

Университет ИТМО

Кафедра ВТ

**Системное программное обеспечение**  
**Лабораторная работа №4**

Выполнил студент 2 курса  
Группы Р3211 Романов Олег  
Преподаватель: Дергачев А.М

Санкт-Петербург  
2016 год

Задания для выполнения лабораторной работы ориентированы на работу с файлами из каталога /usr/local/regex/

**Задание #1 (файл datebook):**

- 1. Вывести все строки, содержащие строку Sun**  
\$ grep 'Sun' datebook
- 2. Вывести все строки, где имена людей начинаются с J**  
\$ grep '^J' datebook
- 3. Вывести все строки, заканчивающиеся на 700**  
\$ grep '700\$' datebook
- 4. Вывести все строки, которые не содержат 834**  
\$ grep -v '834' datebook
- 5. Вывести все строки, с днем рождения в декабре**  
\$ grep '[:]12[/]' datebook
- 6. Вывести все строки с телефонными номерами, начинающимися с 408**  
\$ grep '[:]408[-]' datebook
- 7. Вывести все строки, содержащие следующую последовательность символов: заглавную букву, четыре строчные буквы, запятую, пробел и одну заглавную букву**  
\$ grep '[A-Z][a-z]{4}\{4\}, [A-Z]' datebook
- 8. Вывести все строки, в которых фамилия начинается с К или k**  
\$ grep '^[A-Z][a-z]\* [Kk]' datebook
- 9. Вывести все строки с их порядковыми номерами, где последнее числовое поле записи состоит из шести цифр**  
\$ grep -n '[:] [0-9]\{6\}\$' datebook
- 10. Вывести все строки, содержащие слова Lincoln или lincoln**  
\$ grep '[Ll]incoln' datebook

## Задание #2 (файл db):

**Разобрать команды и объяснить состав регулярного выражения и описать результаты вывода каждой из приведенных ниже команд:**

```
grep '\<Tom\>' db
```

При выполнении команды `grep '\<Tom\>' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `\<Tom\>` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, совпадающие со словом `Tom`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep 'Tom Savage' db
```

При выполнении команды `grep 'Tom Savage' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `Tom Savage` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, совпадающие со строкой `Tom Savage`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep '^Tommy' db
```

При выполнении команды `grep '^Tommy' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `^Tommy` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, начинающиеся со строки `Tommy`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep '\.bak$' db
```

При выполнении команды `grep '\.bak$' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `\.bak$` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, оканчивающиеся на строку `.bak`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep '[Pp]yramid' *
```

При выполнении команды `grep '[Pp]yramid' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `[Pp]yramid` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, совпадающие со строкой `Pyramid` или `pyramid`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep '[A-Z]' db
```

При выполнении команды `grep '[A-Z]' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `[A-Z]` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, в которых присутствует заглавной латинская буква. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep '[0-9]' db
```

При выполнении команды `grep '[0-9]' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `[0-9]` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, в которых присутствует цифра. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep '[A-Z]...[0-9]' db
```

При выполнении команды `grep '[A-Z]...[0-9]' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `[A-Z]...[0-9]` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, в которых присутствует конструкция вида: одна заглавная латинская буква, затем три любых символа и затем одна цифра. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep -w '[tT]est' db
```

При выполнении команды `grep -w '[tT]est' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `-w`, `[tT]est` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, в которых присутствуют слова `test` или `Test`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep -s 'Mark Todd' db
```

При выполнении команды `grep -s 'Mark Todd' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `-s`, `Mark Todd` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, содержащие `Mark Todd`. Ключ `-s` подавляет вывод сообщений об ошибке о несуществующих или недоступных для чтения файлах. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep -v 'Mary' db
```

При выполнении команды `grep -v 'Mary' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `-v`, `Mary` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, не совпадающие со строкой `Mary`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep -i 'sam' db
```

При выполнении команды `grep -i 'sam' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `-i`, `sam` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, в которых содержится подстрока `sam` без учета её регистра. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep -l 'Dear Boss' *
```

При выполнении команды `grep -l 'Dear Boss' *` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `-l`, `Dear Boss` и `*` соответственно. Утилита `grep` распознает аргументы и ищет во всех файлах текущей директории строки, содержащие подстроку `Dear Boss`. На вывод подаются имена файлов, в которых один и более раз встретилась данная строка. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep -n 'Tom' db
```

При выполнении команды `grep -n 'Tom' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `-n`, `Tom` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, содержащие подстроку `Tom`. Благодаря ключу `-n` перед каждой подходящей строкой в вывод записывается её номер в файле. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep "$name" db
```

При выполнении команды `grep "$name" db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов значение переменной `name` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, содержащие значением переменной `name`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
grep '$5' db
```

При выполнении команды `grep '$5' db` интерпретатор запускает утилиту `grep`, передавая в качестве аргументов строки `$5` и `db` соответственно. Утилита `grep` распознает аргументы и ищет в файле `db` строки, содержащие подстроку `$5`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
ps -ef| grep '^ *user1'
```

При выполнении команды `ps -ef | grep '^ *user1'` интерпретатор запускает команду `ps`, передавая в качестве аргумента строку `-ef` и направляет вывод этой команды на вход утилиты `grep` с аргументом `^ *user1`. Утилита `grep` распознает аргументы и ищет во входном потоке строки, начинающиеся с произвольного кол-ва пробелов и последующей строкой `user1`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `grep`, который сохраняется в переменной окружения `$?`

```
egrep '^ +' db
```

При выполнении команды `egrep '^ +' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `^ +` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, начинающиеся с одного и более пробелов. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep '^ *' db
```

При выполнении команды `egrep '^ *' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `^ *` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, начинающиеся с нуля или больше пробелов. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep '(Tom|Dan) Savage' db
```

При выполнении команды `egrep '(Tom|Dan) Savage' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `(Tom|Dan) Savage` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, содержащие подстроку `Tom Savage` или подстроку `Dan Savage`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep '(ab)+' db
```

При выполнении команды `egrep '(ab)+' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `(ab)+` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, содержащие одну или более подряд идущих подстрок `ab`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep '^X[0-9]?' db
```

При выполнении команды `egrep '^X[0-9]?' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `^X[0-9]?` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, начинающиеся с заглавной буквы `X`, за которой идет ноль или одна цифра. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep 'fun\.$' *
```

При выполнении команды `egrep 'fun\.$' *` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `fun\.$` и `*` соответственно. Утилита `egrep` распознает аргументы и ищет во всех файлах текущей директории строки, заканчивающиеся на подстроку `fun`, за которой стоит точка. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep '[A-Z]+' db
```

При выполнении команды `egrep '[A-Z]+' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `[A-Z]+` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, содержащие одну или больше заглавных букв латинского алфавита. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep '[0-9]' db
```

При выполнении команды `egrep '[0-9]' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `[0-9]` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, содержащие цифру. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep '[A-Z]...[0-9]' db
```

При выполнении команды `egrep '[A-Z]...[0-9]' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `[A-Z]...[0-9]` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, содержащие заглавную букву, за которой идут три любых символа и цифра. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep '[tT]est' db
```

При выполнении команды `egrep '[tT]est' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `[tT]est` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, содержащие подстроки `test` или `Test`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep '(Susan|Jean) Doe' db
```

При выполнении команды `egrep '(Susan|Jean) Doe' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `(Susan|Jean) Doe` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, содержащие подстроку `Susan Doe` или подстроку `Jean Doe`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep -v 'Mary' db
```

При выполнении команды `egrep -v 'Mary' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `-v`, `Mary` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, не содержащие подстроку `Mary`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep -i 'sam' db
```

При выполнении команды `egrep -i 'sam' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `-i`, `sam` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, содержащие подстроку `sam` в любом регистре. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep -l 'Dear Boss' *
```

При выполнении команды `egrep -l 'Dear Boss' *` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `-l`, `Dear Boss` и `*` соответственно. Утилита `egrep` распознает аргументы и ищет во всех файлах текущей директории строки, содержащие `Dear Boss`. На вывод подаются имена файлов, в которых один и более раз встретилась данная строка. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep -n 'Tom' db
```

При выполнении команды `egrep -n 'Tom' db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов строки `-n`, `Tom` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, содержащие подстроку `Tom`. Благодаря ключу `-n` перед каждой подходящей строкой в вывод записывается её номер в файле `db`. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`

```
egrep -s "$name" db
```

При выполнении команды `egrep -s "$name" db` интерпретатор запускает команду `egrep`, передавая в качестве аргументов значение переменной `name` и `db` соответственно. Утилита `egrep` распознает аргументы и ищет в файле `db` строки, содержащие значение переменной `name`. Благодаря ключу `-s` выводятся только сообщения об ошибках. Статус выполнения команды можно определить, проанализировав код завершения утилиты `egrep`, который сохраняется в переменной окружения `$?`