

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №12 =====

дисциплина: Операционные системы

Студент: Койфман Кирилл Дмитриевич

Группа: НПИбд-01-21

## Введение.

---

### Цель работы.

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

### Задачи.

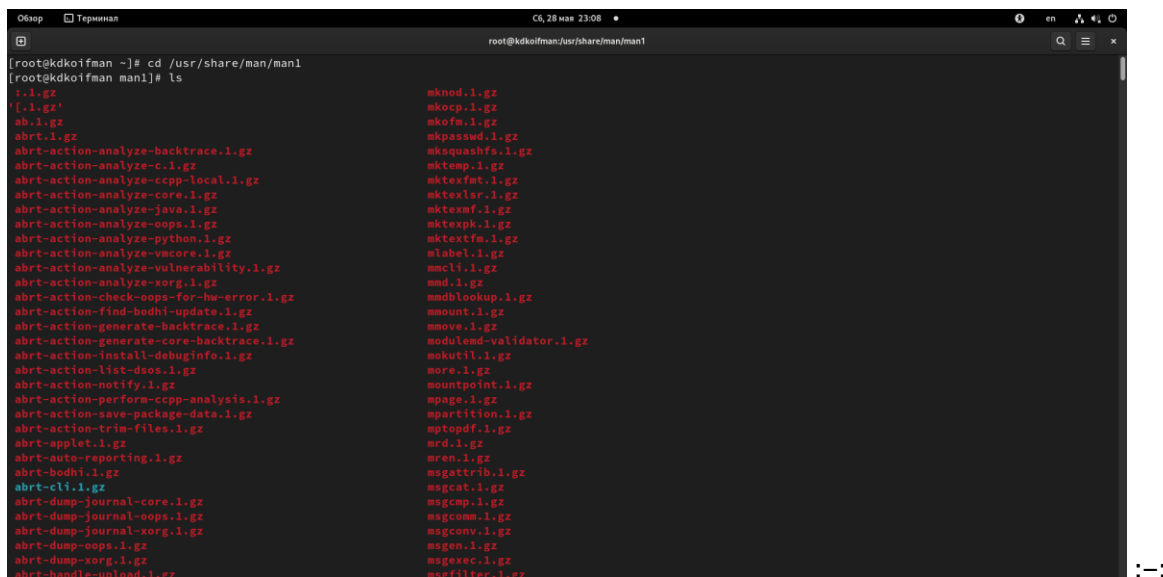
1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени  $t_1$  дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени  $t_2 < t_1$ , также выдавая информацию 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 фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов.
2. Реализовать команду `man` с помощью командного файла. Изучить 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`.

- 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.

## Ход работы.

### 2 задание.

Реализуем команду man с помощью командного файла. Изучим 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(рис.1,2,3,4,5,6)



```
[root@kdkoifman ~]# cd /usr/share/man/man1
[root@kdkoifman man1]# ls
i.1.gz
'.1.gz'
ab.1.gz
abrt.1.gz
abrt-action-analyze-backtrace.1.gz
abrt-action-analyze-c.1.gz
abrt-action-analyze-cpp-local.1.gz
abrt-action-analyze-core.1.gz
abrt-action-analyze-java.1.gz
abrt-action-analyze-oops.1.gz
abrt-action-analyze-python.1.gz
abrt-action-analyze-vmcore.1.gz
abrt-action-analyze-vulnerability.1.gz
abrt-action-analyze-xorg.1.gz
abrt-action-check-oops-for-hw-error.1.gz
abrt-action-find-bodhi-update.1.gz
abrt-action-generate-backtrace.1.gz
abrt-action-generate-core-backtrace.1.gz
abrt-action-install-debuginfo.1.gz
abrt-action-list-dsos.1.gz
abrt-action-notify.1.gz
abrt-action-perform-cpp-analysis.1.gz
abrt-action-save-package-data.1.gz
abrt-action-trim-files.1.gz
abrt-applet.1.gz
abrt-auto-reporting.1.gz
abrt-bodhi.1.gz
abrt-clt.1.gz
abrt-dump-journal-core.1.gz
abrt-dump-journal-oops.1.gz
abrt-dump-journal-xorg.1.gz
abrt-dump-oops.1.gz
abrt-dump-xorg.1.gz
abrt-handle-unload.1.gz
mknod.1.gz
mkoep.1.gz
mkoem.1.gz
mksquashfs.1.gz
mktemp.1.gz
mktextfm.1.gz
mktextlr.1.gz
mktextf.1.gz
mktextpk.1.gz
mktextfm.1.gz
mlabel.1.gz
mmcli.1.gz
mmd.1.gz
mmdlookup.1.gz
mmount.1.gz
mmove.1.gz
modulemd-validator.1.gz
mokutil.1.gz
more.1.gz
mountpoint.1.gz
mpage.1.gz
mpartition.1.gz
mptopdf.1.gz
mrd.1.gz
mren.1.gz
msgattrib.1.gz
msgcat.1.gz
msgcmp.1.gz
msgcomm.1.gz
msgconv.1.gz
msgen.1.gz
msgexec.1.gz
msgfilter.1.gz
```

рис.1(изучим содержимое каталога /usr/share/man/man1)

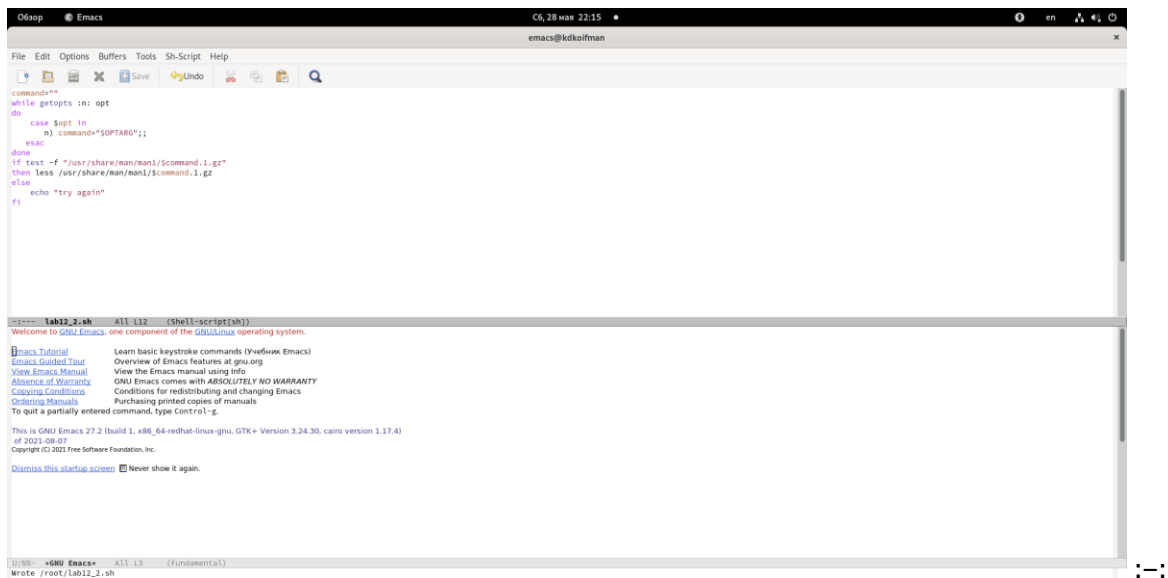


рис.2(код командного файла)

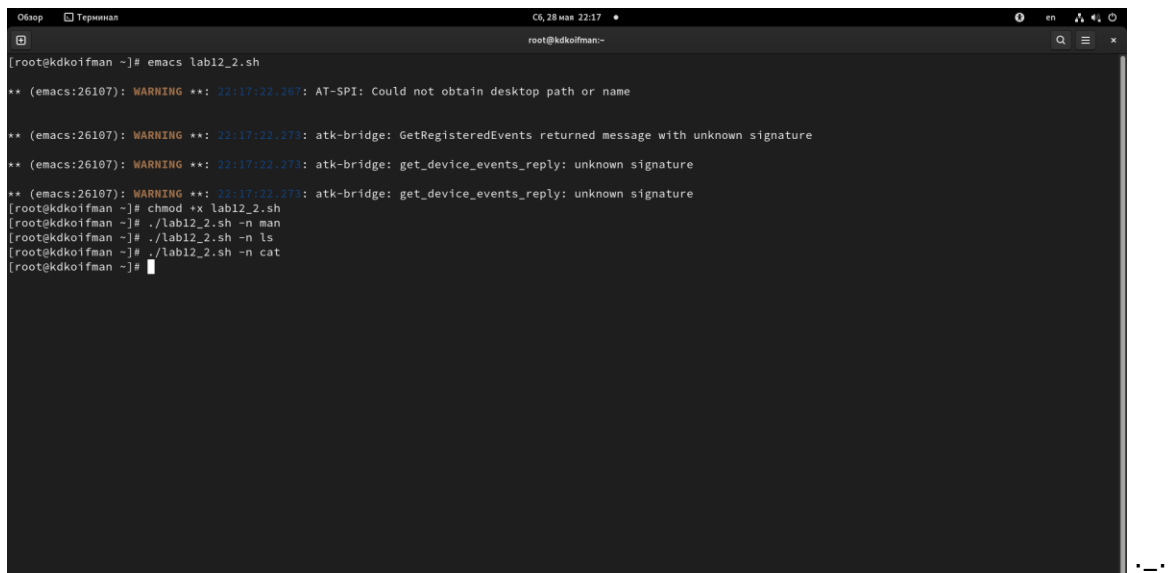


рис.3(выполнение командного файла(тестируем прототип команды man))

```
Обзор Терминал C6, 28 мая 22:17 root@kdkoilman:~
MAN(1) Manual pager utils
ESC[1mNAMEESC[0m
man - an interface to the system reference manuals

ESC[1mSYNOPSISESC[0m
ESC[1mman ESC[22mESC[4mmanESC[24m ESC[4moptionsESC[24m [[ESC[4msectionESC[24m] ESC[4mpageESC[24m ...] ...
ESC[1mman -k ESC[22mESC[4maproposESC[24m ESC[4moptionsESC[24m] ESC[4mregexpESC[24m ...
ESC[1mman -K ESC[22mESC[4mmanESC[24m ESC[4moptionsESC[24m] [ESC[4msectionESC[24m] ESC[4mtermESC[24m ...
ESC[1mman -f ESC[22mESC[4mwhatISC[24m ESC[4moptionsESC[24m] ESC[4mpageESC[24m ...
ESC[1mman -l ESC[22mESC[4mmanESC[24m ESC[4moptionsESC[24m] ESC[4mfileESC[24m ...
ESC[1mman -w ESC[22mESC[1m-W ESC[22mESC[4mmanESC[24m ESC[4moptionsESC[24m] ESC[4mpageESC[24m ...

ESC[1mDESCRIPTIONESC[0m
ESC[1mman ESC[22mis the system's manual pager. Each ESC[4mpageESC[24m argument given to ESC[1man ESC[22mis normally the name of a program, utility or
function. The ESC[4manualESC[24m ESC[4mpageESC[24m associ-
ated with each of these arguments is then found and displayed. A ESC[4msectionESC[24m, if provided, will direct ESC[1man ESC[22mto look only in that
ESC[4msectionESC[24m of the manual.
The default action is to search in all of the available ESC[4msectionsESC[24m following a pre-defined order (see ESC[1mDEFAULTSESC[22m), and to sh
ow only the first ESC[4mpageESC[0m
found, even if ESC[4mpageESC[24m exists in several ESC[4msectionsESC[24m.

The table below shows the ESC[4msectionESC[24m numbers of the manual followed by the types of pages they contain.

1 Executable programs or shell commands
2 System calls (functions provided by the kernel)
3 Library calls (functions within program libraries)
4 Special files (usually found in ESC[4mdevESC[24m)
5 File formats and conventions, e.g. ESC[4m/etc/passwdESC[0m
6 Games
7 Miscellaneous (including macro packages and conventions), e.g. ESC[1manESC[22m(7), ESC[1mgroffESC[22m(7)
8 System administration commands (usually only for root)
9 Kernel routines [Non standard]

/usr/share/man/man1/man.1.gz
```

:-:

рис.4(выполнение командного файла в сочетании с командой *man*(мы вывели справку по команде *man*))

```
Обзор Терминал C6, 28 мая 22:17 root@kdkoilman:~
LS(1) User Commands
ESC[1mNAMEESC[0m
ls - list directory contents

ESC[1mSYNOPSISESC[0m
ESC[1mls ESC[22mESC[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[22mon ESC[1m--sort ESC[22mis specified.

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

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

ESC[1m--ESC[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[22mESC[4mSIZEESC[0m
with ESC[1m-lESC[22m, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

ESC[1m--ESC[22m, ESC[1m--ignore-backupsESC[0m
do not list implied entries ending with ~

ESC[1m-c ESC[22mwith ESC[1m-lESC[22m: sort by, and show, ctime (time of last modification of file status information); with ESC[1m-lESC[22m: show
ctime and sort by name; otherwise:
/usr/share/man/man1/ls.1.gz
```

:-:

рис.5(выполнение командного файла в сочетании с командой *ls*(мы вывели справку по команде *ls*))

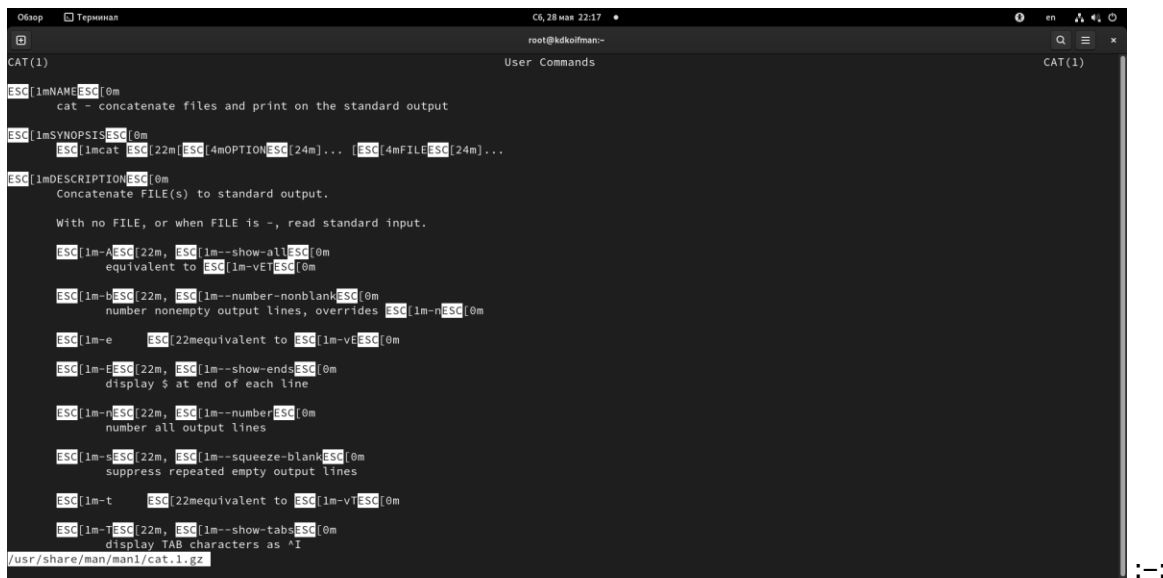


рис.6(выполнение командного файла в сочетании с командой cat(мы вывели справку по команде cat))

### 3 задание.

Используем встроенную переменную \$RANDOM, напомним командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтём, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767(рис.7,8)

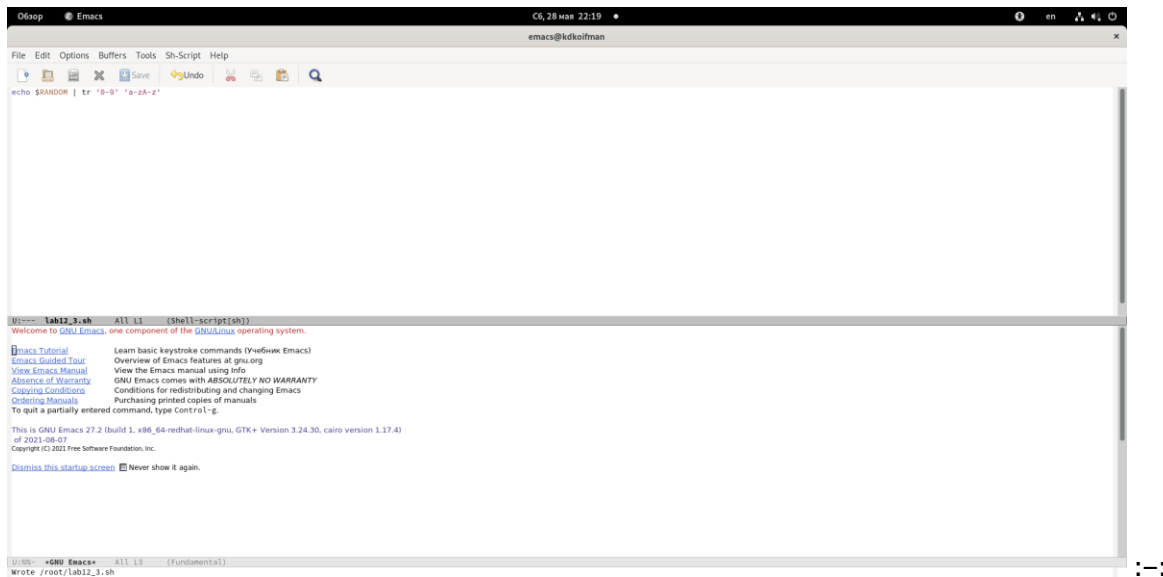
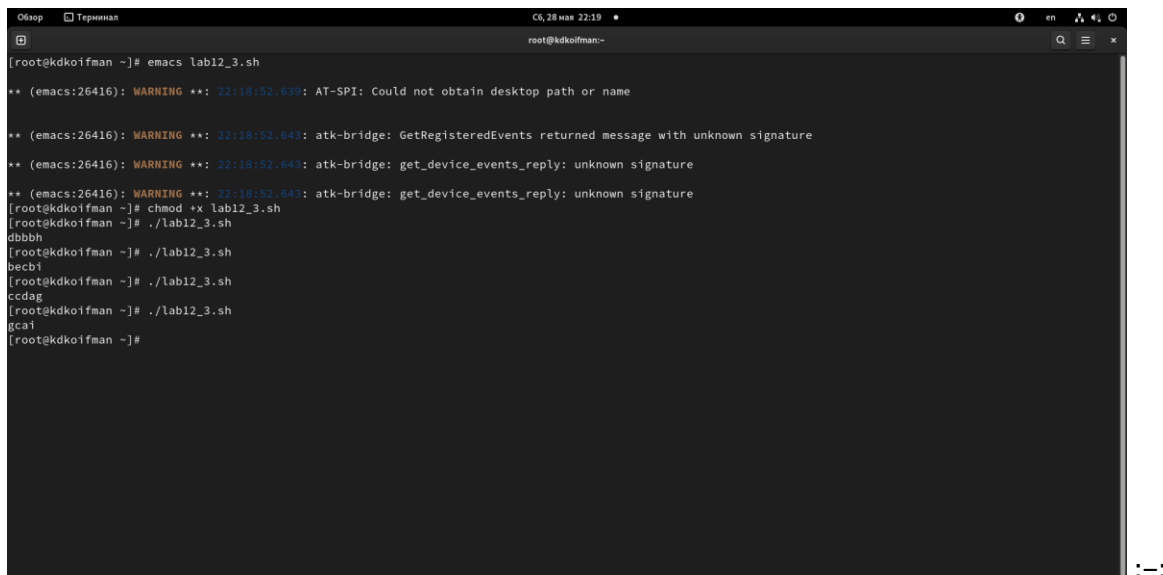


рис.5(код командного файла)



```
[root@kdkoifman ~]# emacs lab12_3.sh
** (emacs:26416): WARNING **: 22:18:52.630: AT-SPI: Could not obtain desktop path or name

** (emacs:26416): WARNING **: 22:18:52.643: atk-bridge: GetRegisteredEvents returned message with unknown signature

** (emacs:26416): WARNING **: 22:18:52.643: atk-bridge: get_device_events_reply: unknown signature

** (emacs:26416): WARNING **: 22:18:52.643: atk-bridge: get_device_events_reply: unknown signature
[root@kdkoifman ~]# chmod +x lab12_3.sh
[root@kdkoifman ~]# ./lab12_3.sh
dbbhh
[root@kdkoifman ~]# ./lab12_3.sh
becb1
[root@kdkoifman ~]# ./lab12_3.sh
ccdag
[root@kdkoifman ~]# ./lab12_3.sh
gca1
[root@kdkoifman ~]#
```

рис.6(выполнение командного файла)

## Вывод.

В ходе выполнения лабораторной работы мной были усвоены основные навыки программирования в оболочке ОС UNIX, а также написания комплексных командных файлов с использованием логических управляющих конструкций и циклов.

## Контрольные вопросы.

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

Нет пробелов до ] и после [. 2. Как объединить (конкатенация) несколько строк в одну? Нужно использовать оператор + в сочетании со строковым типом данных. 3. Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash? Команда seq предназначена для вывода определённой последовательности целых чисел. 4. Какой результат даст вычисление выражения  $\$((10/3))$ ? Результат - 3, так как происходит целочисленное деление без остатка. 5. Укажите кратко основные отличия командной оболочки zsh от bash.

6. Проверьте, верен ли синтаксис данной конструкции  
`for ((a=1; a <= LIMIT; a++))`

Да, всё верно 7. Сравните язык bash с какими-либо языками программирования. Какие преимущества у bash по сравнению с ними? Какие недостатки?