

ИТМО
Факультет ПИиКТ
Системное программное обеспечение

Лабораторная работа №1

Нестеров Дали Константинович
Группа Р3202

Список команд, подлежащих повторению:

Санкт-Петербург
2019 год

ls, pwd, cd, rm, mv, cp, mkdir, rmdir, type, file, find, chmod, ln, wc, tee, cat, tail, head, more, pg, touch

ls [files]

-Семантика: **list**

-Принимает как аргумент файлы и каталоги (по умолчанию ``.``) и выводит эти файлы, а также все файлы в каталогах вместе с дополнительной информацией

-Основные ключи:

- l длинный формат записи
- a не игнорировать файлы, начинающиеся с ``.``
- A как -a, но не выводит файлы ``.`` и ``..``
- R рекурсивно выводить содержимое подкаталогов
- r сортировать в обратном порядке
- h добавлять к каждому размеру файла букву размера (например, М для мегабайт)

pwd

-Семантика: **print working directory**

-Выводит полный путь от корневого каталога к текущему рабочему каталогу:

cd [directory]

-Семантика: **change directory**

-Меняет текущий рабочий каталог

rm [files]

-Семантика: **remove**

-Удаляет файлы

-Основные ключи:

- r рекурсивно удалять содержимое подкаталогов
- i запрашивать подтверждение при удалении
- f не выводить сообщения об ошибках и не запрашивать подтверждения

mv [file1] [file2]

-Семантика: **move**

-Перемещает файл file1 в file2

-Основные ключи:

- i запрашивать подтверждение при перемещении
- f не выводить сообщения об ошибках и не запрашивать подтверждения
- завершает ввод ключей. Нужно для файлов, имена которых начинаются с ``-``

cp [file1] [file2]

-Семантика: **copy**

-Копирует файл file1 в file2

-Основные ключи:

- r, R рекурсивно копировать подкаталоги

- f разрешает удаление файла, в который производится копирование, если он не может быть открыт для записи
- i запрашивать подтверждение
- n не перезаписывать существующий файл
- p копирует у исходного файла права, идентификатор группы и даты последнего изменения и последнего доступа

mkdir [files]

- Семантика: **make a directory**
- Создает каталоги с указанными именами. (Права по умолчанию - 777)
- Основные ключи:
 - p создает родительские каталоги, если не существуют
 - m переопределяет права доступа по умолчанию

Rmdir [files]

- Семантика: **remove a directory**
- Удаляет пустые директории
- Основные ключи:
 - p удалить родительские каталоги, если они пустые

type [names]

- Описывает как будут восприняты аргументы, если их ввести в качестве команды.

file [files]

- Определяет тип файла
- Основные ключи:
 - f file считывает из файла список файлов для проверки
 - L определяет тип файлов указанных по ссылке

find [path]

- Ищет файлы по свойствам (по умолчанию выводит все файлы в рабочем каталоге)
- Основные ключи:
 - name искать по имени
 - type по типу файла (например f = файл, а d = директория)
 - user по пользователю
 - group по группе
 - perm по правам

chmod [files]

- Семантика: **change mode**
- Меняет права доступа к файлам
- Основные ключи:
 - R рекурсивно меняет права доступа у содержимого подкаталогов
 - f не выводить сообщения об ошибках
 - v подробно описывать каждое изменение

ln [file] [link_name]

-Семантика: **link**

-Создает жесткую ссылку на файл

-Основные ключи:

-s создает символическую ссылку

wc [files]

-Семантика: **word count**

-Является фильтром

-Считывает число переводов строк, слов и байт для каждого заданного файла, также выводит итоговую строку, если их было задано несколько. Если файлы не заданы, то считает слова из стандартного ввода.

-Основные ключи:

-l вывести число строк

-w число слов

-c число байт

-m число символов

-Пример использования:

```
$ echo "The quick brown fox..." | wc
      1      4     23
```

tee [file]

-Семантика: название произошло от Т-разветвителя в водопроводах

-Является фильтром

-Выводит содержимое потока ввода на вывод и сохраняет это в файл.

-Основные ключи:

-a дописать данные в конец файла, не переписывая его содержимого

-i игнорировать ошибки

-Пример использования:

```
$ echo "...jumps over the lazy dog" | wc | tee temp.txt | wc -w
3
$ cat temp.txt
      1      5     27
```

cat [files]

-Семантика: **concatenate**

-Является фильтром

-Последовательно выводит указанные файлы, объединяя их в единый поток

-Основные ключи:

-n перед каждой строкой вывода писать ее номер

-v выводить непечатные символы (кроме табуляций и переводов строки)

Пример использования:

```
$ printf "a\nb\nc\n" | cat -n
      1  a
      2  b
      3  c
```

tail [files]

- Является фильтром
- Выводит 10 последних строк файла
- Основные ключи:
 - n позволяет задать число выводимых последних строк
 - f позволяет следить за файлом и в реальном времени выводить строки добавленные в конец файла другими процессами
- Пример использования:

```
$ printf "1\n2\n3\n4\n" | tail -n 3
```

```
2
```

```
3
```

```
4
```

head [files]

- Является фильтром
- Выводит 10 первых строк файла
- Основные ключи:
 - n позволяет задать число выводимых первых строк
- Пример использования:

```
$ printf "1\n2\n3\n4\n" | head -n 3
```

```
1
```

```
2
```

```
3
```

more [file]

- Является фильтром
- Позволяет просматривать содержимое файла с остановкой на границе экрана
- Основные ключи:
 - с очищает экран перед вызовом
 - s сжимает несколько пустых строк в одну
- Пример использования:

```
$ ls -al | more -c
```

pg [file]

- Семантика: **page**
- Является фильтром
- Выводит содержимое файла постранично. Признана устаревшей.
- Основные ключи:
 - число задает размер страницы
- Пример использования:

```
$ ls | pg
```

touch [files]

- Перезаписать дату доступа и последнего изменения файла на текущую, также создает пустой файл, если его нет
- Основные ключи:
 - с не создавать файл, если его не существует
 - а изменить только время доступа

-m изменить только время последнего изменения

Список команд, подлежащих изучению:

su, chown, chgrp, less, split, join, paste, cut, tr, cmp, diff, patch, sort, uniq, echo, alias, ulimit, umask, groups, id, getent, xargs

su [-] [username]

-Семантика: **super user** или **switch user**

-Позволяет пользователю выполнить команду от имени другого пользователя не завершая сеанс. Если установлен аргумент `'-'`, то среда устанавливается, как при у заданного пользователя, а иначе передается текущая среда.

-Основные ключи:

-c указать команду для выполнения

-Пример использования:

```
$ su someUser -c "touch file"
```

chown user[:group] [files]

-Семантика: **change owner**

-Меняет владельца (и группу если задана) указанных файлов.

-Основные ключи:

-R рекурсивно изменять владельца содержимого подкаталогов

-f не выводить сообщения об ошибке

-h работать с самими символическими ссылками, а не с файлами, на которые они указывают

-Пример использования:

```
$ ll test1
```

```
-rw-rw-rw- 1 dali dali 25 Mar  4 14:20 test1
```

```
$ chown NOTdali test1
```

```
$ ll test1
```

```
-rw-rw-rw- 1 NOTdali dali 25 Mar  4 14:20 test1
```

chgrp [group] [files]

-Семантика: **change group**

-Меняет группу для указанных файлов

-Основные ключи:

-R рекурсивно изменять группу содержимого подкаталогов

-f не выводить сообщения об ошибке

-h работать с самими символическими ссылками, а не с файлами, на которые они указывают

-Пример использования:

```
$ ll test1
```

```
-rw-rw-rw- 1 dali dali 25 Mar  4 14:20 test1
```

```
$ chgrp adm test1
```

```
$ ll test1
```

```
-rw-rw-rw- 1 dali adm 25 Mar  4 14:20 test1
```

less [file]

-Является фильтром

-Отображает файл с возможностью прокрутки. В отличие от more, может прокручивать обратно. Не нуждается в полном чтении всего файла перед стартом из-за чего быстрее работает с большими файлами. Навигация: space=следующая страница, b=предыдущая страница, enter=следующая строка, k=предыдущая строка, g=первая строка, G=последняя строка и др.

-Основные ключи:

-N выводит также номера строк

-x при выходе оставлять содержимое файлов на экране

-m показывает позицию в файле

-Пример использования:

```
ls -la | less
```

split [file] [prefix]

-Разделяет файл на несколько новых, названия которых состоят из указанного префикса и двух дополнительных букв (aa, ab, ac и т.д.). По умолчанию файл делится на части по 1000 строк, а префикс = x

-Основные ключи:

-l делить на указанное число строк

-d использовать числовые, а не алфавитные суффиксы

-a использовать суффиксы указанной длины (по умолчанию 2)

-b делить на указанное число байт

-Пример использования:

```
$ printf "12\n23\n34\n45\n56" | split -l 3
```

```
$ cat xaa
```

```
12
```

```
23
```

```
34
```

```
$ cat xab
```

```
45
```

```
56
```

join [file1] [file2]

-Является фильтром

-Находит соответствия между двумя файлами и выводит получившийся текст в поток вывода. Ищет пары строк, обладающие совпадающими первыми полями, и выводит строку, состоящую из этого поля и содержимого обеих строк.

-Основные ключи:

-a filename выводить так же несоединенные строки из файла под номером filename (1 или 2)

-Пример использования:

```
$ printf "common blue\nunique orange" > test1
```

```
$ printf "unique2 dog\ncommon cat" | sort | join test1 -
common blue cat
```

paste [files]

-Является фильтром

-Соединяет файлы по колонкам.

-Основные ключи:

-s получившийся текст соединяется горизонтально, а не вертикально (первый столбец окажется на первой строке, второй столбец на второй строке и т.д.)

-d меняет разделитель (по умолчанию \t). Можно указать несколько разделителей и тогда они все будут использоваться по очереди.

-Пример использования:

```
$ printf "Dali\nAlexey\nNikita\n" > names
```

```
$ printf "Nesterov\nKokov\nRogalenko" | paste names - -s -d " "
```

```
Dali Alexey Nikita
```

```
Nesterov Kokov Rogalenko
```

cut [files]

-Является фильтром

-Извлекает секции из файла

-Основные ключи:

-с выбирает только символы в заданных позициях (например, -7, 2-5, 9-)

-f выбирает только определенные поля, разделенные разделителем (см. -d). Синтаксис как у -с

-d переопределяет разделитель (по умолчанию \t)

-Пример использования:

```
$ echo "123:456:789:0" | cut -d : -f 3-
```

```
789:0
```

tr [str1] [str2]

-Семантика: **t**ranslate

-Является фильтром

-Копирует входной поток в выходной заменяя символы из первой строки соответствующими символами второй строки

-Основные ключи:

-d вместо преобразования удаляет все символы, указанные в строке 1

-Пример использования:

```
$ echo "Hello world" | tr -d "ol"
```

```
He wrd
```

cmp [file1] [file2]

-Семантика: **c**ompare

-Является фильтром

-Сравнивает два файла и выводит первый байт и строку, на котором произошло различие

-Основные ключи:

-с печатает различающиеся символы

-Пример использования:

```
$ echo "11121" > test1
```



```
$ echo "11111" | cmp - test1
- test1 differ: byte 4, line 1
```

diff [file1] [file2]

-Семантика: **difference**

-Является фильтром

-Построчно сравнивает два файла. Выводит отличающиеся строки вместе с их номерами и буквой, поясняющей была ли эта строчка добавлена, удалена или изменена в file2.

-Основные ключи:

-i не учитывать регистр

-w игнорировать пробелы

-u «унифицированный формат»: выводит изменения более кратко, новые и старые строки пишутся рядом.

-Пример использования:

```
$ cat old
aaa
bbb
ccc
ddd
~$ printf "aaa\nbbb\nooo\n" | diff old - -u
--- old 2019-03-04 14:07:29.396080000 +0300
+++ -    2019-03-04 14:10:11.144871100 +0300
@@ -1,4 +1,3 @@
   aaa
   bbb
-ccc
-ddd
+ooo
```

patch < [diff_file]

-Обновить файл в соответствии с файлом заплаткой, который был получен применением команды diff

-Основные ключи:

-R откатить изменения

-b старый файл не удаляется, а переименовывается

-Пример использования:

```
$ cat old
aaa
bbb
ccc
ddd
$ cat new
abb
bbb
ddd
$ diff old new -u > patcher
$ patch < patcher
```

```
patching file old
$ cat old
abb
bbb
ddd
$ patch < patcher -R
patching file old
$ cat old
aaa
bbb
ccc
ddd
```

sort [files]

- Является фильтром
- Сортирует входные строки.
- Основные ключи:
 - d учитывает только буквы и цифры
 - b игнорировать пробелы перед строками
 - f игнорировать регистр
 - r в обратном порядке
 - k сортировать столбец по номеру
- Пример использования:

```
$ printf "b\nt\nz\ny\no\nw\nl\n" | sort -r
z
y
w
t
o
l
b
```

uniq [input_file [output_file]]

- Семантика: **unique**
- Является фильтром
- Выводит файл, сжимая идентичные строчки, расположенные рядом, в одну
- Основные ключи:
 - u вывести только строки, которые не повторялись в файле
 - d вывести только строки, которые повторялись
 - c перед каждой строкой вывести число ее повторений с пробелом
 - s определяет число символов в начале строки, которые игнорируются при сравнении
- Пример использования:

```
$ cat test
qwerty
qwerty
asdf
zxcv
```

```
$ cat test | uniq test -d
qwerty
```

echo [text]

-Выводит текст на экран

-Основные ключи:

-е разрешает использование специальных символов

-Пример использования:

```
$ echo -e "hello\nworld"
hello
world
```

alias [name[=value]]

-Создает псевдоним для команды, который можно использовать в качестве более короткого варианта записи

-Пример использования:

```
$ alias la='ls -A'
```

ulimit

-Семантика: **user's limit**

-Наложить или получить ограничения на системные ресурсы для пользователя

-Основные ключи:

-а отобразить текущие ограничения

Ключи для установки отдельных ограничений см. в «Пример использования»

-Пример использования:

```
$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority     (-e) 0
file size                (blocks, -f) unlimited
pending signals          (-i) 7823
max locked memory       (kbytes, -l) 64
max memory size          (kbytes, -m) unlimited
open files               (-n) 1024
pipe size                (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority       (-r) 0
stack size               (kbytes, -s) 8192
cpu time                 (seconds, -t) unlimited
max user processes       (-u) 7823
virtual memory           (kbytes, -v) unlimited
file locks               (-x) unlimited
```

umask [mask]

-Семантика: **user's mask**

-Устанавливает маску, которая будет определять какие права будут убраны у новых файлов.

-Основные ключи:

-S выводит маску в символьной форме

-Пример использования:

```
$ umask -S
```

```
u=rwx,g=rwx,o=rwx
```

```
$ touch test1
```

```
$ ll test1
```

```
-rw-rw-rw- 1 sundalik sundalik 0 Mar  4 13:01 test1
```

```
$ umask 066
```

```
$ touch test2
```

```
$ ll test2
```

```
-rw----- 1 sundalik sundalik 0 Mar  4 13:01 test2
```

groups [username]

-Выводит все группы, в которые входит текущий или заданный юзер

-Пример использования:

```
$ groups
```

```
dali adm dialout cdrom floppy sudo audio dip video plugdev lxd netdev
```

id [username]

-Семантика: **identity**

-Выводит идентификаторы для текущего или заданного пользователя, а также его групп

-Пример использования:

```
$ id
```

```
uid=1000(dali) gid=1000(dali)
```

```
groups=1000(dali),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),108(lxd),114(netdev)
```

getent

-Семантика: **get entries**

-Выводит список записей из БД, поддерживаемых библиотеками Name Service Switch

-Пример использования:

```
$ getent passwd s243864
```

```
s243864:x:243864:600:Nesterov Dali Konstantinovich:/home/s243864:/usr/bin/ksh
```

xargs

-Семантика: **extended arguments**

-Позволяет ввести аргументы для команды из стандартного ввода

-Основные ключи:

-0 аргументы разделяются NULL-символами, а не пробелами, кавычки и специальные символы воспринимаются буквально

-a file читать аргументы из файла file

-p каждый раз спрашивать запускать ли очередную команду

-n max_args использовать максимально возможное число аргументов но не более чем max_args

-Пример использования:

```
$ xargs touch
```

```
a
```

```
b
```

```
c
```

```
d
```

```
e
```

```
$ ls
```

```
a b c d e
```

Переменные окружения

Переменная окружения – текстовая переменная операционной системы, хранящая какую-либо информацию. Вот некоторые из них:

LANG, TZ, LC_* – переменные, содержащие информацию о локализации. Определяют вывод дат, времени, чисел и т. д. Используются в touch, ls для корректного отображения временных отметок

PATH – переменная, представляющая собой путь к набору каталогов с исполняемыми файлами, используется командой type – поиск пути для аргумента

HOME – содержит путь к домашнему каталогу текущего пользователя, используется командой cd – при вызове без аргументов происходит переход к домашнему каталогу

PWD – содержит путь к рабочему каталогу. Меняется с каждым новым вызовом cd

OLDPWD – содержит путь к предыдущему рабочему каталогу, то есть, значение PWD перед последним вызовом команды cd. Возврат по данному пути производится при вызове cd –

COLUMNS – ширина вывода для терминала – используется для форматирования вывода на терминал. Используется, например, в командах ls, more, less

Описание атрибутов и интерпретация прав доступа для команды ls -l

```
drwxrwxrwx 1 sundalik sundalik 4096 Mar  4 13:32 le_dir
-rw-rw-rw- 2 sundalik sundalik  14 Mar  4 13:32 test
-rw-rw-rw- 2 sundalik sundalik  14 Mar  4 13:32 test_link
lrwxrwxrwx 1 sundalik sundalik   4 Mar  4 13:32 test_linkS -> test
```

1) Первая буква – тип файла:

- обычный файл

d директория

l символическая ссылка

p именованный канал, позволяющий программам обмениваться данными.

особенностью является то, что порядок действий при обращении к пайпу не важен.

c,b символический и блочный файл. Блочные используются для передачи данных, разделенных на пакеты фиксированной длины, а символические отправляют данные сплошным потоком байт. Необходимы для взаимодействия с физическими устройствами

s сокет обеспечивает обмен данными между процессами и отличается от канала тем, что позволяет обмениваться разным ЭВМ соединенными сетью

2) Права доступа

r - чтение, w - запись, x - исполнение

По три буквы для пользователя, группы и всех остальных

3) Число жестких ссылок на файл

4) Имя владельца

5) Группа владельца

6) Размер файла в байтах

7) Дата последнего изменения файла

8) Имя файла (для символической ссылки: имя ссылки -> файл)

Права доступа:

u = user, g = group, o = other

r = read, w = write, x = execute

Для файла или жесткой ссылки:

r = читать содержимое файла

w = изменять содержимое файла

x = выполнить как программу

Для каталога:

r = просматривать содержимое

w = создавать файлы и каталоги внутри

x = заходить в каталог

Для символической ссылки:

Определяются правами файла, на который указывает ссылка

Различия в терминологии:

Команда - символьное обозначение действия для интерпретатора

Микрокоманды - элементарные действия, из которых состоят команды

Наноккоманда - команда, выполняющая самое элементарное действие

Программа - исполняемый файл, состоящий из набора команд

Утилита - программа для узкоспециализированной задачи

Дополнительное задание 1:

```
(tail file -n 2; head file -n 5) | tr '\n' ' ' >&2
```

Дополнительное задание 2:

```
(tail file -n 5 | head -n 1; echo 'to THAT --'; tail file -n3) | tr '\n' ' '
```

Вывод: В ходе лабораторной работы были изучены новые команды и освежены знания старых.