

Отчет по лабораторной работе №6

Архитектура компьютера: Операционные системы

Гашимова Эсма Эльшан кызы

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Выводы	21
6	Ответы на онтрольные вопросы	22

Список иллюстраций

4.1 Команда pwd	9
4.2 Перемещение между директориями	9
4.3 Просмотр содержимого каталога	10
4.4 Просмотр содержимого каталога	11
4.5 Просмотр содержимого каталога	11
4.6 Просмотр содержимого каталога	12
4.7 Просмотр содержимого каталога	12
4.8 Перемещение между директориями и просмотр содержимого каталога	13
4.9 Создание директории	13
4.10 Создание директории	13
4.11 Создание директорий	14
4.12 Удаление директорий	14
4.13 Попытка удаления директории	14
4.14 Удаление директорий	15
4.15 Опция для утилиты	15
4.16 Опция утилиты	15
4.17 Опции команды	16
4.18 Информация о pwd	16
4.19 Информация о mkdir	17
4.20 Информация о rmdir	18
4.21 Информация о rm	19
4.22 Команда history	20
4.23 Модификация команды	20
4.24 Модификация команды	20

Список таблиц

1 Цель работы

Цель данной лабораторной работы – приобретение практических навыков взаимодействия пользователя с системой посредством командной строки.

2 Задание

1. Определить полное имя домашнего каталога.
2. Выполнить следующие действия:
 - Перейти в каталог `/tmp`.
 - Вывести на экран содержимое каталога `/tmp`.
 - Определить, есть ли в каталоге `/var/spool` подкаталог с именем `cron`.
 - Перейти в домашний каталог и вывести на экран его содержимое. Определить, кто является владельцем файлов и подкаталогов.
3. Выполнить следующие действия:
 - В домашнем каталоге создать новый каталог с именем `newdir`.
 - В каталоге `~/newdir` создать новый каталог с именем `morefun`.
 - В домашнем каталоге создать одной командой три новых каталога с именами `letters`, `memos`, `misk`. Затем удалить эти каталоги одной командой.
 - Попробовать удалить ранее созданный каталог `~/newdir` командой `rm`. Проверить, был ли каталог удалён.
 - Удалить каталог `~/newdir/morefun` из домашнего каталога. Проверить, был ли каталог удалён.
4. С помощью команды `man` определить, какую опцию команды `ls` нужно использовать для просмотра содержимого не только указанного каталога, но и подкаталогов, входящих в него.

5. С помощью команды `man` определить набор опций команды `ls`, позволяющий отсортировать по времени последнего изменения выводимый список содержимого каталога с развёрнутым описанием файлов.
6. Использовать команду `man` для просмотра описания следующих команд: `cd`, `pwd`, `mkdir`, `rmdir`, `rm`. Поясните основные опции этих команд.
7. Используя информацию, полученную при помощи команды `history`, выполнить модификацию и исполнение нескольких команд из буфера команд.

3 Теоретическое введение

В операционной системе типа Linux взаимодействие пользователя с системой обычно осуществляется с помощью командной строки посредством построочного ввода команд. При этом обычно используются командные интерпретаторы языка shell: `/bin/sh`; `/bin/csh`; `/bin/ksh`.

Командой в операционной системе называется записанный по специальным правилам текст (возможно с аргументами), представляющий собой указание на выполнение какой-либо функций (или действий) в операционной системе. Обычно первым словом идёт имя команды, остальной текст — аргументы или опции, конкретизирующие действие.

Общий формат команд можно представить следующим образом: Команда `man`. Команда `man` используется для просмотра (оперативная помощь) в диалоговом режиме руководства (`manual`) по основным командам операционной системы типа Linux.

Формат команды: `man`

Файловая система ОС типа Linux — иерархическая система каталогов, подкаталогов и файлов, которые обычно организованы и сгруппированы по функциональному признаку. Самый верхний каталог в иерархии называется корневым и обозначается символом `/`. Корневой каталог содержит системные файлы и другие каталоги.

В работе с командами, в качестве аргументов которых выступает путь к какому-либо каталогу или файлу, можно использовать сокращённую запись пути.

4 Выполнение лабораторной работы

Полное имя домашнего каталога можно узнать с помощью утилиты `pwd` (рис. fig. 4.1).



Рис. 4.1: Команда `pwd`

С помощью утилиты `cd` перехожу в подкаталог `tmp` корневого каталога (рис. fig. 4.2).



Рис. 4.2: Перемещение между директориями

С помощью утилиты ls, пока что без ключей, просматриваю содержимое каталога tmp (рис. fig. 4.3).

```
ls
8cfa45b78d484cad918-abrtd.service-gL6D8D
8cfa45b78d484cad918-chronyd.service-4F9bmA
8cfa45b78d484cad918-colord.service-LtWpGA
8cfa45b78d484cad918-dbus-broker.service-dZeb44
8cfa45b78d484cad918-low-memory-monitor.service-gk

8cfa45b78d484cad918-ModemManager.service-TNFTZt
8cfa45b78d484cad918-passim.service-qQ4xpf
8cfa45b78d484cad918-polkit.service-nqJYYr
8cfa45b78d484cad918-power-profiles-daemon.service

8cfa45b78d484cad918-rtkit-daemon.service-oXcd5e
8cfa45b78d484cad918-switcheroo-control.service-Bx
8cfa45b78d484cad918-systemd-logind.service-48k23n
```

Рис. 4.3: Просмотр содержимого каталога

Пробую использовать команду ls с разными опциями. Опция -l позволит увидеть дополнительную информацию о файлах в каталоге: время создания, владельца, права (рис. fig. 4.4).

```
ls -l
```

b 22 13:27 systemd-private-97043ffab07a48cfa45b78c
b 22 13:27 systemd-private-97043ffab07a48cfa45b78c
mA
b 22 13:27 systemd-private-97043ffab07a48cfa45b78c
A
b 22 13:27 systemd-private-97043ffab07a48cfa45b78c
dZeb44
b 22 13:27 systemd-private-97043ffab07a48cfa45b78c
ervice-gkUsCs
b 22 13:27 systemd-private-97043ffab07a48cfa45b78c
-TNFTZt
b 22 13:29 systemd-private-97043ffab07a48cfa45b78c
c

Рис. 4.4: Просмотр содержимого каталога

Опция -а покажет скрытые файлы в каталоге (рис. fig. 4.5).

```
ls -la
```

```
8cfa45b78d484cad918-abrtld.service-gL6D8D  
8cfa45b78d484cad918-chronyd.service-4F9bmA  
8cfa45b78d484cad918-colord.service-LtWpGA  
8cfa45b78d484cad918-dbus-broker.service-dZeb44  
8cfa45b78d484cad918-low-memory-monitor.service-gk0  
  
8cfa45b78d484cad918-ModemManager.service-TNFTZt  
8cfa45b78d484cad918-passim.service-qQ4xpf  
8cfa45b78d484cad918-polkit.service-nqJYYr  
8cfa45b78d484cad918-power-profiles-daemon.service-
```

Рис. 4.5: Просмотр содержимого каталога

Перехожу в каталог `/var/spool/` с помощью `cd`. Чтобы определить, есть ли в каталоге подкатлог с соответствующим именем, на самом деле, достаточно начать вводить какую-нибудь команду и имя файла и воспользоваться подсказкой `tab`,

многие окружения рабочего стола обозначают файлы и каталоги разными цветами. Но на всякий случай воспользуемся утилитой `ls` с флагом `-F`, чтобы проверить, что мы найдем именно каталог. И да, в директории действительно есть такой каталог (рис. fig. 4.6). (рис. fig. 4.7).



```
cd /var/spool/  
spool$
```

Рис. 4.6: Просмотр содержимого каталога



```
ls -F  
ail/  plymouth/
```

Рис. 4.7: Просмотр содержимого каталога

Возвращаюсь в домашний каталог, для этого достаточно ввести команду `cd`. Затем проверяю содержимое каталога с помощью утилиты `ls`, опция `-l` позволяет определить владельцев файлов, опция `-a` показывает все содержимое каталога, `-F` поможет определить что из содержимого каталога файл, а что каталог (рис. fig. 4.8).

```

$ ls -laF
eegashimova 1056 мар 22 13:29 ./
root         70 мар  2 23:05 ../
eegashimova 9839 мар 14 21:24 .bash_history
eegashimova  18 фев  9 2024 .bash_logout
eegashimova 144 фев  9 2024 .bash_profile
eegashimova 728 мар  8 00:52 .bashrc
eegashimova 472 мар  8 00:53 .cache/
eegashimova 422 мар  7 23:29 .config/
eegashimova 442 мар  3 17:45 .gitconfig
eegashimova  98 мар  8 01:23 git-extended/
root         530 мар  8 00:51 gitflow/
eegashimova 136 мар 14 22:32 .gnupg/
eegashimova  20 мар  2 23:13 .local/
eegashimova  48 мар  3 17:45 .mozilla/
eegashimova  92 мар  7 23:29 .npm/
eegashimova 225 мар  7 22:41 package.json
eegashimova  50 мар 14 20:38 .password-store/
eegashimova 132 мар 14 21:44 .ssh/
eegashimova 274 мар  3 19:49 study_2024-2025_os-in

```

Рис. 4.8: Перемещение между директориями и просмотр содержимого каталога

Создаю директорию newdir с помощью утилиты mkdir, затем проверяю, что директория создалась с помощью ls (рис. fig. 4.9).

```

$ mkdir newdir
$ ls
study_2024-2025_os-intro  Загрузки  'Рабочий стол'
                           Изображения  Шаблоны
                           Музыка
                           Общедоступные

```

Рис. 4.9: Создание директории

Создаю для каталога newdir подкаталог morefun, проверяю, что каталог собран (рис. fig. 4.10).

```

mkdir newdir/morefun
ls newdir/

```

Рис. 4.10: Создание директории

Чтобы создать несколько директорий одной строчкой нужно перечислить названия директорий через пробел после утилиты `mkdir` (рис. fig. 4.11). Проверяю, что все файлы созданы.

```
mkdir letters memos misk
ls
work      Изображения  Шаблон
Видео     Музыка
Документы Общедоступные
-2025_os-intro Загрузки  'Рабочий стол'
```

Рис. 4.11: Создание директорий

Чтобы удалить несколько **пустых** директорий одной строчкой нужно перечислить названия директорий через пробел после утилиты `rmdir` (рис. fig. 4.12). Проверяю, что все файлы удалены.

```
rmdir letters/ memos/ misk/
ls
-2025_os-intro  Загрузки      'Рабочий стол'
                 Изображения   Шаблоны
                 Музыка
                 Общедоступные
```

Рис. 4.12: Удаление директорий

Пытаюсь удалить `newdir` с помощью `rm`. Утилита `rm` по умолчанию удаляет файлы, чтобы она удалила пустую директорию нужно добавить опцию `-d`, но `newdir` не пустая дериктория, поэтому нужно добавить опцию для рекурсивного удаления `-r`. Использовалась утилиты без опций, поэтому каталог не был удален (рис. fig. 4.13).

```
rm newdir/
newdir/': Это каталог
ls
-2025_os-intro  Загрузки      'Рабочий стол'
                 Изображения   Шаблоны
                 Музыка
```

Рис. 4.13: Попытка удаления директории

Удаляю директорию newdir с помощью утилиты rmdir, т.к директория не пустая, я добавляю флаг удалить рекурсивно -r, чтобы удалился и все подкаталоги (рис. fig. 4.14).

```
rmdir -p newdir/morefun/  
ls  
work      Изображения  Шаблоны  
Видео     Музыка  
Документы Общедоступные  
Загрузки  'Рабочий стол'
```

Рис. 4.14: Удаление директорий

С помощью команды man ls я могу прочесть документацию к команде ls, опция, которая позолить выводить все подкаталоги каталогов предоставлена на скриншоте, это -R (рис. fig. 4.15).

```
-R, --recursive  
показывать каталоги рекурсивно
```

Рис. 4.15: Опция для утилиты

Так как мне нужно найти опцию утилиты ls для сортировки, то логично сузить поиск до результатов с таким же вопросом (рис. fig. 4.16). Выяснила, что для сортировки и вывода информации нужна комбинация опций -lt.

```
man ls | grep "sort"
```

Рис. 4.16: Опция утилиты

С помощью man cd узнаю описание команды cd и ее опции. Основных опций немного (рис. fig. 4.17). 1. -P - позволяет следовать по символическим ссылкам перед тем, как обработаны все переходы ‘..’ 3. -L - переходит по символическим ссылкам только после того, как обработаны все переходы “..” 4. -e - позволяет выйти с ошибкой, если диреткория, в которую нужно перейти, не найдена.

```

$ man cd
$ man bash | grep 'cd'
31: warning: cannot select font 'CW'
$ working directory as set by the cd c
working directory as set by the cd co
path for the cd command. This is a
specified by the cd command. A
in command. The value of this variab
4: warning: cannot select font 'CW'
4: warning: cannot select font 'CW'
s, where [a-d] is equivalent to [abcd
t working directory as set by cd, pus

```

Рис. 4.17: Опции команды

С помощью `man pwd` узнаю описание команды `pwd` и ее опции (рис. fig. 4.18).

1. `-L` - брать директорию из переменной окружения, даже если она содержит символические ссылки.
2. `-P` - отбрасывать все символические ссылки.

```

PWD(1)                                User Commands
NAME
    pwd - print name of current/working directory
SYNOPSIS
    pwd [OPTION]...
DESCRIPTION
    Print the full filename of the current working directory.

    -L, --logical
        use PWD from environment, even if it contains symlinks

    -P, --physical
        avoid all symlinks

    --help display this help and exit

    --version
        output version information and exit

    If no option is specified, -P is assumed.

```

Рис. 4.18: Информация о `pwd`

С помощью `man mkdir` узнаю описание команды `mkdir` и ее опции (рис. fig. 4.19).

1. `-m` - устанавливает права доступа создаваемой директории как `chmod`, синтаксис тоже как у `chmod`.
2. `-p` - позволяет рекурсивно создавать директории и их подкаталоги
3. `-v` - выводит сообщение о созданных директориях
4. `-z` - установить контекст SELinux для создаваемой директории по умолчанию
5. `-context` - установить контекст SELinux для создаваемой директории в значении CTX

```

MKDIR(1)                                User Commands
NAME
    mkdir - make directories

SYNOPSIS
    mkdir [OPTION]... DIRECTORY...

DESCRIPTION
    Create the DIRECTORY(ies), if they do not already exist.

    Mandatory arguments to long options are mandatory for short
    too.

    -m, --mode=MODE
        set file mode (as in chmod), not a=rwx - umask

    -p, --parents
        no error if existing, make parent directories as needed
        their file modes unaffected by any -m option.

    -v, --verbose
        print a message for each created directory
```

Рис. 4.19: Информация о `mkdir`

С помощью `man rmdir` узнаю описание команды `rmdir` и ее опции (рис. fig. 4.20).

1. `-ignore-fail-on-non-empty` - отменяет вывод ошибки, если каталог не пустой, просто его игнорирует
2. `-p` - удаляет рекурсивно каталоги, если они все содержат в себе только удаляемый каталог
3. `-v` - выводит сообщение о каждом удалении директории.

```
RMDIR(1)                                User Commands

NAME
    rmdir - remove empty directories

SYNOPSIS
    rmdir [OPTION]... DIRECTORY...

DESCRIPTION
    Remove the DIRECTORY(ies), if they are empty.

    --ignore-fail-on-non-empty
        ignore each failure to remove a non-empty directory

    -p, --parents
        remove DIRECTORY and its ancestors; e.g., 'rmdir -p a/b'
        is similar to 'rmdir a/b a'

    -v, --verbose
        output a diagnostic for every directory processed

    --help
        display this help and exit
```

Рис. 4.20: Информация о rmdir

С помощью `man rm` узнаю описание команды `rm` и ее опции (рис. fig. 4.21). 1. `-f` - игнорировать несуществующие файлы или аргументы, никогда не выводить запрос на подтверждение удаления 2. `-i` - выводить запрос на подтверждение удаления каждого файла 3. `-I` - вывести запрос на подтверждение удаления один раз, для всех файлов, если удаляется больше 3-х файлов или идет рекурсивное удаление 4. `-interactive` - заменяет предыдущие три опции, можно выбрать одну из них. 5. `-one-file-system` - во время рекурсивного удаления пропускать директорию из других файловых систем 6. `-no-preserve-root` если в качестве директории задана корневая, то считать что это обычная директория и начать удаление. 7. `-r`, `-R` - удаляет директории их содержимое рекурсивно 8. `-d`, `-dir` - удаляет пустые директории 9. `-v` - прописывает все действия команды

```
OPTIONS
  Remove (unlink) the FILE(s).

  -f, --force          ignore nonexistent files and arguments, never prompt

  -i                  prompt before every removal

  -I                  prompt once before removing more than three files, or w
moving recursively; less intrusive than -i, while still providing
protection against most mistakes

  --interactive[=WHEN] prompt according to WHEN: never, once (-I), or always
without WHEN, prompt always

  --one-file-system    when removing a hierarchy recursively, skip any direct
is on a file system different from that of the corresponding
command line argument

  --no-preserve-root
```

Рис. 4.21: Информация о rm

Опции `-help` `-version` применимы почти ко всем утилитам, они показывают справку по команде и ее версию соответственно.

Вывела историю команд с помощью утилиты `history` (рис. 4.22).

```
$ history

linux version"

linux version"

linux version"

linux version"
detected Mhz processor"
PU0"
emory available"
ypervisor detected"

ount"
git
ch
```

Рис. 4.22: Команда history

Модифицировала команду (рис. fig. 4.23).

```
!364:s/newdir/fff
```

Рис. 4.23: Модификация команды

Модифицировала команду (рис. fig. 4.24).

```
!384:s/c/la
```

Рис. 4.24: Модификация команды

5 Выводы

Я приобрела практические навыки взаимодействия пользователя с системой посредством командной строки.

6 Ответы на онтрольные вопросы

1. Командная строка - это текстовая система, которая передает команды компьютеру и возвращает результаты пользователю. В операционной системе типа Linux взаимодействие пользователя с системой обычно осуществляется с помощью командной строки посредством построчного ввода команд.
2. Для определения абсолютного пути к текущему каталогу используется команда `pwd`. Например: если я введу `pwd` в своем домашнем каталоге то получу `/home/evdvorkina`
3. С помощью команды `ls` можно определить имена файлов, при помощи опции `-F` уже мы сможем определить тип файлов, если нам необходимы скрытые файлы, добавим опцию `-a`. Пример есть в лабораторной работе.
4. С помощью команды `ls` можно определить имена файлов, если нам необходимы скрытые файлы, добавим опцию `-a`. Пример есть в лабораторной работе.
5. `rmdir` по умолчанию удаляет пустые каталоги, не удаляет файлы. `rm` удаляет файлы, без дополнительных опций (`-d`, `-r`) не будет удалять каталоги. Удалить в одной строчке одной командой можно файл и каталог. Если файл находится в каталоге, используем рекурсивное удаление, если файл и каталог не связаны подобным образом, то добавим опцию `-d`, введя имена через пробел после утилиты.
6. Вывести информацию о последних выполненных пользователем команд можно с помощью `history`. Пример приведет в лабораторной работе.
7. Используем синтаксис `!номеркоманды` в выводе `history:s/что заменяем/на`

что заменяем Примеры приведены в лабораторной работе.

8. Предположим, я нахожусь не в домашнем каталоге. Если я введу “cd ; ls”, то окажусь в домашнем каталоге и получу вывод файлов внутри него.
9. Символ экранирования - (обратный слеш) добавление перед спецсимволом обратный слеш, чтобы использовать специальный символ как обычный. Также позволяет читать системе название директорий с пробелом. Пример:
cd work/Операционные системы/
10. Опция -l позволит увидеть дополнительную информацию о файлах в каталоге: время создания, владельца, права доступа
11. Относительный путь к файлу начинается из той директории, где вы находитесь (она сама не прописывается в пути), он прописывается относительно данной директории. Абсолютный путь начинается с корневого каталога.
12. Использовать man или -help
13. Клавиша Tab.