

# Лабораторная работа №12

Программирование в командном процессоре ОС UNIX. асширенное  
программирование

---

Буллер Т. А.

11 мая 2024

Российский университет дружбы народов, Москва, Россия

# Информация

---

- Буллер Татьяна Александровна
- студент группы НБИбд-01-23
- Российский университет дружбы народов

# **Вводная часть**

---

- виртуальная машина Kali Linux
- текстовый редактор nano
- командная оболочка bash

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

- виртуальная машина Kali Linux
- текстовый редактор nano
- командная оболочка bash
- Процессор pandoc для входного формата Markdown
- Результирующие форматы
  - pdf
  - html
- Автоматизация процесса создания: Makefile

# Выполнение лабораторной работы

---



Необходимо написать командный файл, реализующий упрощённый механизм семафоров. В случае, если файл уже используется некоторым другим процессом, командный файл ожидает его освобождения 5 секунд и выдает соответствующее сообщение. Затем, если файл не освободился, цикл повторяется. Если же файл после ожидания стал свободен, то выводится сообщение о записи в файл и в сам файл записывается некоторая фраза.

```
1 #!/bin/bash
2 while test -f lockfile
3 do
4 sleep 5
5 echo "waiting"
6 done
7
8 touch lockfile
9 let c=10
10 while ((c-->0))
11 do
12 echo "writing"
13 echo "in file">>lockfile
14 sleep 7
15 done
16
17 rm lockfile
18
```

Проверим работу файла: откроем два окна терминала и в одном из них запустим файл в привелигированном режиме, а во втором - в фоновом, переведя вывод в первое окно. Для перевода вывода в окно графического терминала используем команду `> /dev/pts/number`, где `number` - номер графического интерфейса.

A terminal window with a dark background. The prompt is `(tabuller@jordi)-[~]`. The user enters the command `$ bash 1.sh > /dev/pts/1 &`. The terminal responds with `[1] 7629`.

```
(tabuller@jordi)-[~]  
$ bash 1.sh > /dev/pts/1 &  
[1] 7629
```

Рис. 2: Перевод вывода в другое окно

Видим, что запущенный в первом окне файл первое время производит запись без проблем. Потом, когда подключается второй процесс, один из файлов начинает выводить сообщения об ожидании, когда файл записи оказывается занят.

```
(tabuller@jordi)-[~]
```

```
$ bash 1.sh
```

```
writing
```

```
writing
```

```
waiting
```

```
waiting
```

```
writing
```

```
waiting
```

```
writing
```

```
waiting
```

```
waiting
```

```
writing
```

```
^C
```

```
(tabuller@jordi)-[~]
```

```
$ waiting
```

```
(tabuller@jordi)-[~]
```

```
$ cat lockfile
```

```
in file
```

```
in file
```

```
in file: quieter you beco
```

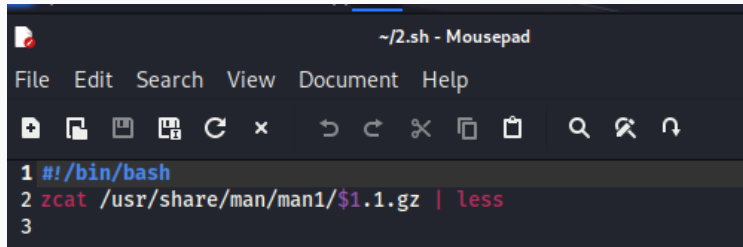
```
in file
```

```
in file
```

## Man с помощью командного файла

содержимое каталога `/usr/share/man/man1` - архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. На некоторых дистрибутивах Linux каждый архив можно открыть командой `less`, сразу же просмотрев содержимое справки, однако в случае Kali эта опция `less` по умолчанию отключена. Для того, чтобы открыть архив `.gz` на Kali использую `zcat` и перевожу вывод в `less`. В качестве аргумента передаем название программы, которое вставится в код скрипта.

# Man с помощью командного файла

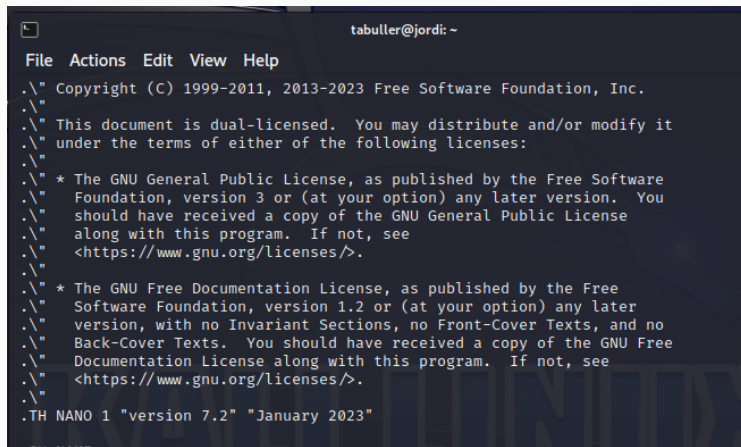


```
#!/bin/bash
zcat /usr/share/man/man1/$1.1.gz | less
```

Рис. 4: Скрипт man



# Man с помощью командного файла



```
tabuller@jordi: ~
File Actions Edit View Help
.\" Copyright (C) 1999-2011, 2013-2023 Free Software Foundation, Inc.
.\"
.\" This document is dual-licensed.  You may distribute and/or modify it
.\" under the terms of either of the following licenses:
.\"
.\" * The GNU General Public License, as published by the Free Software
.\"   Foundation, version 3 or (at your option) any later version.  You
.\"   should have received a copy of the GNU General Public License
.\"   along with this program.  If not, see
.\"   <https://www.gnu.org/licenses/>.
.\"
.\" * The GNU Free Documentation License, as published by the Free
.\"   Software Foundation, version 1.2 or (at your option) any later
.\"   version, with no Invariant Sections, no Front-Cover Texts, and no
.\"   Back-Cover Texts.  You should have received a copy of the GNU Free
.\"   Documentation License along with this program.  If not, see
.\"   <https://www.gnu.org/licenses/>.
.\"
.TH NANO 1 "version 7.2" "January 2023"
```

Рис. 5: Результат работы скрипта

Зададим переменную `alphabet`: массив, который заполним латинскими маленькими буквами (`a..z`). Далее зададим переменную-ограничитель и проведем цикл по ней: на каждой итерации в переменную `numb` запишем случайное число, которое ограничим установленным до этого лимитом. Заполним полученными элементами переменную-массив вывода и вызовем ее в конце программы:

# Случайная комбинация

```
1 #!/bin/bash
2 declare -a alphabet
3 alphabet={a..z}
4 let limit=26
5 let i=10
6 while ((i--=1))
7 do
8   numb=$RANDOM
9   let numb%=limit
10  output=$output${alphabet[$numb]}
11 done
12 echo $output
13
```

File Actions Edit View Help

(tabuller@jordi)-[~]  
\$ bash 3.sh  
danpaijsw

(tabuller@jordi)-[~]  
\$ bash 3.sh  
pdnauimpi

(tabuller@jordi)-[~]  
\$ bash 3.sh  
izdxsvdnn

(tabuller@jordi)-[~]  
\$ |

Рис. 6: Скрипт и результат выполнения

## **Выводы**

---

Изучены основы программирования в оболочке ОС UNIX/Linux. Написаны более сложные командные файлы с использованием логических управляющих конструкций и циклов.