

Лабораторная работа №12

Презентация по выполненной работе

Атанесов Алекс

01 января 1970

Российский университет дружбы народов, Москва, Россия

Jan Watan Turkmenistan!

Информация

- Атанесов Александр Николаевич
- начинающий профессор , студент первого курса НБИ
- <https://github.com/NEGODAY7484>



Вводная часть

- Крайне важно и актуально

- Научиться программировать в OS UNIX

- Процессор **pandoc** для входного формата Markdown
- Результирующие форматы
 - pdf
 - html
- Автоматизация процесса создания: **Makefile**

Создание презентации

- Pandoc: преобразователь текстовых файлов
- Сайт: <https://pandoc.org/>
- Репозиторий: <https://github.com/jgm/pandoc>

- Использование LaTeX
- Пакет для презентации: beamer
- Тема оформления: **metropolis**

```
slide_level: 2  
aspectratio: 169  
section-titles: true  
theme: metropolis
```

- Используется фреймворк `reveal.js`
- Используется тема `beige`

- Тема задаётся в файле `Makefile`

```
REVEALJS_THEME = beige
```

Результаты

Выполнение лабораторной работы

1. Создаю файл command.sh. (рис. (fig:001?))

A terminal window with a dark background and light-colored text. The prompt is [aatanesov@fedora ~]\$ and the command being entered is touch command.sh.

```
[aatanesov@fedora ~]$ touch command.sh
```

Рис. 1: Использую команду touch

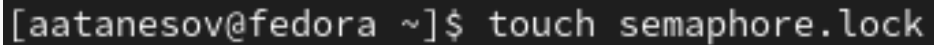
2. Делаю файл исполняемым. (рис. (fig:002?))

A terminal window with a dark background and light gray text. The prompt is [aatanesov@fedora ~]\$ and the command entered is chmod +x command.sh.

```
[aatanesov@fedora ~]$ chmod +x command.sh
```

Рис. 2: Использую команду chmod

3. Создаю файл semaphore.lock. (рис. (fig:003?))

A terminal window with a dark background and light gray text. The prompt is [aatanesov@fedora ~]\$ and the command being entered is touch semaphore.lock.

```
[aatanesov@fedora ~]$ touch semaphore.lock
```

Рис. 3: Использую команду touch

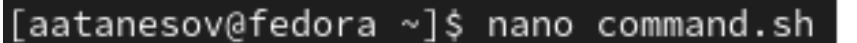
4.Создаю файл output.txt для автоматической записи изменений . (рис. (fig:004?))

A terminal window with a dark background. The prompt is [aatanesov@fedora ~]\$ and the command touch output.txt is entered.

```
[aatanesov@fedora ~]$ touch output.txt
```

Рис. 4: Использую команду touch

5. Открываю файл `command.sh` через `nano`. (рис. (fig:005?))

A terminal window with a dark background and light gray text. The prompt is `[aatanesov@fedora ~]$` followed by the command `nano command.sh`. A white cursor is positioned at the end of the command.

```
[aatanesov@fedora ~]$ nano command.sh
```

Рис. 5: Использую команду `nano`

6. Открываю файл `command.sh` через `nano`. (рис. (fig:006?))

A terminal window with a black background and white text. The prompt is [aatanesov@fedora lab12]\$ and the command nano command.sh has been entered.

```
[aatanesov@fedora lab12]$ nano command.sh
```

Рис. 6: Используя команду `nano`

7. Пишу необходимый код для выполнения условий задачи 1 . (рис. (fig:007?))

```
#!/bin/bash

# Проверяем наличие аргументов
if [ $# -lt 3 ] || [ $# -gt 4 ]; then
    echo "Usage: $0 <semaphore_name> <t1> <t2> [<output_device>]"
    exit 1
fi

# Задаем переменные
semaphore_name="$1"
t1="$2"
t2="$3"
semaphore_file="/tmp/${semaphore_name}.lock"

# Создаем семафор
if [ ! -e "${semaphore_file}" ]; then
    touch "${semaphore_file}"
    echo "Semaphore '${semaphore_name}' created!"
fi

# Захватываем семафор
while [ -e "${semaphore_file}" ]; do
    sleep "${t1}"
    echo "Waiting for semaphore '${semaphore_name}' to be released..."
done
touch "${semaphore_file}"
echo "Semaphore '${semaphore_name}' captured by process $$!"

# Используем ресурс
sleep "${t2}"
echo "Resource used for ${t2} seconds by process $$!"

# Освобождаем семафор
rm "${semaphore_file}"
echo "Semaphore '${semaphore_name}' released by process $$!"

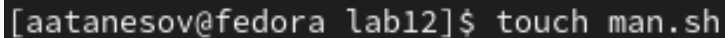
# Проверяем наличие аргумента для перенаправления вывода
if [ $# -eq 4 ] && [ -c "/dev/${4}" ]; then
    exec > "/dev/${4}" 2>&1
fi

# Запускаем процесс в фоновом режиме
while true; do
    if ! ./${0} "${semaphore_name}" "${t1}" "${t2}" "${4}" &>/dev/null; then
        echo "Failed to start background process!"
        exit 1
    fi
done
```

8. Запускаю файл commmand.sh с semaphore . (рис. (fig:008?))

```
[aatanesov@fedora lab12]$ sudo bash ./command.sh semaphore 1 5
[sudo] пароль для aatanesov:
Попробуйте ещё раз.
[sudo] пароль для aatanesov:
Semaphore 'semaphore' created!
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
Waiting for semaphore 'semaphore' to be released...
```

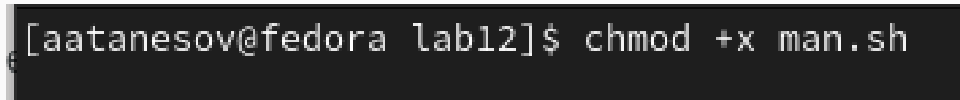
9. Создаю файл man.sh . (рис. (fig:009?))

A terminal window with a black background and white text. The prompt is [aatanesov@fedora lab12]\$ and the command being entered is touch man.sh.

```
[aatanesov@fedora lab12]$ touch man.sh
```

Рис. 9: Использую команду touch


10. Делаю файл исполняемым. (рис. (fig:010?))

A terminal window with a black background and white text. The prompt is '[aatanesov@fedora lab12]\$' and the command being entered is 'chmod +x man.sh'.

```
[aatanesov@fedora lab12]$ chmod +x man.sh
```

Рис. 10: Используя команду chmod

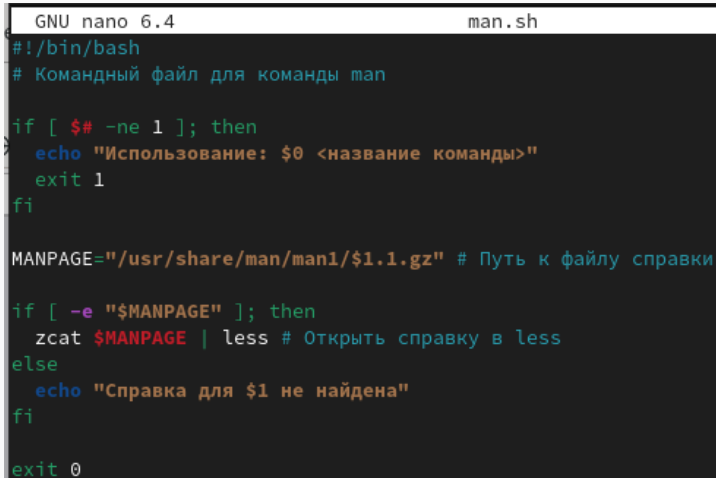
11. Открываю файл man.sh через nano. (рис. (fig:011?))

A terminal window with a black background and white text. The prompt is [aatanesov@fedora lab12]\$ and the command nano man.sh has been entered. A vertical cursor is visible at the end of the command.

```
[aatanesov@fedora lab12]$ nano man.sh
```

Рис. 11: Использую команду nano

12. Пишу необходимый код. (рис. (fig:012?))



```
GNU nano 6.4                                man.sh
#!/bin/bash
# Командный файл для команды man

if [ $# -ne 1 ]; then
    echo "Использование: $0 <название команды>"
    exit 1
fi

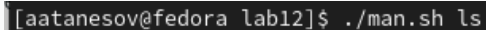
MANPAGE="/usr/share/man/man1/$1.1.gz" # Путь к файлу справки

if [ -e "$MANPAGE" ]; then
    zcat $MANPAGE | less # Открыть справку в less
else
    echo "Справка для $1 не найдена"
fi

exit 0
```

Рис. 12: Используя редактор nano

13. Запускаю файл `man.sh` с выводом справки для команды `ls`. (рис. (fig:013?))

A terminal window with a dark background. The prompt is [aatanesov@fedora lab12]\$ and the command being entered is ./man.sh ls. The text is white on a black background.

```
[aatanesov@fedora lab12]$ ./man.sh ls
```

Рис. 13: Использую команду `./`

14. Вижу вывод на команду ls. (рис. (fig:014?))

```
\" DO NOT MODIFY THIS FILE! It was generated by help2man 1.48.5.
TH LS "1" "January 2023" "GNU coreutils 9.1" "User Commands"
SH NAME
ls \- list directory contents
SH SYNOPSIS
B ls
[fI\,OPTION\/\fR]... [fI\,FILE\/\fR]...
SH DESCRIPTION
\" Add any additional description here
PP
list information about the FILES (the current directory by default).
Sort entries alphabetically if none of \fB\--cftuvSUX\fR nor \fB\--sort\fR is sp
ecified.
PP
Mandatory arguments to long options are mandatory for short options too.
TP
\fB\--a\fR, \fB\--all\fR
do not ignore entries starting with .
TP
\fB\--A\fR, \fB\--almost-all\fR
do not list implied . and ..
TP
\fB\--author\fR
with \fB\--l\fR, print the author of each file
TP
\fB\--b\fR, \fB\--escape\fR
print C\style escapes for nongraphic characters
TP
\fB\--block-size\fR=\fI\,SIZE\/\fR
with \fB\--l\fR, scale sizes by SIZE when printing them;
e.g., '\--block-size=M'; see SIZE format below
TP
```

(image/13.png){#fig:014 width=90%}

15. Запускаю файл `man.sh` с выводом справки для команды `rm`. (рис. (fig:015?))

A terminal window with a black background and white text. The prompt is `[aatanesov@fedora lab12]$` and the command entered is `./man.sh rm`.

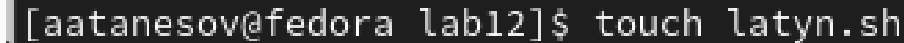
```
[aatanesov@fedora lab12]$ ./man.sh rm
```

Рис. 14: Открываю файл через `./`

16. Вижу вывод на команду rm. (рис. (fig:016?))

```
.\\" DO NOT MODIFY THIS FILE!  It was generated by help2man 1.48.5.
.TH RM "1" "January 2023" "GNU coreutils 9.1" "User Commands"
.SH NAME
rm \- remove files or directories
.SH SYNOPSIS
.B rm
[\fI\,OPTION\[/\fR]... [\fI\,FILE\[/\fR]...
.SH DESCRIPTION
This manual page
documents the GNU version of
.BR rm .
.B rm
removes each specified file.  By default, it does not remove
directories.
.P
If the \fI\-I\fR or \fI\--interactive=once\fR option is given,
and there are more than three files or the \fI\-r\fR, \fI\-R\fR,
or \fI\--recursive\fR are given, then
.B rm
prompts the user for whether to proceed with the entire operation.  If
the response is not affirmative, the entire command is aborted.
.P
Otherwise, if a file is unwritable, standard input is a terminal, and
the \fI\-f\fR or \fI\--force\fR option is not given, or the
\fI\-i\fR or \fI\--interactive=always\fR option is given,
.B rm
prompts the user for whether to remove the file.  If the response is
not affirmative, the file is skipped.
.SH OPTIONS
.PP
Remove (unlink) the FILE(s).
.TP
:
```

17. Создаю файл latyn.sh для выполнения третьего этапа. (рис. (fig:017?))

A terminal window with a black background and white text. The prompt is [aatanesov@fedora lab12]\$ and the command being entered is touch latyn.sh. A cursor is visible at the end of the command.

```
[aatanesov@fedora lab12]$ touch latyn.sh
```

Рис. 15: Использую команду ./

18. Делаю файл исполняемым . (рис. (fig:018?))

A terminal window with a black background and white text. The prompt is '[aatanesov@fedora lab12]\$' and the command being entered is 'chmod +x latyn.sh'.

```
[aatanesov@fedora lab12]$ chmod +x latyn.sh
```

Рис. 16: Использую команду chmod

19. Открываю файл latyn.sh через nano. (рис. (fig:019?))

A terminal window with a black background and white text. The prompt is [aatanesov@fedora lab12]\$ and the command nano latyn.sh has been entered. A small cursor icon is visible at the end of the command.

```
[aatanesov@fedora lab12]$ nano latyn.sh
```

Рис. 17: Использую команду nano

20. Ввожу необходимый код. (рис. (fig:020?))

```
#!/bin/bash

for i in {1..10} # генерируем 10 букв
do
    rand_num=$((RANDOM % 26)) # генерируем случайное число от 0 до 25
    letter=$(echo {A..Z} | cut -d ' ' -f $((rand_num + 1))) # находим букву по н>
    echo -n $letter # выводим букву без перевода строки
done
echo # перевод строки после последней буквы
```

[Прочитано 9 строк]

^G Справка

^O Записать

^W Поиск

^K Вырезать

^T Выполнить

^C Позиция

^X Выход

^R ЧитФайл

^\ Замена

^U Вставить

^J Выводить

^/ К строке

21. Запускаю файл latyn.sh . (рис. (fig:021?))

```
[aatanesov@fedora lab12]$ ./latyn.sh
URKHUTHRYZ
[aatanesov@fedora lab12]$ ./latyn.sh
NZRTDZUZDV
[aatanesov@fedora lab12]$ ./latyn.sh
JZQRUDXEEI
```

Рис. 19: Используя команду ./

Выводы

- Познал основы программирования в UNIX. # Ответы на контрольные вопросы
 - 1. Отсутствует пробел между скобками и оператором условия. Правильно: `while [$1 != "exit"]`.
 - 2. Можно использовать оператор конкатенации - символ "+" или переменную, содержащую объединяемые строки.
 - 3. Утилита `seq` используется для генерации числовых последовательностей. Её функционал можно реализовать с помощью циклов с оператором перебора и арифметических операций.
 - 4. Результатом вычисления будет число 3.3333333, но в `bash` результат целочисленного деления (оператор `//`) будет без округления, т.е. равным 3.
 - 5. `zsh` имеет более широкие возможности для настройки и расширения, например, более продвинутые автодополнение команд и параметров. Она также поддерживает более мощный синтаксис и некоторые удобные функции, например, поддержку массивов. Однако `bash` является более распространенным и стабильным в использовании.