

Университет ИТМО  
Факультет программной инженерии и компьютерной техники  
Системное программное обеспечение

## **ЛАБОРАТОРНАЯ РАБОТА №1**

Группа: Р3202

Работу выполнил студент  
Коков Алексей Тимурович

2019 г.

## Выполнение

- Список команд, подлежащих повторению:

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

### 1. **ls [-options] [files]**

*Семантика* – LiSt. Не является фильтром.

Для каждого файла, который является каталогом, выдать список содержимого каталога. Для каждого файла, который представляет собой обычный файл, повторить его имя и любую другую запрашиваемую информацию. Вывод отсортирован в алфавитном порядке по умолчанию. Когда не заданы аргументы, указан текущий каталог.

*Основные ключи:*

- a - не игнорировать записи, начинающиеся с точки
- l - использование «длинного» формата записи
- 1 - печать одной записи в строке вывода
- r - обратный порядок сортировки
- R - рекурсивный вывод списка подкаталогов

*Переменные окружения:* LANG, COLUMNS, LC\_ALL, LC\_MESSAGES, NLSPATH, TZ.

### 2. **pwd**

*Семантика* – Print Working Directory. Не является фильтром.

Вывести полный путь от корневого каталога к текущему рабочему каталогу.

### 3. **cd [directory]**

*Семантика* – Change working Directory. Не является фильтром.

Изменить текущий рабочий каталог на каталог, указанный в аргументе directory.

*Основные ключи:*

- L – использовать физическую структуру каталогов
- P – следовать по символическим ссылкам

*Переменные окружения:* HOME, PWD, OLDPWD.

### 4. **rm [-options] [files]**

*Семантика* – ReMove. Не является фильтром.

Удалить указанные объекты.

*Основные ключи:*

- L – использовать физическую структуру каталогов
- P – следовать по символическим ссылкам

### 5. **mv [-options] [file1] [file2]**

*Семантика* – MoVe. Не является фильтром.

Переместить указанный объект file1 по направлению в file2.

*Основные ключи:*

- f – не выводить сообщения об ошибках
- i – запросить подтверждение для перемещения файла

## 6. **cp [-options] [file1] [file2]**

*Семантика* – CoPy. Не является фильтром.

Копировать (содержимое) file1 в file2.

*Основные ключи:*

-r, R - копировать рекурсивно

-f - разрешает удаление файла, в который производится копирование, если он не может быть открыт для записи

-i - запрашивать подтверждения при копировании

-p - копировать дату посл. изменения, идентификатор группы, права доступа исходного файла

-n - не перезаписывать существующий файл

## 7. **mkdir [-options] [files]**

*Семантика* – MaKe a DIRectory. Не является фильтром.

Создать каталог(и) с указанным именем в аргументе files.

*Основные ключи:*

-m – выставить права в формате chmod (например, 777)

-p – создает родительские каталоги, если они еще не существуют.

## 8. **rmdir [-options] [files]**

*Семантика* – ReMove a DIRectory. Не является фильтром.

Создать каталог(и) с указанным именем в аргументе files.

*Основные ключи:*

-p – удалить также родительские каталоги

## 9. **type [names]**

*Семантика* – write description of command TYPE. Не является фильтром.

Показать, каким образом каждый аргумент будет интерпретироваться, если он будет использован в качестве имени команды.

Переменные окружения: PATH.

## 10. **file [-options] [files]**

*Семантика* – determine a FILE type. Не является фильтром.

Определить тип файла.

*Основные ключи:*

-s – проверить файл на ошибки формата

-f ffile – ffile является списком файлов, подлежащих проверке

## 11. **find [path] [-options]**

*Семантика* – FIND. Не является фильтром.

Найти файлы по определенным свойствам.

*Некоторые операнды:*

-name – найти по имени

-type – найти по типу файла

-user – по пользователю

-group – по группе

-perm – по правам доступа

-links n – по количеству ссылок n

## 12. **chmod [-options] [files]**

*Семантика* – CHange the permissions MODe of a file. Не является фильтром.

Изменить права доступа к файлам/каталогам.

*Основные ключи:*

-f – не выводить сообщения об ошибках или запросах

-R – рекурсивно установить те же права доступа для каталогов и их содержимого

## 13. **ln [-options] [file1] [file2]**

*Семантика* – LiNk. Не является фильтром.

Создать жесткую (с ключом -s – символическую) ссылку на файл.

*Основные ключи:*

-s – создать символическую ссылку вместо жесткой

## 14. **wc [-options] [files]**

*Семантика* – Word Count. Является фильтром.

Вывести число переводов строк, слов и байт для каждого указанного файла, и итоговый подсчет, если было задано несколько файлов.

*Основные ключи:*

-l – вывести количество строк

-w – вывести количество слов

-m – вывести количество символов

-c – вывести количество байт

*Пример использования:*

```
$echo 'hello' | wc -lwmc
```

```
1 1 6 6
```

## 15. **tee [-options] [files]**

*Семантика* – Т-разветвитель в водопроводах, разделяющий воду на два потока. Является фильтром.

Считать со стандартного потока ввода и перенаправить на стандартный поток вывода и файлы, указанные в аргументах.

*Основные ключи:*

-a – добавлять данные в конец файла (не переписывая его)

-i – игнорировать ошибки

*Пример использования:*

```
$echo 'hello' | tee file1 file2
```

```
hello
```

## 16. **cat [-options] [files]**

*Семантика* – conCATenate. Является фильтром.

Вывести последовательно содержимое файлов (или стандартного потока ввода), объединяя их в единый поток.

*Основные ключи:*

-n – выводить номер каждой строки рядом с ней

-v – отображать непечатные символы

*Пример использования:*

```
$echo 'hello' | cat  
hello
```

#### 17. **tail [-options] [files]**

*Семантика* – tail (хвост). Является фильтром.

Вывести n (по умолчанию – 10) последних строк файлов (потока ввода).

*Основные ключи:*

-n – вывести n строк

*Пример использования:*

```
ls | tail -n1  
spl
```

#### 18. **head [-options] [files]**

*Семантика* – head (голова). Является фильтром.

Вывести n (по умолчанию – 10) первых строк файлов (потока ввода).

*Основные ключи:*

-n – вывести n строк

*Пример использования:*

```
ls | head -n1  
file
```

#### 19. **more (больше) [-options] [file]**

*Семантика* – more (больше). Является фильтром.

Вывести содержимое файла с пагинацией размером в экран.

*Основные ключи:*

-c – очистить экран перед выводом

-s – объединять несколько пустых строк в одну

-d – ожидание нажатия клавиш (пробел чтобы закончить, 'q' - для выхода)

*Пример использования:*

```
ls -l | more -c
```

#### 20. **pg [-options] [file]**

*Семантика* – PaGe. Является фильтром.

Выводить содержимое экрана постранично.

*Основные ключи:*

-c – очищать экран перед выводом

*Пример использования:*

```
ls -l | pg -c
```

#### 21. **touch [-options] [files]**

*Семантика* – touch (трогать). Не является фильтром.

Изменить последнее время доступа/изменения файла на текущее время/дату, либо создать пустой файл.

*Основные ключи:*

- a – изменить только время доступа
- m – изменить только время модификации
- с – если файла не существует, не создавать его

*Переменные окружения:* TZ (timezone).

- 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

### 1. **su – [username]**

*Семантика* – Switch User. Не является фильтром.

Войти в систему под другим именем, не завершая текущий сеанс.

*Основные ключи:*

- с command – выполнить одну команду command
- , -l – начать работу в среде, подобной реальному входу (начальной среде)

*Переменные окружения:* LC\_CTYPE (обработка символов), LC\_MESSAGES (вывод диагностических/информационных сообщений).

*Пример использования:*

su – nailstorm –с “touch test”

### 2. **chown [-options] [owner:[group]] [files]**

*Семантика* – CHange file OWNer. Не является фильтром.

Изменить владельца и/или группу для каждого указанного файла.

*Основные ключи:*

- R – изменить рекурсивно
- с – при каждом изменении выводить диагностику
- f – подавлять сообщения об ошибках

*Пример использования:*

chown root:staff /test

### 3. **chgrp [-options] [group] [files]**

*Семантика* – CHange file GRouP. Не является фильтром.

Изменить группу для каждого указанного файла.

*Основные ключи:*

- R – изменить рекурсивно
- с – при каждом изменении выводить диагностику
- f – подавлять сообщения об ошибках

*Пример использования:*

chgrp staff /test

### 4. **less [file]**

*Семантика* – less (меньше). Является фильтром.

Вывести содержимое файла с возможностью прокрутки.

*Основные ключи:*

- c – очистить экран перед выводом
- s – объединять несколько пустых строк в одну

*Переменные окружения:* HOME (домашний каталог), COLUMNS

*Пример использования:*

```
cat test1 test2 | less
```

## 5. **split [-options] [file [prefix]]**

*Семантика* – split (разделить). Не является фильтром.

Копировать и разделить содержимое файла (входного потока) по частям, префиксы названия которых указаны в prefix (если нет, то используется префикс x).

*Основные ключи:*

- l – количество строк
- d – вместо буквенных суффиксов использовать десятичные (начиная с 00)
- b [n](k) – количество байт, если указан k, то количество\*1024

*Пример использования:*

```
split -d /test newTest; ls  
newTest00 newTest01
```

## 6. **join [-options] [file1] [file2]**

*Семантика* – join (присоединиться). Является фильтром.

Объединить строки двух файлов, если у них имеются общие поля.

*Основные ключи:*

- i – игнорировать регистр
- 1/2 field – производить соединение по полю field файла 1/2

*Пример использования:*

```
echo "10 hello\n30 sos" > file1; echo "10 hi\n20 lol"; join file1 file2  
10 hello hi
```

## 7. **paste [-options] [files]**

*Семантика* – paste (вставить). Является фильтром.

Рассмотреть файлы, как вертикальные колонки, соединить их и вывести результат в стандартный поток вывода.

*Основные ключи:*

- d – изменить разделитель (по умолчанию табуляция)
- s – поменять расположение строк со столбцами

*Пример использования:*

```
paste file1 file2  
10 hello      10 hi  
30 sos        20 sos
```

## 8. **cut [-options] list [files]**

*Семантика* – cut (вырезать). Является фильтром.

Вывести определенные диапазоны строк из каждого файла в стандартный поток ввода.

*Основные ключи:*

- b – выбрать только приведенные байты
- c – выбрать только приведенные символы (-3; 4-6; 5-)
- f – выбрать только приведенные колонки
- d – изменить разделитель

*Пример использования:*

```
cat file3 | cut -f1 -d ' '
```

```
10
```

```
30
```

## 9. **tr (TRanslate or delete characters) [-options] [str1] [str2]**

*Семантика* – TRanslate or delete characters. Является фильтром.

Заменить или удалить выбранные символы.

*Основные ключи:*

- d – удалить все символы в приведенном наборе, str2 игнорируется
- s – заменить все символы из str1 на символы из str2

*Пример использования:*

```
echo 'abcdef' | tr -s 'a-c' 'd-f'
```

```
defdef
```

## 10. **cmp [-options] [files] [skip]**

*Семантика* – CoMPare. Является фильтром.

Сравнить два файла побитово. Если было обнаружено различие, сообщает о первом байте и строке, где оно было встречено.

*Основные ключи:*

- l – вывести различающиеся байты и их номера

*Пример использования:*

```
cmp file1 file2
```

```
file1 file2 differ: byte 5, line 1
```

## 11. **diff [-options] [file1] [file2]**

*Семантика* – DIFFerence. Является фильтром.

Построчно сравнить два файла (что изменить в file1, чтобы получить file2).

*Основные ключи:*

- w – игнорировать пробелы и табуляции при поиске изменений
- i – игнорировать регистр
- r – рекурсивное сравнение подкаталогов
- s – вывести отчет, если пары идентичны



*Пример использования:*

```
diff file1 file2
1,2c1,2
< 10 hello
< 30 sos
---
> 10 hi
> 20 sos
```

## 12. **patch [-options] < [file]**

*Семантика* – patch (заплата). Не является фильтром.

Перенести правки между разными версиями текстовых файлов.

*Основные ключи:*

- R – откатить изменения
- b – создать копию оригинала

*Пример использования:*

```
diff file1 file2 > file.patch; patch file2 < file.patch; cat file2
10 hello
30 sos
```

## 13. **sort [-options] [files]**

*Семантика* – sort (сортировать). Является фильтром.

Построчно отсортировать файлы, указанные в аргументах, вывести в стандартный поток вывода.

*Основные ключи:*

- r – обратный порядок
- f – не учитывать регистр
- b – игнорировать пробелы в начале сортируемых элементов

*Пример использования:*

```
echo 'b \nc \a' | sort
a
b
c
```

## 14. **uniq [-options] [in\_file] [out\_file]**

*Семантика* – UNIQue. Не является фильтром.

Убрать из вывода дублирующиеся строки.

*Основные ключи:*

- c – указывать число повторов
- d – выбрать только повторяющиеся строки
- u – выбрать только уникальные строки

*Пример использования:*

```
echo "1 \n1 \n2 \n2 \n2 \n3" | uniq -c
2 1
3 2
1 3
```

## 15. **echo [string]**

*Семантика* – echo (эхо). Не является фильтром.

Вывести строку текста в стандартный поток вывода.

*Пример использования:*

```
echo 'hello'
```

## 16. **alias [name[=value]]**

*Семантика* – alias (псевдоним). Не является фильтром.

Создать псевдоним для команды или конвейера.

*Пример использования:*

```
alias greet='echo "hello"'
```

## 17. **ulimit [-options] or -a**

*Семантика* – User LIMIT. Не является фильтром.

Установить/вывести ограничения на системные ресурсы.

*Основные ключи:*

-a – вывод ограничений

-v – макс. допустимый размер использования виртуальной памяти

-f – макс. размер создаваемого файла

-u – макс. число запущенных процессов

-d – макс. размер сегмента данных

*Пример использования:*

```
ulimit -a
```

```
virtual memory      (kbytes, -v) unlimited
```

```
max memory size    (kbytes, -m) unlimited
```

```
...
```

## 18. **umask [-s] [mask]**

*Семантика* – User MASK. Не является фильтром.

Установить маску прав доступа при создании новых файлов – сброс указанных битов прав доступа.

*Основные ключи:*

-S – задание в символическом виде (u=rwx)

*Пример использования:*

```
touch file1; umask 444; touch file2; ls -l
```

```
total 0
```

```
-rw-rw-r-- 1 nailstorm nailstorm 0 map  3 23:00 file1
```

```
--w--w--w- 1 nailstorm nailstorm 0 map  3 23:01 file2
```

## 19. **groups [-username]**

*Семантика* – groups (группы). Не является фильтром.

Вывести список групп для указанных пользователей или для текущего процесса.

*Пример использования:*

```
groups
```

```
nailstorm adm cdrom sudo dip plugdev lpadmin sambashare
```

## 20. **id [-options] [-user]**

*Семантика* – real and effective user and group IDs. Не является фильтром.

Получить информацию об указанном пользователе либо (если без параметров) о пользователе, запустившем программу.

*Пример использования:*

```
id
uid=1000(nailstorm) gid=1000(nailstorm) groups=1000(nailstorm), 4(adm), 24(cdrom), 27(sudo),
30(dip), 46(plugdev), 116(lpadmin), 126(sambashare)
```

## 21. **getent [-options] database [key]**

*Семантика* – GET ENTries. Не является фильтром.

Собрать элементы из административных баз данных.

*Пример использования:*

```
getent passwd nailstorm
nailstorm:x:1000:1000:Alexey,,,:/home/nailstorm:/bin/bash
```

## 22. **xargs [-options] [command [initial-arguments]]**

*Семантика* – build and execute command lines. Не является фильтром.

Объединяет набор заданных в командной строке аргументов с аргументами, прочитанными со стандартного ввода, и выполняет указанную команд.

*Основные ключи:*

-n – выполнить команду с не более чем n аргументами

-l - выполнять команду для каждой группы из заданного числа аргументов со стандартного ввода

*Пример использования:*

```
echo hi hello | xargs -n 1 echo
hi
hello
```

## **Переменные окружения**

LANG, TZ, LC\_\* - переменные, содержащие информацию о локализации (часовой пояс, язык и т.д.). Определяют вывод дат, времени, чисел. Используются в ls, touch для корректного отображения временных отметок.

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

HOME – переменная, содержащая путь к домашнему каталогу текущего пользователя, используется командой cd; при вызове без аргументов происходит переход к каталогу, указанному в этой переменной.

PWD - переменная, содержащая путь к рабочему каталогу. Меняется с каждым новым вызовом cd.

OLDPWD - переменная, содержащая путь к предыдущему рабочему каталогу - значение PWD перед последним вызовом команды cd. Возврат по данному пути производится при вызове cd –.

COLUMNS – переменная, содержащая ширину вывода для терминала, используется для форматирования вывода на терминал. Используется в командах ls, more, less.

## **Описание атрибутов и интерпретация прав доступа**

```
$ ls -l
```

|            |   |           |           |      |     |         |                   |
|------------|---|-----------|-----------|------|-----|---------|-------------------|
| -rw-rw-r-- | 1 | nailstorm | nailstorm | 0    | map | 3 23:00 | file1             |
| --w--w--w- | 1 | nailstorm | nailstorm | 0    | map | 3 23:01 | file2             |
| lrwxrwxrwx | 1 | nailstorm | nailstorm | 8    | map | 3 23:09 | link1 -> ../file1 |
| drwxrwxrwx | 2 | nailstorm | nailstorm | 4096 | map | 3 23:10 | testdir           |

### *Описание атрибутов*

#### 1) Тип файла

1. - - обычный файл
2. d - каталог
3. l - символическая ссылка
4. c - символьное устройство (для обращения к аппаратному обеспечению компьютера; последовательное обращение; пример - терминал)
5. b - блочное устройство (произвольное обращение; пример – floppy disk)
6. p - (pipe, конвейер) канал межпроцессорного взаимодействия
7. s - сокет

#### 2) Права доступа к файлу (в символьном виде):

последовательно для пользователя (u, user), группы (g, group), всех остальных (o, other)  
 r — read (читать), w — write (запускать), x — execute (выполнять)

#### 3) Количество жестких ссылок

#### 4) Имя владельца

#### 5) Группа владельца

#### 6) Размер файла в байтах

#### 7) Дата и время последнего изменения файла

#### 8) Имя файла (для символической ссылки - имя ссылки → имя файла, на который она ссылается)

*Модификаторы доступа для файлов и жестких ссылок:*

r — просматривать содержимое  
 w — записывать/редактировать файл  
 x — попытаться выполнить как программу

Для *символических ссылок* права определяются правами файла, на которой они указывают.

*Модификаторы доступа для каталога:*

r — просматривать содержимое  
 w — создавать файлы и каталоги внутри этого каталога  
 x — войти в каталог

## **Терминология**

*Команда* — символьное обозначение какого-либо действия для интерпретатора.

*Микрокоманда* — элементарные действия уровнем ниже команд (из них состоят команды).

*Нанокоманда* — команда, выполняющая наиболее элементарное действие (ниже микрокоманды).

*Программа* — исполняемый код/набор команд.

*Утилита* — программа, нацеленная на выполнение конкретной (узкоспециализированной) задачи.

## **Дополнительное задание №1**

(tail file -n2; head file -n5) | tr -s '\n' ' ' >&2

## **Дополнительное задание №2**

(head -n3 file | tail -n1; echo 'to THAT'; tail -n3 file | head -n1; echo ' -- '; tail -n2 file) | tr -s '\n' ' ' >&2

## **Выводы**

Shell дает в распоряжение огромное количество команд, так или иначе облегчающих работу пользователю. Например, при работе с текстовыми файлами в некоторых случаях использование предоставленных команд может заменить использование текстовых редакторов.