

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Бровкин Александр НБИбд-01-21¹

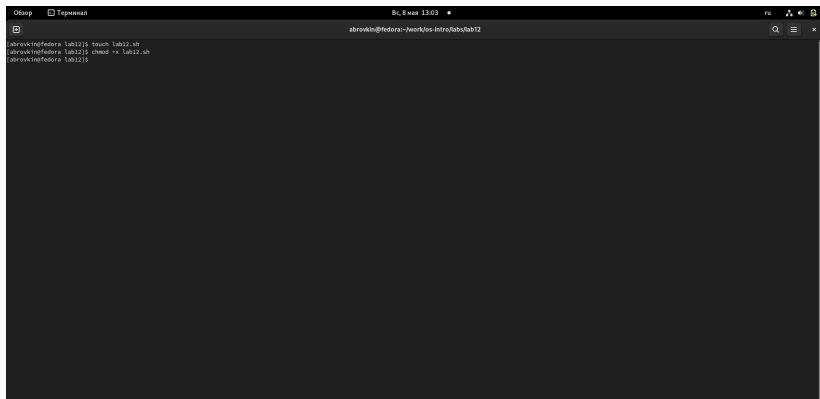
8 мая, 2022, Москва, Россия

¹Российский Университет Дружбы Народов

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

Ход работы:

1. Написал 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). Запустил 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.



The image shows a terminal window with a dark background. The title bar at the top includes the text 'Обзор Терминал' on the left, 'БС, 6 мая 13:03' in the center, and system icons on the right. The terminal content shows a user 'abrovin@fedora' in the directory '~/work/bs-intra/labs/lab12' performing three commands: 'touch lab12.sh', 'chmod +x lab12.sh', and a prompt for the next command.

```
abrovin@fedora:~/work/bs-intra/labs/lab12
labrovin@fedora lab12$ touch lab12.sh
labrovin@fedora lab12$ chmod +x lab12.sh
labrovin@fedora lab12$
```

Figure 1: Создаю файл

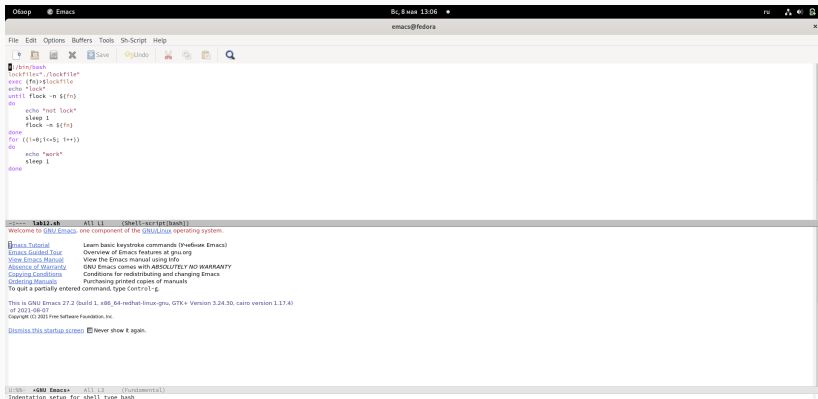
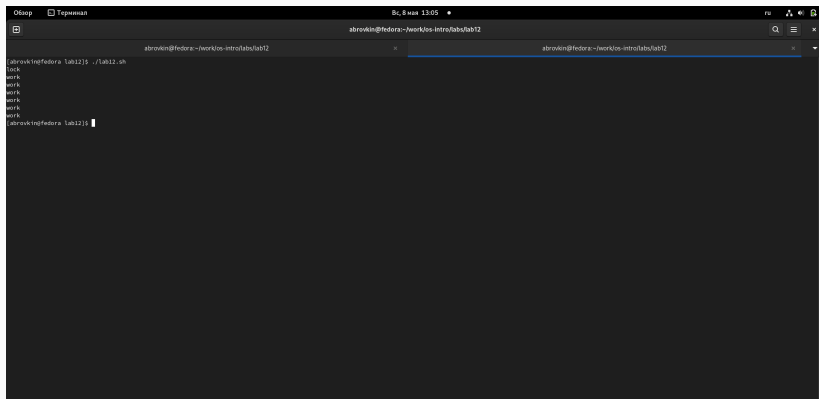


Figure 2: Пишу скрипт

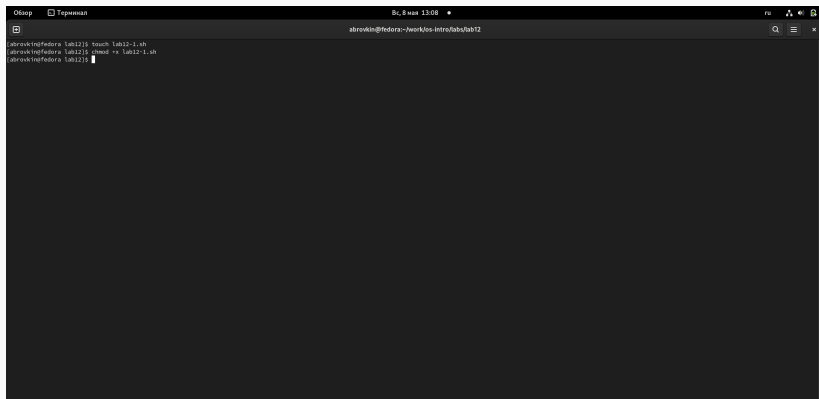


The image shows a terminal window titled "Терминал" (Terminal) with a dark background. The window has a title bar with "Обзор" (Overview) and "Терминал" (Terminal) buttons, and a status bar showing "Вт, 6 мая 13:05" (Tue, 6 May 13:05). The terminal content shows a user prompt "abrovkin@fedora: ~/work/os-intro/labs/lab12" followed by a command prompt "[abrovkin@fedora lab12]\$./lab12.sh". The script output consists of five lines of "work" text. The prompt then changes to "[abrovkin@fedora lab12]\$".

```
abrovkin@fedora: ~/work/os-intro/labs/lab12
[abrovkin@fedora lab12]$ ./lab12.sh
work
work
work
work
work
[abrovkin@fedora lab12]$
```

Figure 3: Программа

2. Реализовал команду `man` с помощью командного файла. Изучил 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`.



The image shows a terminal window titled "Терминал" (Terminal) with a dark background. The window's title bar includes "Обзор" (Overview), "Терминал", the system clock "Вт, 6 мая 13:08", and standard window controls. The terminal content shows a user named "abrovin@fedora" in the directory "~/.work/ps-intra/labs/lab12". The user has executed three commands: "touch lab12-1.sh", "chmod +x lab12-1.sh", and is currently at the prompt for a fourth command. The first two commands are visible on separate lines, and the third is on a line that is partially cut off at the bottom of the terminal area.

```
abrovin@fedora:~/.work/ps-intra/labs/lab12
labrovin@fedora lab12$ touch lab12-1.sh
labrovin@fedora lab12$ chmod +x lab12-1.sh
labrovin@fedora lab12$
```

Figure 4: Создаю файл

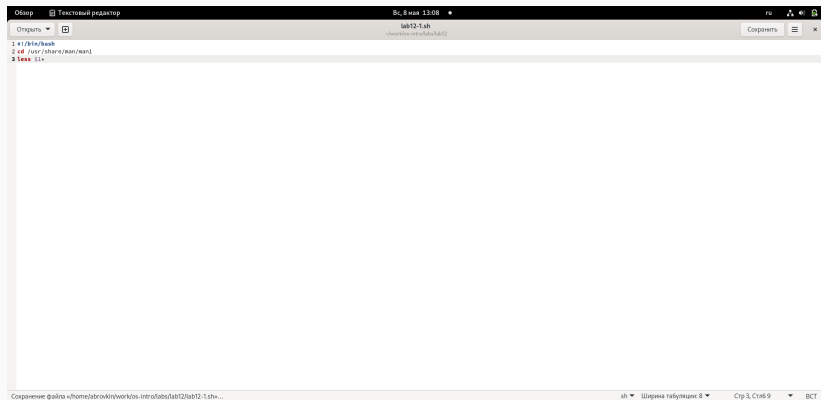
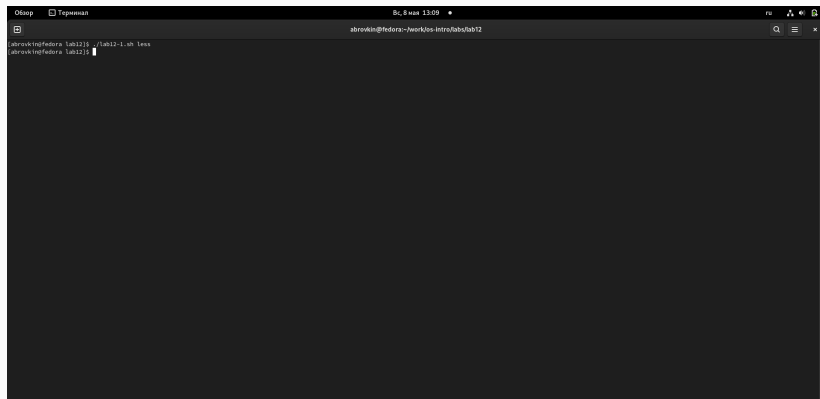


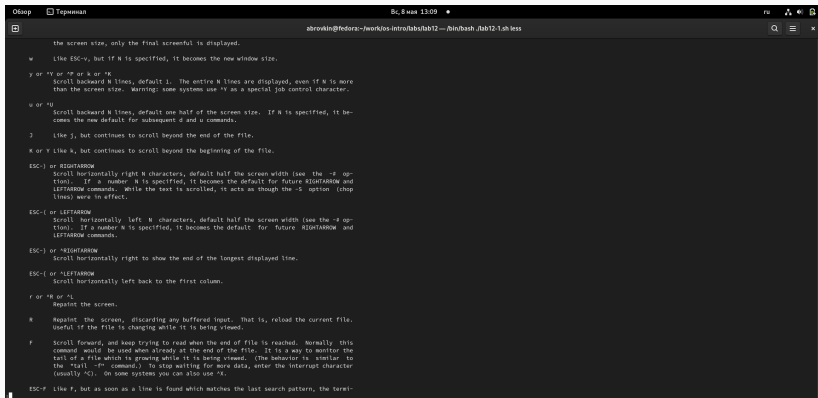
Figure 5: Написал скрипт



The image shows a terminal window with a dark background. The title bar at the top indicates the window is titled "Терминал" (Terminal) and shows the system time as "Вт, 6 мая 13:09". The terminal content shows a user prompt "abroavkin@fedora: ~/work/ps-intra/labs/lab12" followed by the command `./lab12-1.sh less`. The output of the command is a large, empty space, suggesting the command executed successfully but produced no visible output.

```
abroavkin@fedora: ~/work/ps-intra/labs/lab12
abroavkin@fedora: lab12$ ./lab12-1.sh less
```

Figure 6: Программа



The screenshot shows a terminal window with the title bar "БС, 6 мая 13:09". The terminal content displays the help text for the `less` command, starting with "the screen size, only the final screenful is displayed." and listing various keyboard shortcuts like `w`, `y`, `u`, `j`, `k`, `ESC-)`, `ESC-(`, `ESC-)`, `ESC-(`, `r`, `R`, `F`, and `ESC-F`.

```
the screen size, only the final screenful is displayed.

w Like ESC-v, but if N is specified, it becomes the new window size.

y or ^Y or ^P or k or ^K
  Scroll backward N lines, default 1. The entire N lines are displayed, even if N is more
  than the screen size. Warning: some systems use ^Y as a special job control character.

u or ^U
  Scroll backward N lines, default one half of the screen size. If N is specified, it be-
  comes the new default for subsequent d and u commands.

j Like j, but continues to scroll beyond the end of the file.

k or ^K like k, but continues to scroll beyond the beginning of the file.

ESC-) or RIGHTARROW
  Scroll horizontally right N characters, default half the screen width (see the -P op-
  tion). If a number N is specified, it becomes the default for future RIGHTARROW and
  LEFTARROW commands. While the text is scrolled, it acts as though the -S option (chop
  lines) were in effect.

ESC-( or LEFTARROW
  Scroll horizontally left N characters, default half the screen width (see the -P op-
  tion). If a number N is specified, it becomes the default for future RIGHTARROW and
  LEFTARROW commands.

ESC-) or ^RIGHTARROW
  Scroll horizontally right to show the end of the longest displayed line.

ESC-( or ^LEFTARROW
  Scroll horizontally left back to the first column.

r or ^R or ^L
  Repaint the screen.

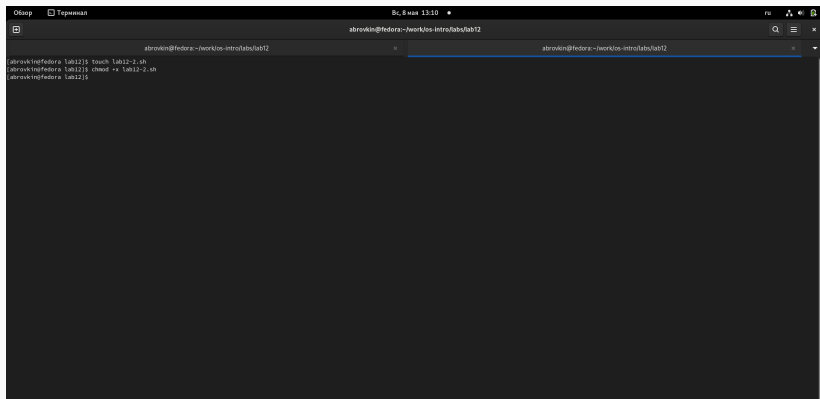
R Repaint the screen, discarding any buffered input. That is, reload the current file.
  Useful if the file is changing while it is being viewed.

F Scroll forward, and keep trying to read when the end of file is reached. Normally this
  command would be used when already at the end of the file. It is a way to monitor the
  tail of a file which is growing while it is being viewed. (The behavior is similar to
  the "tail -f" command.) To stop waiting for more data, enter the interrupt character
  (usually ^C). On some systems you can also use ^K.

ESC-F Like F, but as soon as a line is found which matches the last search pattern, the term-
```

Figure 8: Программа

3. Используя встроенную переменную \$RANDOM, написал командный файл, генерирующий случайную последовательность букв латинского алфавита. Учел, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

A terminal window with a dark background. The title bar shows 'Обзор' (Overview) and 'Терминал' (Terminal) on the left, and '8c, 6 мая 13:10' on the right. The terminal content shows a user 'abrovin' at a prompt 'abrovin@fedora: ~/work/os-intro/labs/lab12'. The user enters three commands: 'touch lab12-1.sh', 'chmod +x lab12-1.sh', and 'touch lab12-2.sh'. The prompt returns after each command.

```
abrovin@fedora: ~/work/os-intro/labs/lab12
[abrovin@fedora lab12]$ touch lab12-1.sh
[abrovin@fedora lab12]$ chmod +x lab12-1.sh
[abrovin@fedora lab12]$
```

Figure 9: Создаю текстовый файл

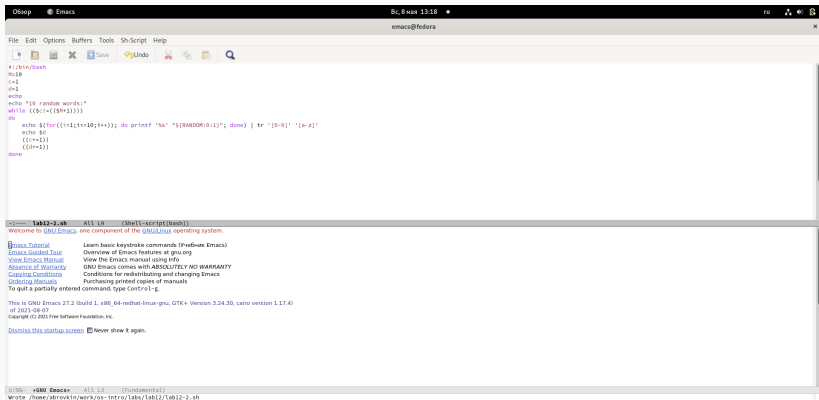
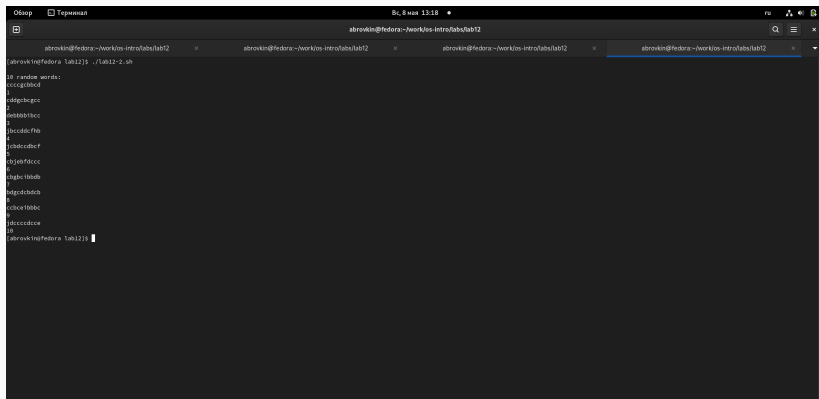


Figure 10: Пишу скрипт



The image shows a terminal window titled "Терминал" (Terminal) with the user "abrovkin" and host "fedora". The terminal displays the output of a script named "lab12-2.sh". The script generates 10 random words, each consisting of 10 lowercase letters. The words are listed as follows:

```
10 random words:
1 cccgcghcd
2 edgcbqgc
3 eebbbrlacc
4 lbccdcflh
5 jchdcufef
6 bjebfdccc
7 bgbclhdbb
8 fdgdcdbcb
9 cchcafhbc
10 dccccdcca
```

The terminal prompt is "abrovkin@fedora lab12\$".

Figure 11: Программа

Вывод:

Изучил основы программирования в оболочке ОС UNIX, научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.