

**Лабораторная работа № 12.**  
**Программирование в командном**  
**процессоре ОС UNIX.Расширенное**  
**программирование**

**Мигачев Максим**

# **Цель работы**

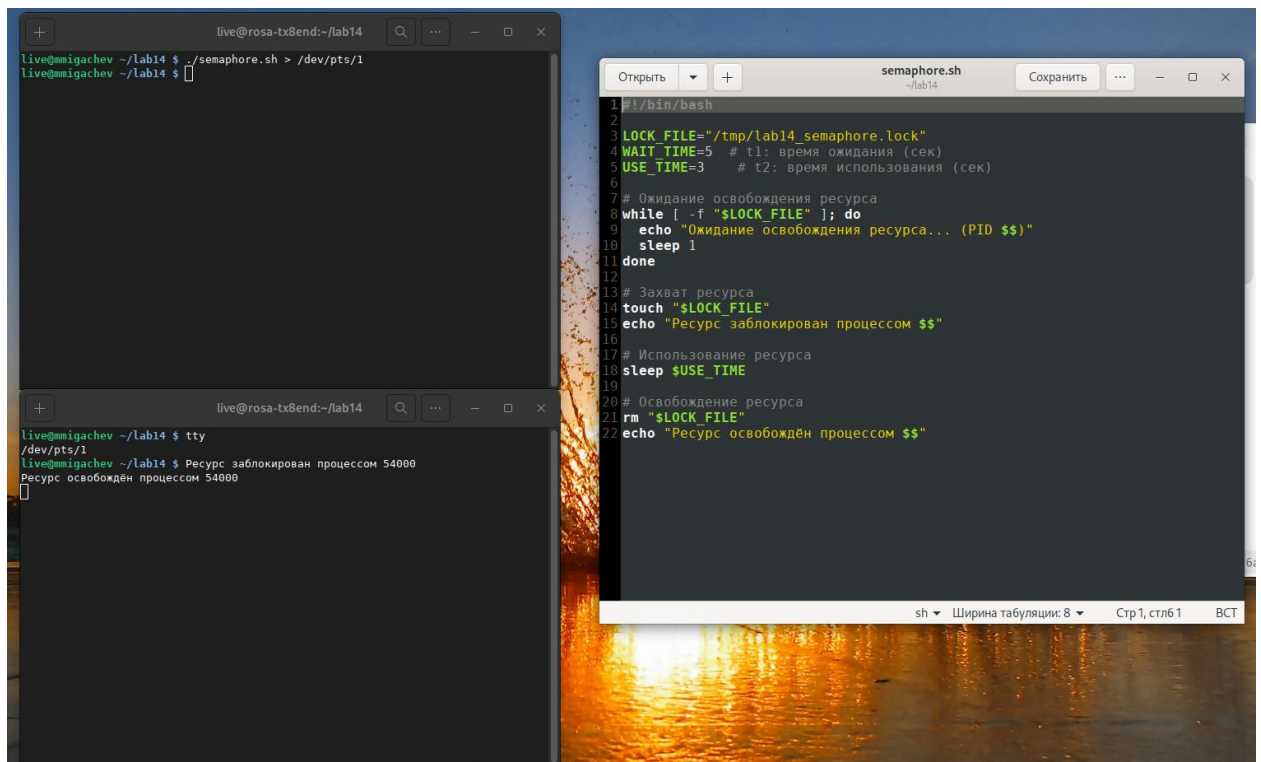
**Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.**

# Результат

## 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 имела возможность взаимодействия трёх и более процессов.



The image shows three terminal windows illustrating the execution of a semaphore script. The top-left window shows the script being executed in the background: `live@rosa-tx8end:~/lab14$ ./semaphore.sh > /dev/pts/1`. The bottom-left window shows the script being executed in the foreground: `live@rosa-tx8end:~/lab14$ ./semaphore.sh`, with output indicating resource acquisition and release. The right window shows the script's source code, which implements a semaphore using a lock file and sleep functions.

```
1#!/bin/bash
2
3LOCK_FILE="/tmp/lab14_semaphore.lock"
4WAIT_TIME=5 # t1: время ожидания (сек)
5USE_TIME=3 # t2: время использования (сек)
6
7# Ожидание освобождения ресурса
8while [ -f "$LOCK_FILE" ]; do
9    echo "Ожидание освобождения ресурса... (PID $$)"
10    sleep 1
11done
12
13# Захват ресурса
14touch "$LOCK_FILE"
15echo "Ресурс заблокирован процессом $$"
16
17# Использование ресурса
18sleep $USE_TIME
19
20# Освобождение ресурса
21rm "$LOCK_FILE"
22echo "Ресурс освобожден процессом $$"
```

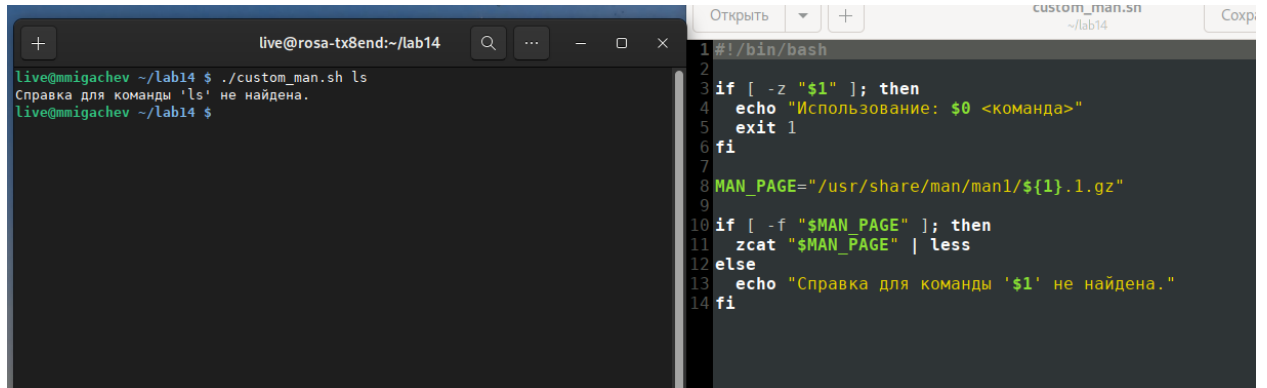
Рис 1

## 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 просмотреть 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`.



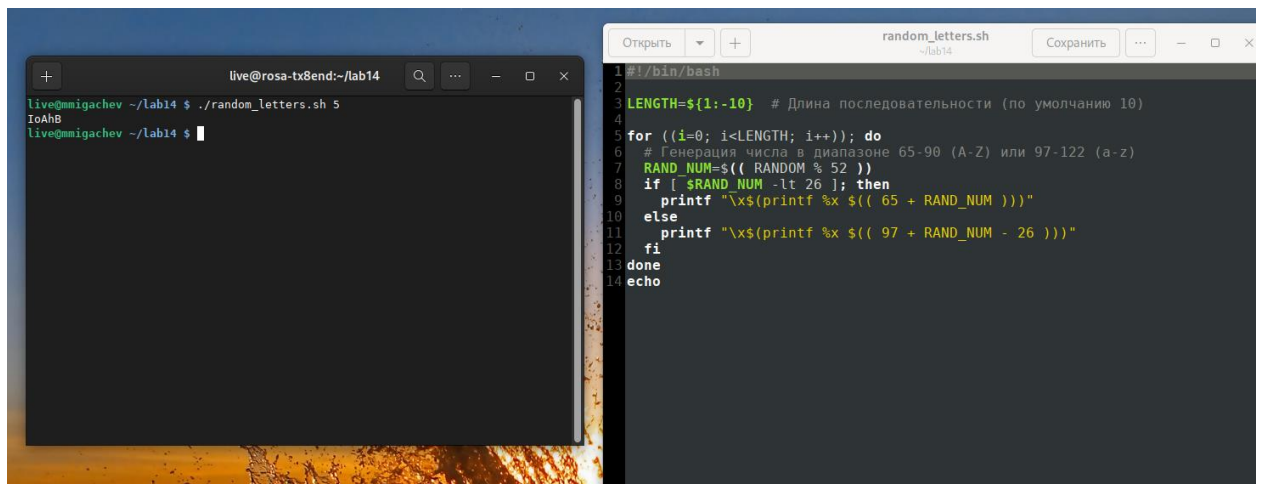
The screenshot shows a terminal window with the prompt `live@rosa-tx8end:~/lab14`. The user runs `./custom_man.sh ls`. The output is: `Справка для команды 'ls' не найдена.` The terminal also shows the source code of `custom_man.sh` in a separate window. The script is as follows:

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

**Рис 2**

### 3.

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.



The screenshot shows a terminal window with the prompt `live@rosa-tx8end:~/lab14`. The user runs `./random_letters.sh 5`. The output is: `IoAhB`. The terminal also shows the source code of `random_letters.sh` in a separate window. The script is as follows:

```
1#!/bin/bash
2
3LENGTH=${1:-10} # Длина последовательности (по умолчанию 10)
4
5for ((i=0; i<LENGTH; i++)); do
6    # Генерация числа в диапазоне 65-90 (A-Z) или 97-122 (a-z)
7    RAND_NUM=$((RANDOM % 52))
8    if [ $RAND_NUM -lt 26 ]; then
9        printf "\x$(printf %x $((65 + RAND_NUM)))"
10    else
11        printf "\x$(printf %x $((97 + RAND_NUM - 26)))"
12    fi
13done
14echo
```

**Рис 3**

## **Вывод**

**Освоены управляющие конструкции,  
работа с файлами и переменными в  
bash.**

### **Ответы на контрольные вопросы:**

- 1. Ошибка в строке: `while [$1 != "exit"]` → нужно пробелы: `while [ "$1" != "exit" ]`.**
- 2. Конкатенация строк: `str3="${str1}${str2}"`.**
- 3. Утилита `seq`: Альтернатива — `for i in {1..10}` или цикл на `while`.**