

Лабораторная работа №14

Программирование в командном процессоре ОС UNIX. Расширенное программирование.

Головина М.И.

17 мая 2025

Российский университет дружбы народов, Москва, Россия

Факультет Физико-математических и естественных наук

Информация

- Головина Мария Игоревна
- Бакалавр направления подготовки Математика и механика
- студентка группы НММбд - 02- 24
- Российский университет дружбы народов
- 1132246810@pfur.ru

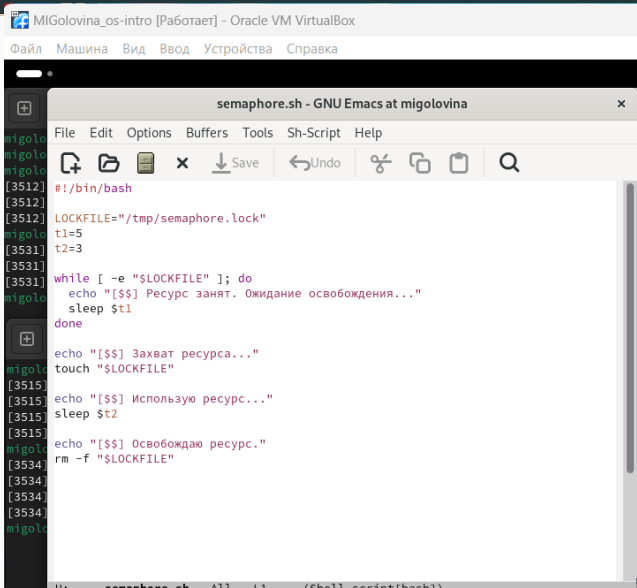


- Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

1. Написать командный файл, реализующий упрощённый механизм семафоров. 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 фоновом, а в привилегированном режиме.
2. Реализовать команду `map` с помощью командного файла.
3. Используя встроенную переменную `$RANDOM`, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита.
4. Ответить на контрольные вопросы.

Ход работы

Написала командный файл,
реализующий упрощённый механизм
семафоров

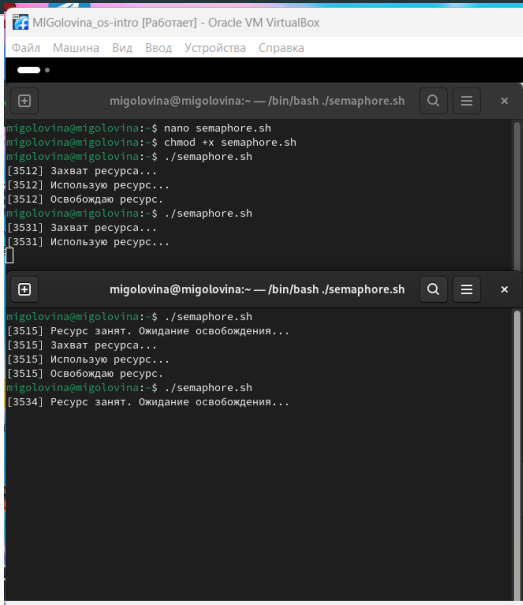


```
MIGolovina_os-intro [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

semaphore.sh - GNU Emacs at migolovina
File Edit Options Buffers Tools Sh-Script Help
[Icons: Save, Undo, Cut, Copy, Paste, Find]

migolo [3512] #!/bin/bash
migolo [3512] LOCKFILE="/tmp/semaphore.lock"
migolo [3512] t1=5
[3531] t2=3
[3531] while [ -e "$LOCKFILE" ]; do
migolo [3531]     echo "[$$] Ресурс занят. Ожидание освобождения..."
[3531]     sleep $t1
[3531] done
[3531] echo "[$$] Захват ресурса..."
migolo [3515] touch "$LOCKFILE"
[3515] echo "[$$] Использую ресурс..."
[3515] sleep $t2
[3515] echo "[$$] Освобождаю ресурс."
migolo [3534] rm -f "$LOCKFILE"
[3534]
[3534]
migolo
```

Запустила скрипт №1

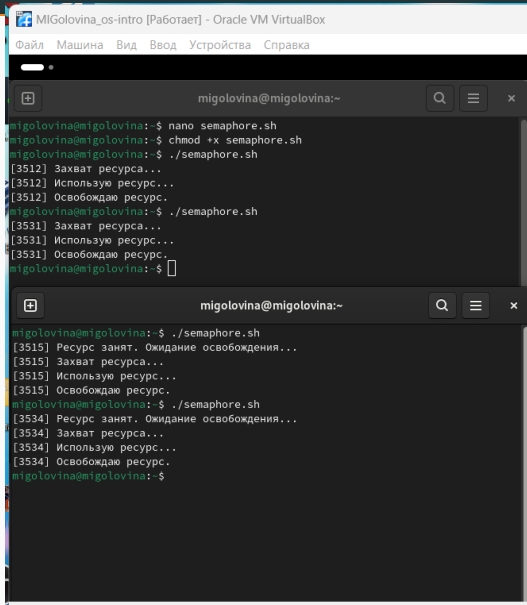


The screenshot shows a terminal window titled "MIGolovina_os-intro [Работает] - Oracle VM VirtualBox". The terminal displays the execution of a script named "semaphore.sh". The user runs "nano semaphore.sh", "chmod +x semaphore.sh", and then ". /semaphore.sh". The script output shows two processes, [3512] and [3531], each performing a sequence of actions: "Захват ресурса..." (Resource acquisition...), "Использую ресурс..." (Using resource...), and "Освобождаю ресурс." (Releasing resource.).

```
migolovina@migolovina:~ — /bin/bash ./semaphore.sh
migolovina@migolovina:~$ nano semaphore.sh
migolovina@migolovina:~$ chmod +x semaphore.sh
migolovina@migolovina:~$ ./semaphore.sh
[3512] Захват ресурса...
[3512] Использую ресурс...
[3512] Освобождаю ресурс.
migolovina@migolovina:~$ ./semaphore.sh
[3531] Захват ресурса...
[3531] Использую ресурс...

migolovina@migolovina:~ — /bin/bash ./semaphore.sh
migolovina@migolovina:~$ ./semaphore.sh
[3515] Ресурс занят. Ожидание освобождения...
[3515] Захват ресурса...
[3515] Использую ресурс...
[3515] Освобождаю ресурс.
migolovina@migolovina:~$ ./semaphore.sh
[3534] Ресурс занят. Ожидание освобождения...
```


Запустила скрипт №1

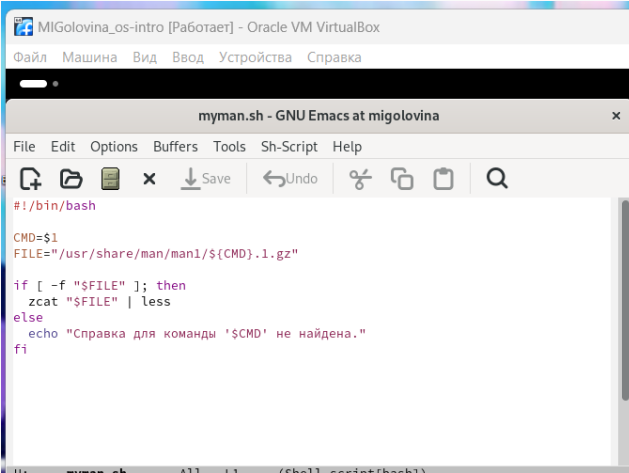


The screenshot shows a terminal window titled "MIGolovina_os-intro [Работает] - Oracle VM VirtualBox". The terminal displays the execution of a script named "semaphore.sh". The script uses the "sleep" command to simulate resource acquisition and release. The output shows two processes, [3512] and [3531], successfully acquiring and releasing the resource. A second instance of the script is then run, showing processes [3515] and [3534] waiting for the resource to be released before acquiring it.

```
migolovina@migolovina:~$ nano semaphore.sh
migolovina@migolovina:~$ chmod +x semaphore.sh
migolovina@migolovina:~$ ./semaphore.sh
[3512] Захват ресурса...
[3512] Использую ресурс...
[3512] Освобождаю ресурс.
migolovina@migolovina:~$ ./semaphore.sh
[3531] Захват ресурса...
[3531] Использую ресурс...
[3531] Освобождаю ресурс.
migolovina@migolovina:~$

migolovina@migolovina:~$ ./semaphore.sh
[3515] Ресурс занят. Ожидание освобождения...
[3515] Захват ресурса...
[3515] Использую ресурс...
[3515] Освобождаю ресурс.
migolovina@migolovina:~$ ./semaphore.sh
[3534] Ресурс занят. Ожидание освобождения...
[3534] Захват ресурса...
[3534] Использую ресурс...
[3534] Освобождаю ресурс.
migolovina@migolovina:~$
```

Реализовала команду man с
помощью командного файла

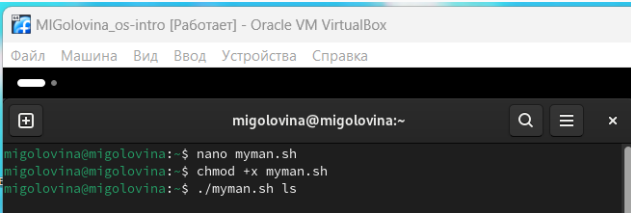


```
#!/bin/bash

CMD=$1
FILE="/usr/share/man/man1/${CMD}.1.gz"

if [ -f "$FILE" ]; then
    zcat "$FILE" | less
else
    echo "Справка для команды '$CMD' не найдена."
fi
```

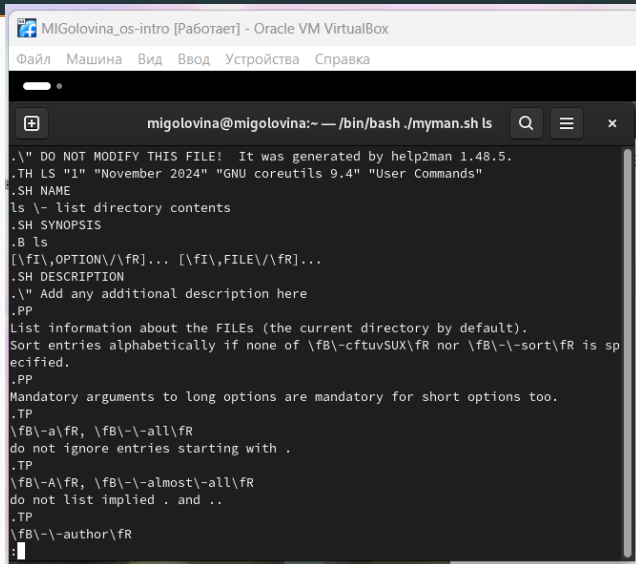
Запустила скрипт №2



The screenshot shows a terminal window titled "MIGolovina_os-intro [Работает] - Oracle VM VirtualBox". The terminal interface has a dark background with a top bar containing the username "migolovina@migolovina:~" and search, menu, and close icons. The terminal history shows the following commands and their outputs:

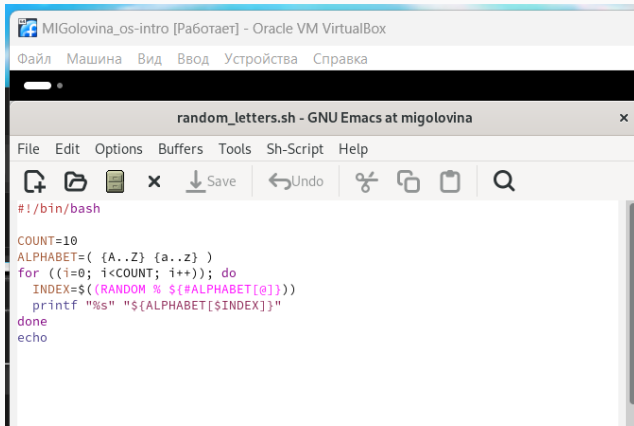
```
migolovina@migolovina:~$ nano myman.sh
migolovina@migolovina:~$ chmod +x myman.sh
migolovina@migolovina:~$ ./myman.sh ls
```

Результат выполнения скрипта №2



```
migolovina@migolovina:~ — /bin/bash ./myman.sh ls
.\" DO NOT MODIFY THIS FILE!  It was generated by help2man 1.48.5.
.TH LS "1" "November 2024" "GNU coreutils 9.4" "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
:
```

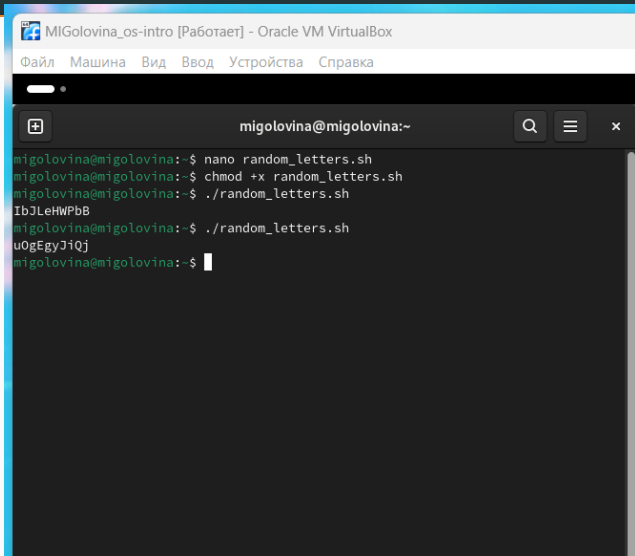
Используя встроенную переменную \$RANDOM, написала командный файл, генерирующий случайную последовательность букв латинского алфавита



```
#!/bin/bash

COUNT=10
ALPHABET=( {A..Z} {a..z} )
for ((i=0; i<COUNT; i++)); do
    INDEX=$((RANDOM % ${#ALPHABET[@]}))
    printf "%s" "${ALPHABET[$INDEX]}"
done
echo
```

Запустила скрипт №3



The screenshot shows a terminal window titled "MIGolovina_os-intro [Работает] - Oracle VM VirtualBox". The terminal interface has a dark background with a search icon, a menu icon, and a close icon in the top right corner. The prompt is "migolovina@migolovina:~". The following commands and their outputs are shown:

```
migolovina@migolovina:~$ nano random_letters.sh
migolovina@migolovina:~$ chmod +x random_letters.sh
migolovina@migolovina:~$ ./random_letters.sh
IbJLeHWPbB
migolovina@migolovina:~$ ./random_letters.sh
uOgEgyJiQj
migolovina@migolovina:~$
```

Ответы на контрольные вопросы

Ответы на контрольные вопросы

1. Найдите синтаксическую ошибку в следующей строке: `while [$1 != "exit"]`.

\$1. Так же между скобками должны быть пробелы. В противном случае скобки и рядом стоящие символы будут восприниматься как одно целое

2. Как объединить (конкатенация) несколько строк в одну?

```
migolovina@migolovina:~$ cat file.txt | xargs | sed -e 's/\n /./g'
```

3. Найдите информацию об утилите `seq`. Какими иными способами можно реализовать её функционал при программировании на `bash`?

`Seq` - выдает последовательность чисел. Реализовать ее функционал можно командой `for n in {1..5} do done`

4. Какой результат даст вычисление выражения $\$((10/3))$?

Вывод

Я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Дорогу осилит идущий
