

Титульный лист

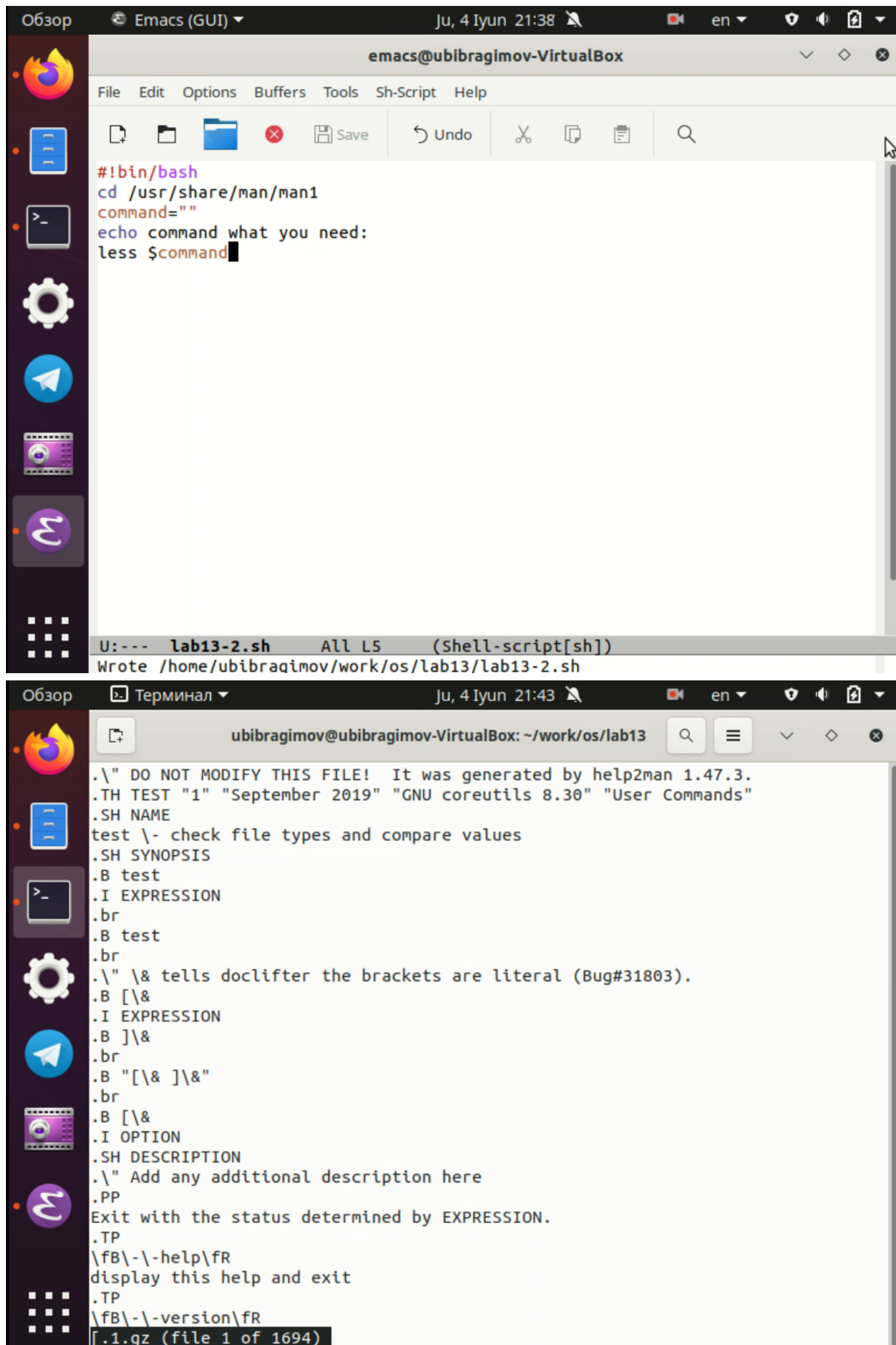
Отчёт по лабораторной работе №13 Ибрагимов Улугбек Ботырхонович

Цель работы

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

Ход работы.

1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени **t1** дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени **t2<>t1**, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой **> /dev/tty#**, где#— номер терминала куда перенаправляется вывод, в котором также запущен этот файл, но не в фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов
2. Реализовать команду **man** с помощью командного файла. Изучите содержимое каталога **/usr/share/man/man1**. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой **less** сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге **man1**



The image shows two terminal windows from a virtual machine named 'ubibragimov-VirtualBox'.

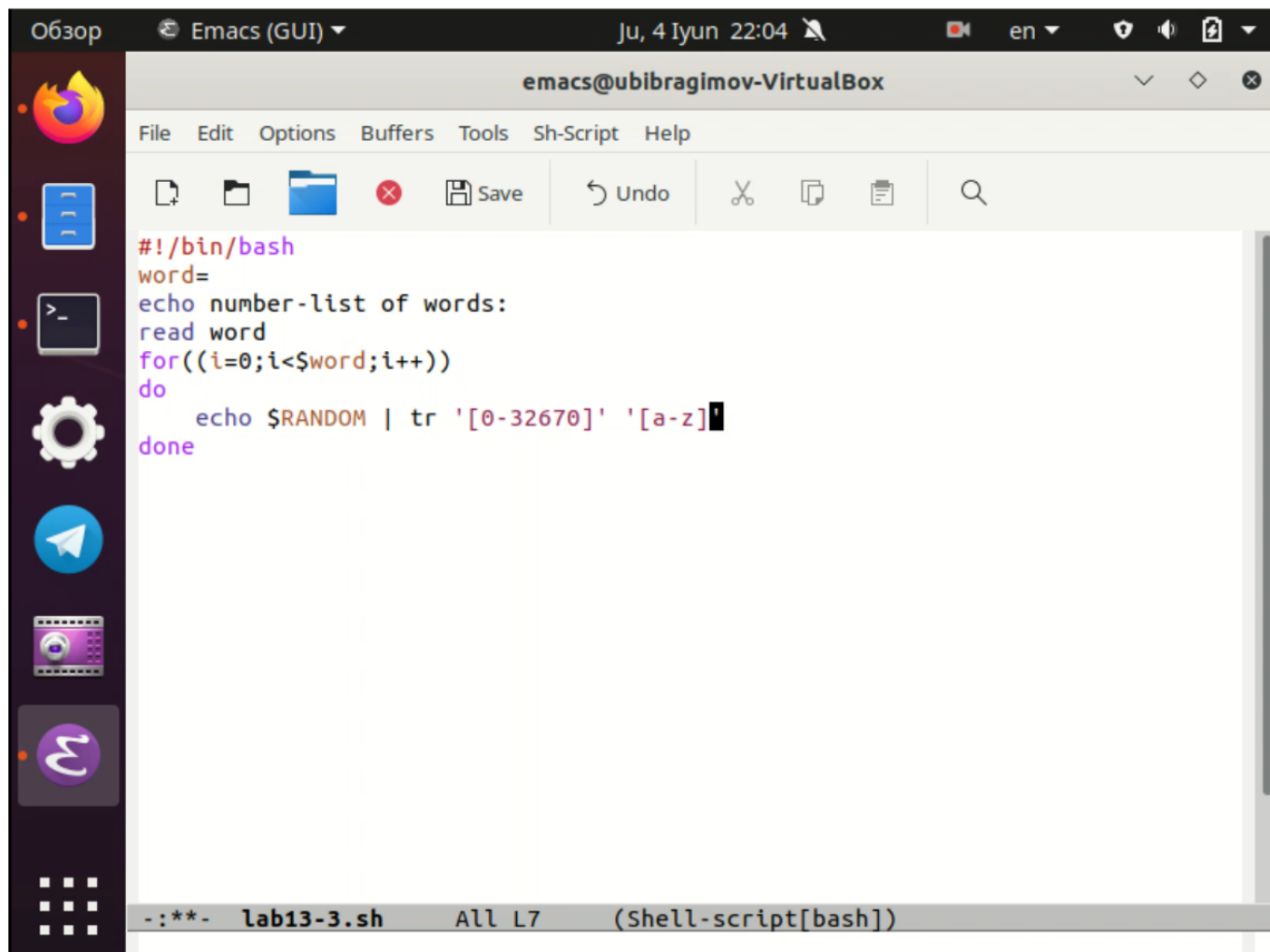
The top window is the Emacs GUI, titled 'emacs@ubibragimov-VirtualBox'. It shows a shell script being edited. The script content is:

```
#!/bin/bash
cd /usr/share/man/man1
command=""
echo command what you need:
less $command
```

The bottom window is a terminal titled 'ubibragimov@ubibragimov-VirtualBox: ~/work/os/lab13'. It shows the output of the script, which is a man page for the 'test' command. The output is:

```
.\ " DO NOT MODIFY THIS FILE!  It was generated by help2man 1.47.3.
.TH TEST "1" "September 2019" "GNU coreutils 8.30" "User Commands"
.SH NAME
test \- check file types and compare values
.SH SYNOPSIS
.B test
.I EXPRESSION
.br
.B test
.br
.\ " \& tells doclifter the brackets are literal (Bug#31803).
.B [\&
.I EXPRESSION
.B ]\&
.br
.B "[\& ]\&"
.br
.B [\&
.I OPTION
.SH DESCRIPTION
.\ " Add any additional description here
.PP
Exit with the status determined by EXPRESSION.
.TP
\fb\-\-help\fr
display this help and exit
.TP
\fb\-\-version\fr
[.1.gz (file 1 of 1694)
```

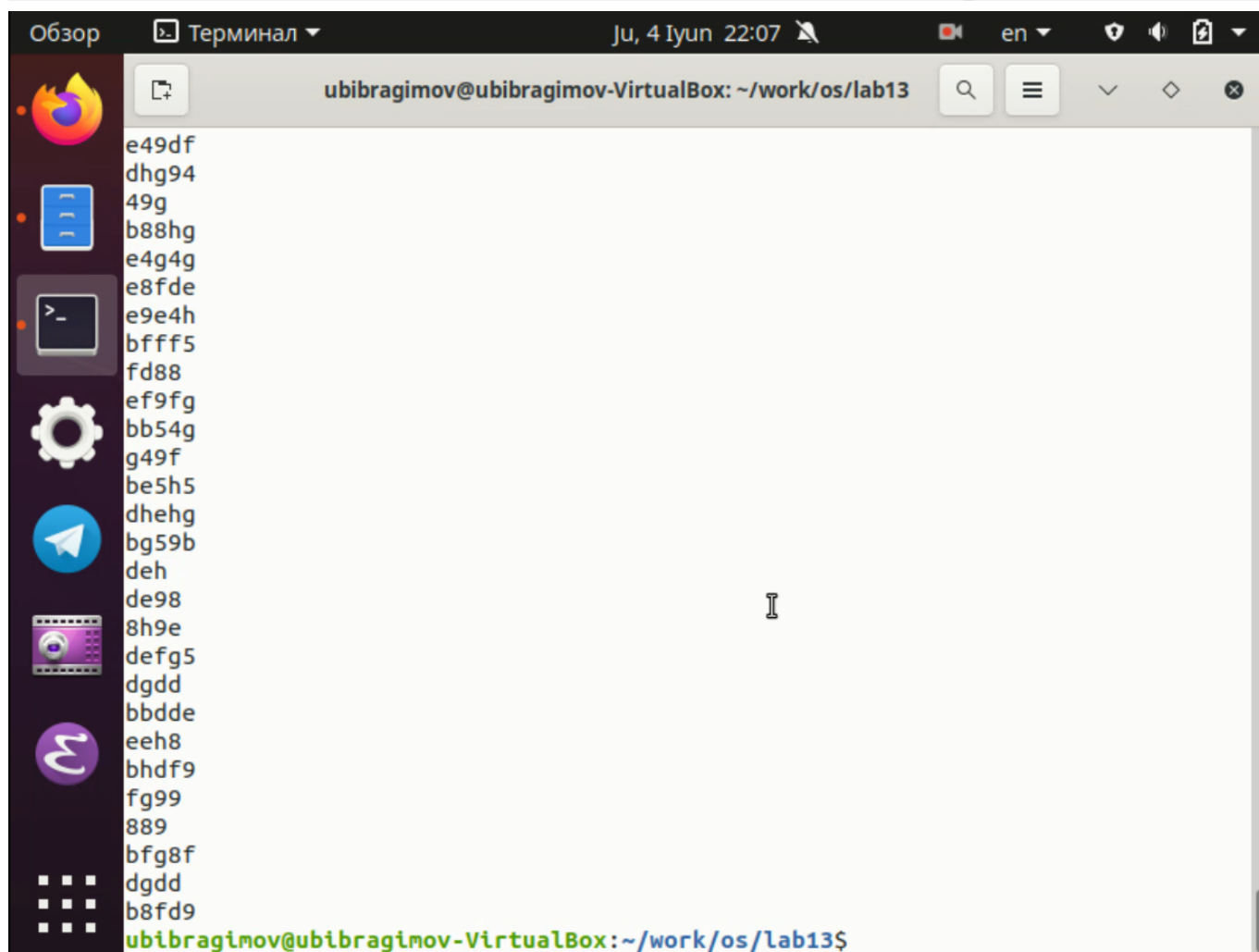
3. Используя встроенную переменную `$RANDOM`, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что `$RANDOM` выдаёт псевдослучайные числа в диапазоне от 0 до 32767



The screenshot shows the Emacs GUI window titled "emacs@ubibragimov-VirtualBox". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". The toolbar contains icons for file operations and editing. The main text area displays a shell script:

```
#!/bin/bash
word=
echo number-list of words:
read word
for((i=0;i<$word;i++))
do
    echo $RANDOM | tr '[0-32670]' '[a-z]'
done
```

The status bar at the bottom indicates the file is "lab13-3.sh", it is "All L7", and it is a "Shell-script[bash]".



The screenshot shows a terminal window titled "Терминал" with the prompt "ubibragimov@ubibragimov-VirtualBox: ~/work/os/lab13". The terminal displays the output of the script, which consists of 20 lines of random alphanumeric strings:

```
e49df
dhg94
49g
b88hg
e4g4g
e8fde
e9e4h
bfff5
fd88
ef9fg
bb54g
g49f
be5h5
dhehg
bg59b
deh
de98
8h9e
defg5
dgdd
bbdde
eeh8
bhdf9
fg99
889
bfg8f
dgdd
b8fd9
```

The prompt "ubibragimov@ubibragimov-VirtualBox: ~/work/os/lab13\$" is visible at the bottom.

Вывод

В ходе выполнения Лабораторной работы №13, были приобретены навыки по написанию сложных скриптов, взаимодействие с логикой работы операционной системы.

Контрольные вопросы

1. Для правильного исполнения командной строки `while [$1 != "exit"]` квадратные скобки нужно поменять на круглые скобки
2. Использовать знак `$` между двумя переменными символьного типа для объединения нескольких строк в одно единую целую строку
3. `seq` - выводит последовательность целых или действительных чисел для передачи в другие программы, на языке `bash` её можно реализовать с помощью цикла `for`
4. $10 / 3 = 3$ (дробная часть будет отброшена, так как числа целые, но если бы было $10.000/3.000 = 3.333$)
5. `Zsh` выполнен на основе `bash`, а поэтому наследует как недостатки так и преимущества `bash`, но в `zsh` есть автодополнение, горячие клавиши, алиасы, различные удобства в виде тем, плагинов и расширенной поддержке, однако есть минус всей этой красоты и удобства, на узкоспециализированных машинах это красоты быть не может в принципе из-за ограничений железа или системного администратора
6. Синтаксиси верен, однако для надежности лучше писать `for` и `((some words))` без пробела, т.е. слитно
7. Python vs Bash
 - Плюсы Python - является полным ООП-языком программирования, значит будет меньше проблем при написании кода, а также имеет удобства в виде автозаполнения, синтаксического сахара, кроссплатформенности, подходит для автоматизирования процессов разработки ПО
 - Минусы Python - у меня при работе с ним были проблемы по части создания переменных сред исполнения кода, поэтому с этой темой надо быть аккуратнее
 - Плюсы Bash - является низкоуровневым языком программирования, что позволяет писать скрипты приближенные к синтаксису C-language, подходит больше для системного администрирования
 - Минусы Bash - не является кроссплатформенным языком для написания скриптов