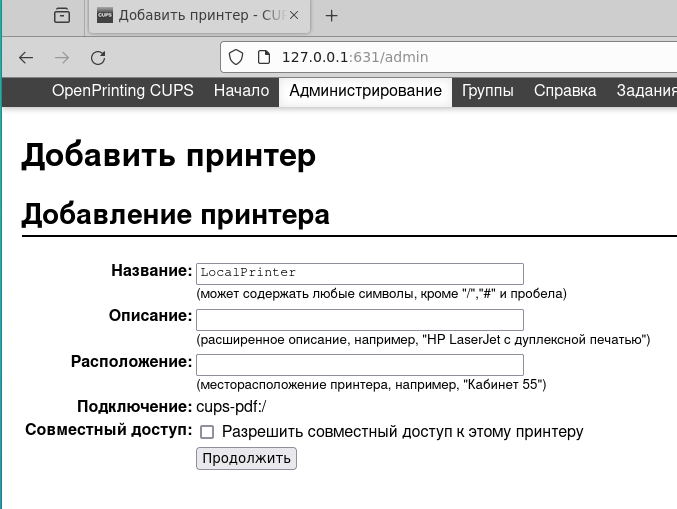
# Упражнение 11.1. Средства печати UNIX

1. Установите систему печати **cups**:



1. Установите виртуальный драйвер для печати в PDF в систему печати **cups**:

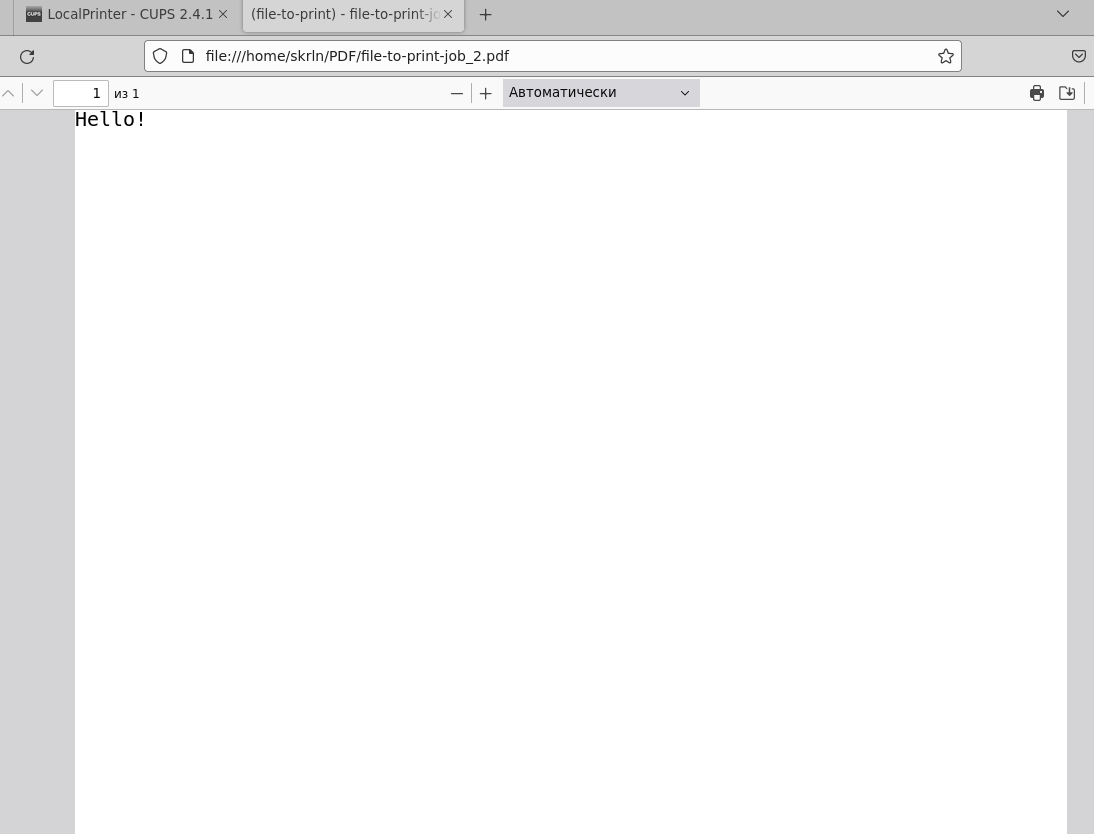


1. Используя браузер в графической среде, зайдите по адресу http://127.0.0.1:631 и добавьте принтер с именем **LocalPrinter** использующий драйвер PDF. 
2. При помощи команд **lpr**, **lpq**, **lprm** (**lp**, **lpstat**, **cancel**):
   1. просмотрите состояния принтера с именем **LocalPrinter**:



* 1. распечатайте любой файл на принтере **LocalPrinter**, проследите за сообщениями:

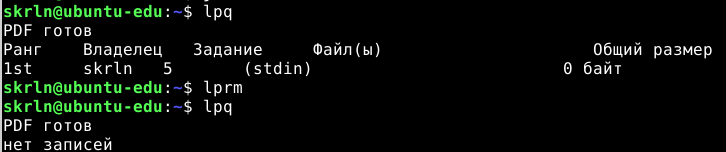




* 1. просмотрите состояния принтера **LocalPrinter**, проследите за сообщениями:

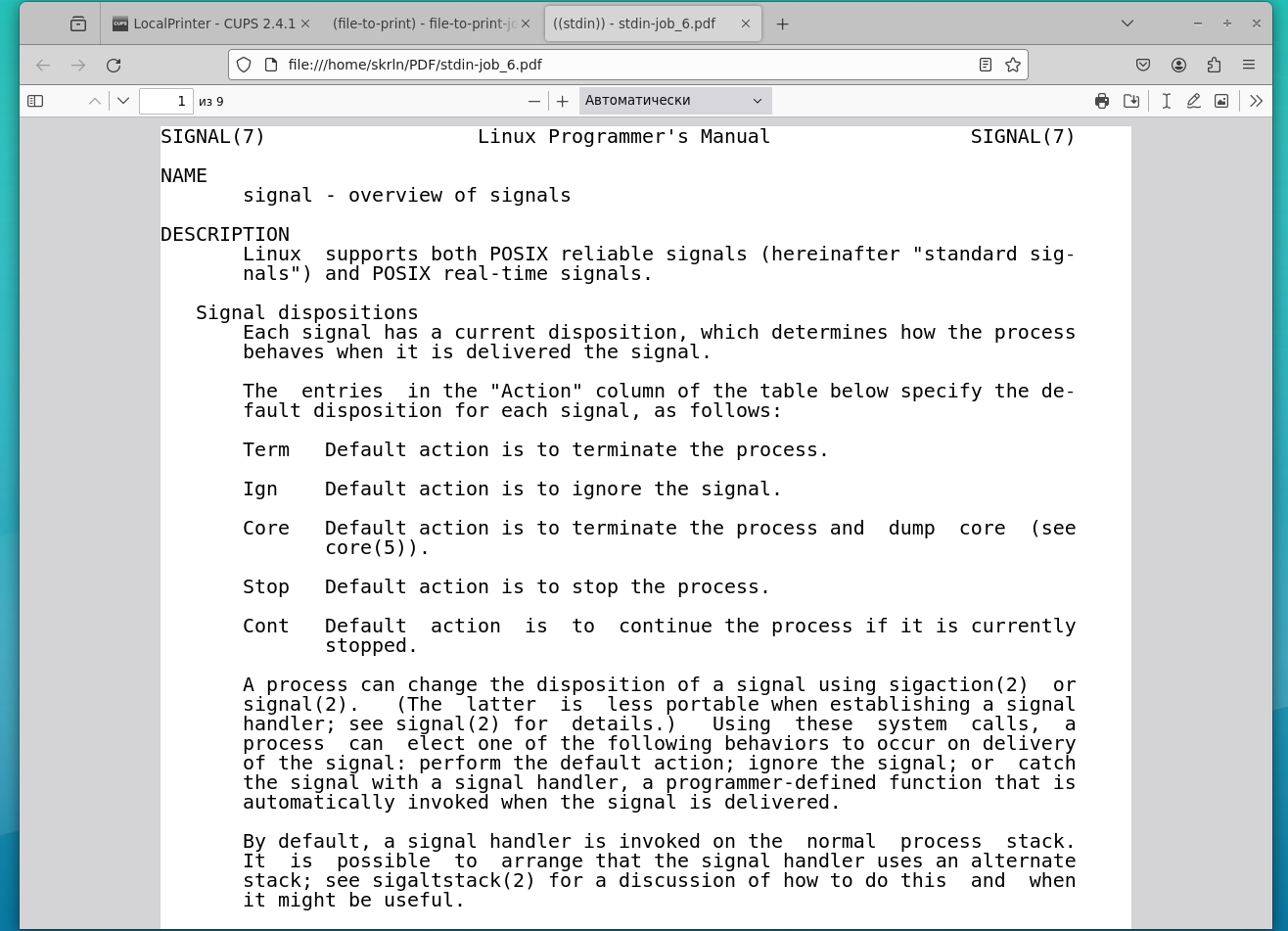


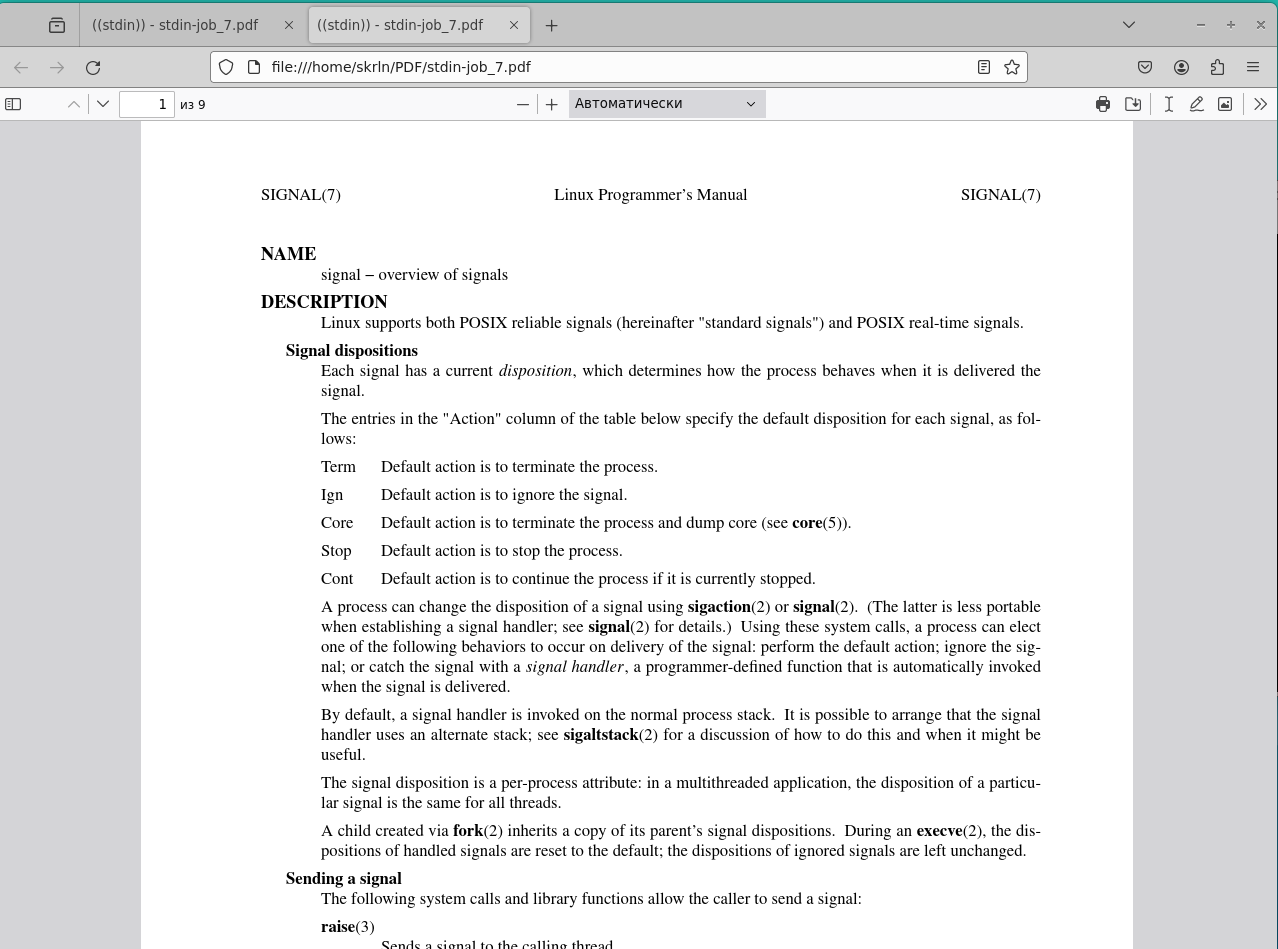
d. удалите задание на печать из очереди принтера **LocalPrinter**, проследите за сообщениями:



e. распечатайте любую известную страницу руководства **man** на принтере **LocalPrinter**, проследите за сообщениями:







!!!

Лабораторная работа зачтена:

Дата:

Подпись преподавателя:

**Основы программирования на языке командного интерпретатора. Индивидуальное задание**

# Создание скелета сценария на языке командного интерпретатора BASH

1. Получите задание на разработку сценария у преподавателя.
2. Создайте файл сценария, назовите его именем, отражающим смысловое назначение. Документируйте назначение вашего сценария в теле самого сценария.
3. Согласно стандарту POSIX.2 придумайте мнемонические модификаторы, которые будут изменять алгоритм работы вашего сценария. Документируйте назначение придуманных модификаторов в теле самого сценария.
4. Создайте скелет сценария выполняющий обработку модификатора -h и выдачу короткой справки по модификаторам и назначению сценария.
5. Дополните скелет сценария, обработкой и проверкой корректности задания остальных модификаторов и/или входных аргументов сценария.

# Реализация алгоритма сценария на языке командного интерпретатора BASH

1. Дополните скелет сценария, алгоритмом, выполняющим смысловое назначение сценария.
2. Оттестируйте полученный сценарий в широком спектре входных параметров (как корректных, так и некорректных)

Лабораторная работа зачтена:

Дата:

Подпись преподавателя: