

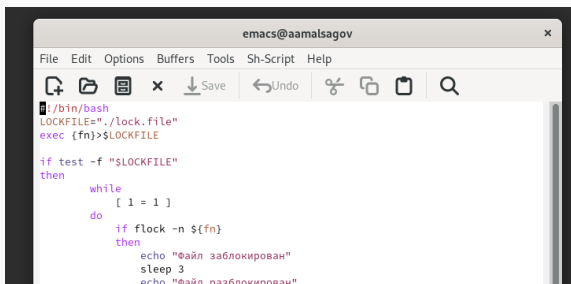
Программирование в командном процессоре ОС UNIX. Расширенное программирование

Мальсагов М.А.

Изучить основы программирования в оболочке ОС UNIX.
Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Выполнение лабораторной работы

1. Создал script1 и открыл его в emacs. Написал программу, реализующая упрощённый механизм семафоров. Командный файл в течение некоторого времени t1 дожидается освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует его в течение некоторого времени $t_2 < t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом.(рис. 1)

A screenshot of the Emacs text editor window titled 'emacs@aamalsagov'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The toolbar shows icons for file operations and editing. The script content is as follows:

```
#!/bin/bash
LOCKFILE="./lock.file"
exec {fn}>$LOCKFILE

if test -f "$LOCKFILE"
then
    while
        [ 1 = 1 ]
    do
        if flock -n ${fn}
        then
            echo "Файл заблокирован"
            sleep 3
            echo "Файл разблокирован"
```

Выполнение лабораторной работы

2. Проверил его работу. (рис. 2)

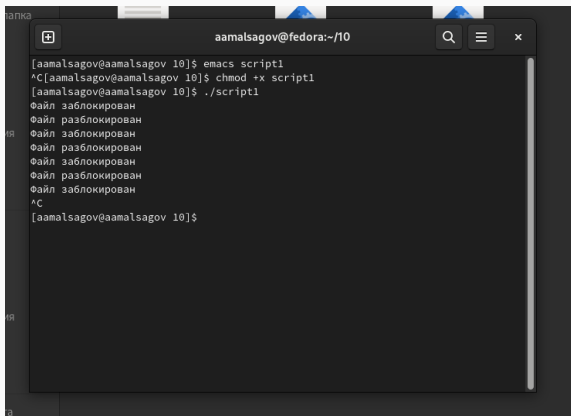
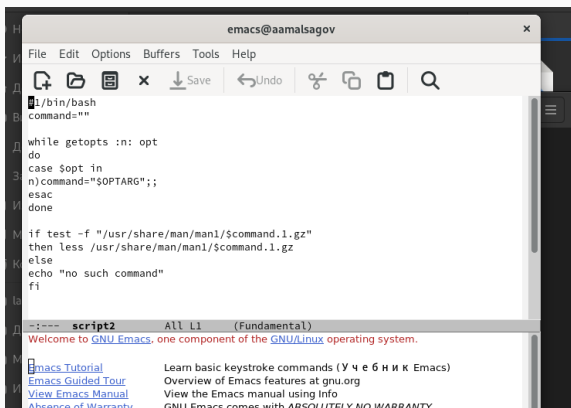
A terminal window titled 'aamalsagov@fedora:~/10' with standard window controls. The terminal shows a sequence of commands and their outputs. The user runs 'emacs script1', then '^C' to exit. Then they run 'chmod +x script1'. Finally, they run './script1', which produces six lines of Russian text: 'Файл заблокирован', 'Файл разблокирован', 'Файл заблокирован', 'Файл разблокирован', 'Файл заблокирован', and 'Файл разблокирован'. The session ends with '^C' and the prompt '[aamalsagov@aamalsagov 10]\$'.

Figure 2: Результат работы 1 скрипта

Выполнение лабораторной работы

3. Написал командный файл, который получает в виде аргумента командной строки название команды и в виде результата выдает справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге man1. (рис. 3)



The screenshot shows an Emacs window titled 'emacs@aamalsagov'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', and 'Help'. The toolbar contains icons for opening, saving, undo, redo, and search. The main text area displays a shell script:

```
#!/bin/bash
command=""

while getopts :n: opt
do
case $opt in
n) command="$OPTARG";;
esac
done

if test -f "/usr/share/man/man1/$command.1.gz"
then less /usr/share/man/man1/$command.1.gz
else
echo "no such command"
fi
```

Below the script, the Emacs startup screen is visible, showing the version '25.1.1.1', the file 'script2', and the mode '(Fundamental)'. It includes a welcome message: 'Welcome to GNU Emacs, one component of the GNU/Linux operating system.' and a list of links: 'Emacs Tutorial', 'Emacs Guided Tour', 'View Emacs Manual', and 'Absence of Warranty'. To the right of these links are instructions: 'Learn basic keystroke commands (Учебник Emacs)', 'Overview of Emacs features at gnu.org', 'View the Emacs manual using Info', and 'GNU Emacs comes with ABSOLUTELY NO WARRANTY.'

Выполнение лабораторной работы

4. Запустил скрипт.(рис. 4)

```
LS(1)                                User Commands                                LS(1)

ESC[1mNAMEESC[0m
ls - list directory contents

ESC[1mSYNOPSISESC[0m
ESC[1mls ESC[22m[ESC[4mOPTIONESC[24m]... [ESC[4mFILEESC[24m]...

ESC[1mDESCRIPTIONESC[0m
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of ESC[1m-cftuvSUX ESC[22mnor ESC[1m--sort ESC[22mfied.

Mandatory arguments to long options are mandatory for short options
too.

ESC[1m-aESC[22m, ESC[1m--allESC[0m
do not ignore entries starting with .

ESC[1m-AESC[22m, ESC[1m--almost-allESC[0m
do not list implied . and ..

ESC[1m--authorESC[0m
with ESC[1m-lESC[22m, print the author of each file

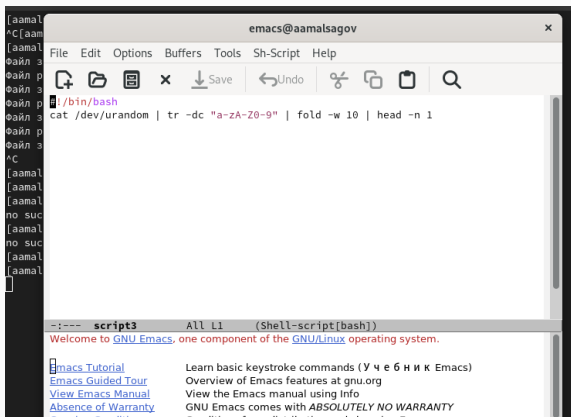
ESC[1m-bESC[22m, ESC[1m--escapeESC[0m
print C-style escapes for nongraphic characters

ESC[1m--block-sizeESC[22m=ESC[4mSIZEESC[0m
```

Figure 4: Работа скрипта

Выполнение лабораторной работы

5. Используя встроенную переменную \$RANDOM, написал командный файл, генерирующий случайную последовательность букв латинского алфавита.(рис. 5)



```
[aamal  
^C[aam  
[aamal  
Файл з  
Файл р  
Файл з  
Файл р  
Файл з  
Файл р  
Файл з  
^C  
[aamal  
[aamal  
[aamal  
no suc  
[aamal  
no suc  
[aamal  
[aamal  
[aamal  
]
```

emacs@aamalsagov

File Edit Options Buffers Tools Sh-Script Help

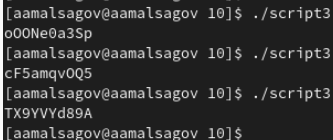
Save Undo

```
#!/bin/bash  
cat /dev/urandom | tr -dc "a-zA-Z0-9" | fold -w 10 | head -n 1
```

--- script3 All L1 (Shell-script[bash])
Welcome to [GNU Emacs](#), one component of the [GNU/Linux](#) operating system.
[Emacs Tutorial](#) Learn basic keystroke commands (Учебник Emacs)
[Emacs Guided Tour](#) Overview of Emacs features at gnu.org
[View Emacs Manual](#) View the Emacs manual using Info
[Absence of Warranty](#) GNU Emacs comes with **ABSOLUTELY NO WARRANTY**
[GNU Emacs](#) Conditions for distribution and copying Emacs

Figure 5: Код 3 скрипта

6. Проверил работу скрипта. (рис. 6)

A terminal window with a black background and white text. It shows three instances of running a script named './script3' from the directory of user 'aamalsagov' on host 'aamalsagov'. Each run produces a different alphanumeric string.

```
[aamalsagov@aamalsagov 10]$ ./script3  
o00Ne@a3Sp  
[aamalsagov@aamalsagov 10]$ ./script3  
cF5amqv0Q5  
[aamalsagov@aamalsagov 10]$ ./script3  
TX9YVYd89A  
[aamalsagov@aamalsagov 10]$
```

Figure 6: Запуск скрипта

Мы научились писать более сложные команды файлы.