

Отчет по лабораторной работе №11

**Программирование в командном процессоре ОС UNIX. Ветвления и
циклы**

Кочкарев “sakochkarev” Станислав

Содержание

1	Цель работы	3
2	Задание	4
3	Выполнение лабораторной работы	5
4	Выводы	11
5	Контрольные вопросы	12

1 Цель работы

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

2 Задание

1. Написать командный файл, реализующий упрощенный механизм семафоров.
2. Написать командный файл, аналог команды `map`.
3. Написать командный файл, генерирующий случайную строку.

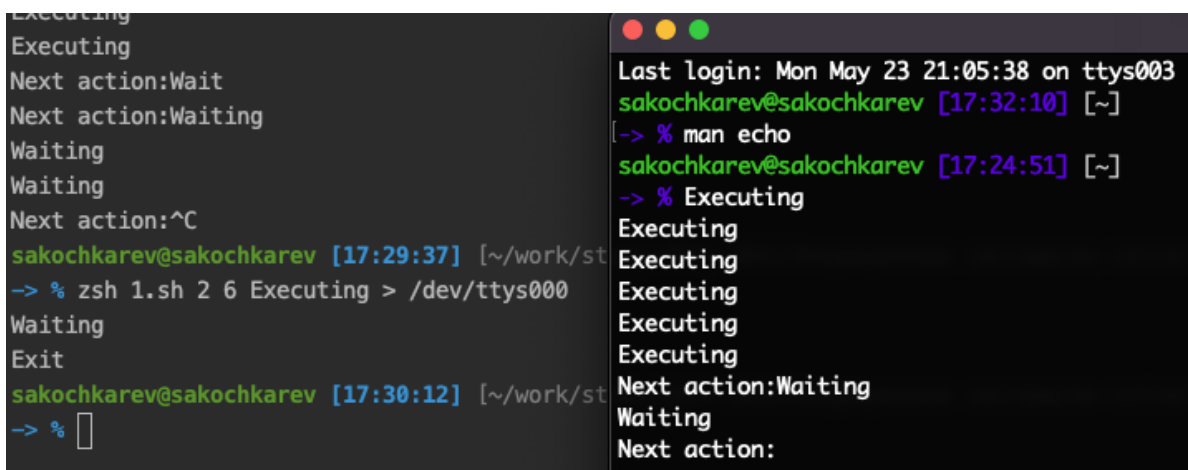
3 Выполнение лабораторной работы

Первым заданием было написание командного файла, реализующего упрощенный механизм семафоров.

Ниже приведен листинг итогового командного файла (рис. 3.6), а также результат его работы (рис. 3.7).

```
1 #!/bin/zsh
2
3 function waiting {
4     time1=$(date +%s)
5     time2=$(date +%s)
6     ((time_diff = time2 - time1))
7
8     while ((time_diff < $1)); do
9         echo "Waiting"
10        sleep 1
11        time2=$(date +%s)
12        ((time_diff = time2 - time1))
13    done
14 }
15
16 function execution {
17     time1=$(date +%s)
18     time2=$(date +%s)
19     ((time_diff = time2 - time1))
20
21     while ((time_diff < $2)); do
22
23         echo "Executing"
24         sleep 1
25         time2=$(date +%s)
26         ((time_diff = time2 - time1))
27     done
28 }
29
30 command=$3
31
32 while true; do
33     if [[ "$command" = "Exit" ]]; then
34         exit;
35     elif [[ "$command" = "Waiting" ]]; then
36         waiting "$@"
37     elif [[ "$command" = "Executing" ]]; then
38         execution "$@"
39     fi
40     echo -n "Next action:"
41     read -r command
42 done
```

Рис. 3.1: Листинг командного файла



```
Executing
Next action:Wait
Next action:Waiting
Waiting
Waiting
Next action:^C
sakochkarev@sakochkarev [17:29:37] [~/work/st
-> % zsh 1.sh 2 6 Executing > /dev/ttys000
Waiting
Exit
sakochkarev@sakochkarev [17:30:12] [~/work/st
-> % █

Last login: Mon May 23 21:05:38 on ttys003
sakochkarev@sakochkarev [17:32:10] [~]
-> % man echo
sakochkarev@sakochkarev [17:24:51] [~]
-> % Executing
Executing
Executing
Executing
Executing
Executing
Next action:Waiting
Waiting
Next action:
```

Рис. 3.2: Выполнение команды

Следующим заданием было написание командного файла, который альтернирует поведение команды `man`. Командный файл принимает один аргумент – название команды, мануал которой пользователь хочет просмотреть. Для работы командного файла была использована директория `/usr/share/man/man1`, в которой происходил поиска мануала приведенной команды. Далее мануал выводился с помощью команды `less`.

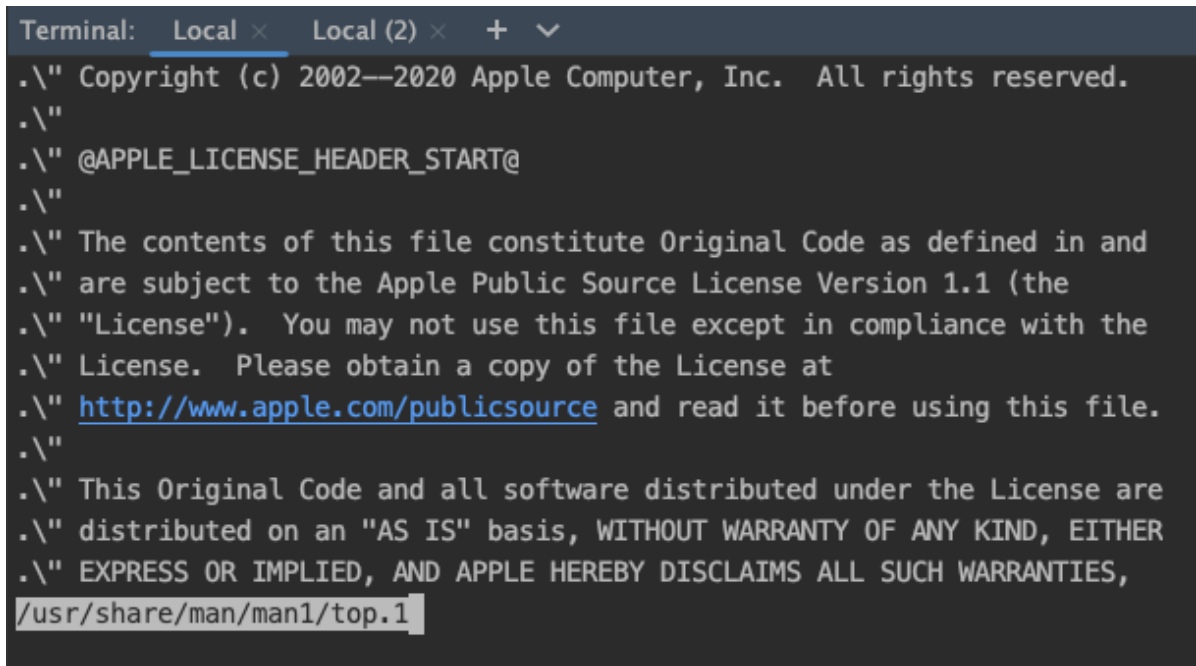
Ниже приведен листинг итогового командного файла (рис. 3.3), а также результат его работы (рис. 3.4, 3.5).

```
1 #!/bin/zsh
2
3 if [ $# -eq 0 ];
4 then
5     echo "No command name provided";
6     exit;
7 fi
8
9 DIR=/usr/share/man/man1
10 CMD_FILE="$1.1"
11 FULL_PATH="$DIR/$CMD_FILE"
12
13 if [[ -f "$FULL_PATH" ]];
14 then
15     less "$FULL_PATH";
16 else
17     echo "No manual for this command";
18 fi
```

Рис. 3.3: Листинг командного файла

```
sakochkarev@sakochkarev [20:12:45] [~/work/study/2021-2022/Операционные системы/os-intro/labs/lab12] [master *]
-> % zsh 2.sh
No command name provided
sakochkarev@sakochkarev [20:12:53] [~/work/study/2021-2022/Операционные системы/os-intro/labs/lab12] [master *]
-> % zsh 2.sh top
sakochkarev@sakochkarev [20:14:57] [~/work/study/2021-2022/Операционные системы/os-intro/labs/lab12] [master *]
```

Рис. 3.4: Выполнение команды

A screenshot of a macOS Terminal window. The title bar shows 'Terminal: Local x Local (2) x + v'. The terminal content displays the output of a command, which is the Apple Public Source License header. The text is as follows:

```
.\" Copyright (c) 2002—2020 Apple Computer, Inc. All rights reserved.
.\"
.\" @APPLE_LICENSE_HEADER_START@
.\"
.\" The contents of this file constitute Original Code as defined in and
.\" are subject to the Apple Public Source License Version 1.1 (the
.\" "License"). You may not use this file except in compliance with the
.\" License. Please obtain a copy of the License at
.\" http://www.apple.com/publicsource and read it before using this file.
.\"
.\" This Original Code and all software distributed under the License are
.\" distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, EITHER
.\" EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES,
/usr/share/man/man1/top.1
```

Рис. 3.5: Выполнение команды

Последним заданием было написание командного файла, который, используя встроенную переменную \$RANDOM выводит случайную последовательности латинских букв.

Ниже приведен листинг итогового командного файла (рис. 3.6), а также результат его работы (рис. 3.7).

```
1 #!/bin/zsh
2
3 COUNT=0
4 ENG_LETTERS=26
5 LENGTH=8
6 STRING=""
7
8 for ((i=1; i<LENGTH; i++)); do
9     POS=$((1 + RANDOM % ENG_LETTERS))
10    ((COUNT=0))
11    for letter in {a..z}; do
12        if [[ $COUNT -eq $POS ]]; then
13            STRING+=$letter;
14            break;
15        else
16            ((COUNT+=1));
17        fi
18    done
19 done
20
21 echo "$STRING"
```

Рис. 3.6: Листинг командного файла

```
sakochkarev@sakochkarev [20:31:16] [~/work/study/2021-2022/Операционные системы/os-intro/labs/lab12] [master *]
-> % zsh 3.sh
ezrdoiz
```

Рис. 3.7: Выполнение команды

4 Выводы

По выполнении лабораторной работы мы изучили основы программирования в оболочке ОС UNIX, а также научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

5 Контрольные вопросы

1. Синтаксическая ошибка заключается в отсутствии пробелов вокруг выражения в []. Правильная строка:

```
while [ $1 != "exit" ]
```

2. Можно использовать подстановку переменных в строку. Например:

```
string1="hello"  
string2="world"  
result="${string1} ${world}"
```

3. Подобный seq функционал можно реализовать используя формат (()) вместе с Си-подобным итерированием, например:

```
for ((i=0; i<10; i++)); do  
    echo $i;  
done
```

4. 3.

5.
 - Прокаченное авто-дополнение
 - Переход без cd
 - Поправление ошибок
 - Переходы по истории

6. Да, синтаксис верен.

7. Сравнивая `bash`, например, с высокоуровневым языком программирования Python, `bash` уступает во многих сферах и применениях, так как это не нацеленный на разработку ПО язык, однако в среде UNIX, особенно в командных файлах (скриптах), ему практически нет равных.