

# **Лабораторная работа №14**

**Дисциплина: Операционные системы**

Комягин Андрей Николаевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задания</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
3.1	Код 1 . . . . .	8
3.2	Код 2 . . . . .	9
3.3	Код 3 . . . . .	9
<b>4</b>	<b>Вывод</b>	<b>10</b>

# Список иллюстраций

3.1	код 1	. . . . .	8
3.2	код 2	. . . . .	9
3.3	код 3	. . . . .	9

## Список таблиц

# 1 Цель работы

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

## 2 Задания

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. 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`.
3. Используя встроенную переменную `$RANDOM`, напишите командный файл, генерирующий случайную последовательность букв латинско-

го алфавита. Учтите, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

## 3 Выполнение лабораторной работы

### 3.1 Код 1

```
1 #!/bin/bash
2
3 if [ $# -ne 2 ]; then
4     echo "Использование: $0 <t1> <t2>"
5     exit 1
6 fi
7
8 t1=$1
9
10 t2=$2
11
12 semaphore_file="semaphore.lock"
13
14 touch $semaphore_file
15
16 function access_resource {
17     while ! ln $semaphore_file $0.lock 2>/dev/null; do
18         echo "Ресурс занят, ожидание освобождения..."
19         sleep $t1
20     done
21
22     echo "Ресурс освобожден, начало использования на $t2
    секунд"
23     sleep $t2
24     echo "Ресурс освобожден, использование завершено"
25
26     rm $0.lock
27 }
28
29 access_resource
```

Рис. 3.1: код 1



## 3.2 Код 2

```
1 #!/bin/bash
2
3 if [ $# -ne 1 ]; then
4     echo "Использование: $0 <название_команды>"
5     exit 1
6 fi
7
8 command_name=$1
9
10 man_directory="/usr/share/man/man1"
11
12 if [ -f "$man_directory/$command_name.1.gz" ]; then
13     zcat "$man_directory/$command_name.1.gz" | less
14 else
15     echo "Справка для команды '$command_name' не найдена"
16 fi
```

Рис. 3.2: код 2

## 3.3 Код 3

```
1 #!/bin/bash
2
3 generate_random_letter() {
4     # Случайное число от 0 до 25
5     random_number=$((RANDOM % 26))
6
7     letter=$(printf \\$(printf '%03o' $((65 +
8 random_number))))
9     echo -n "$letter"
10 }
11
12 random_sequence=""
13 for ((i=0; i<10; i++)); do
14     random_sequence="$random_sequence$(
15 generate_random_letter)"
16 done
17 echo "Случайная последовательность букв латинского алфавита:
18 $random_sequence"
```

Рис. 3.3: код 3

## 4 Вывод

В ходе работы я изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.