

Лабораторная работа №12

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Ежова А. М.

26 апреля 2023

Российский университет дружбы народов, Москва, Россия

Факультет физико-математических и естественных наук

Информация

- Ежова Алиса Михайловна
- студентка НБИбд-01-22
- Российский университет дружбы народов
- <https://github.com/AlisaEzhova>

Вводная часть

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

- 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.

- 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`.

- 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.

- Процессор **pandoc** для входного формата Markdown
- Результирующие форматы
 - pdf
 - html
- Автоматизация процесса создания: **Makefile**

Создание презентации

- Pandoc: преобразователь текстовых файлов
- Сайт: <https://pandoc.org/>
- Репозиторий: <https://github.com/jgm/pandoc>

- Использование LaTeX
- Пакет для презентации: beamer
- Тема оформления: **metropolis**

```
slide_level: 2  
aspectratio: 169  
section-titles: true  
theme: metropolis
```

- Используется фреймворк `reveal.js`
- Используется тема `beige`

- Тема задаётся в файле `Makefile`

```
REVEALJS_THEME = beige
```

Результаты

- Полученный **pdf**-файл можно демонстрировать в любой программе просмотра **pdf**
- Полученный **html**-файл содержит в себе все ресурсы: изображения, **css**, скрипты

- 1) Написала 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, использовала его в течение некоторого времени $t_2 < t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустила 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. Доработала программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

```
amezhova@dk6n55 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ touch lab12.sh
amezhova@dk6n55 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ chmod +x lab12.sh
```

Рис. 1: Создание файла



The image shows a code editor window titled "lab12.sh" with a file path of "~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12". The editor contains a shell script with 15 lines of code. The code implements a locking mechanism using flock and a loop that runs 6 times, printing "work" and sleeping for 1 second. The script starts with a shebang line and defines a lockfile path.

```
1 #!/bin/bash
2 lockfile="./lockfile"
3 exec {fn}>$lockfile
4 echo "lock"
5 until flock -n ${fn}
6 do
7     echo "not lock"
8     sleep 1
9     flock -n ${fn}
10 done
11 for ((i=0;i<=5; i++))
12 do
13     echo "work"
14     sleep 1
15 done
```

Рис. 2: Написание кода

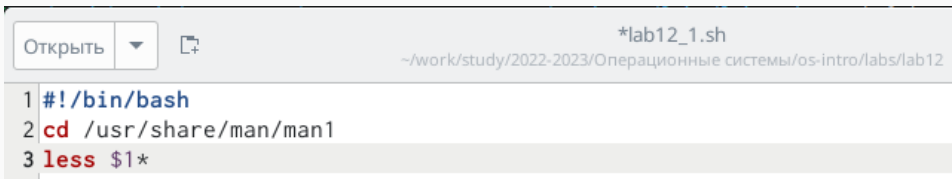
```
amezhova@dk6n55 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12.sh
lock
work
work
work
work
work
work
```

Рис. 3: Результат

2) Реализовала команду `man` с помощью командного файла. Изучила 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`.

```
amezhova@dk6n55 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ touch lab12_1.sh
amezhova@dk6n55 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ chmod +x lab12_1.sh
```

Рис. 4: Создание файла



The screenshot shows a code editor window with a title bar containing a button labeled "Открыть" (Open), a dropdown arrow, and a file icon. The file name is "*lab12_1.sh" and the path is "~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12". The editor contains three lines of code:

```
1 #!/bin/bash
2 cd /usr/share/man/man1
3 less $1*
```

Рис. 5: Написание кода

```
amezhova@dk6n55 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12_1.sh less
```

Рис. 6: Результат

```
LESS(1)                                     General Commands Manual                                     LESS(1)

NAME
    less - opposite of more

SYNOPSIS
    less -?
    less --help
    less -V
    less --version
    less [-[+]aABcCdeEfFgGiIJKLmMnNqQrRsSuUVwWX~]
        [-b space] [-h lines] [-j line] [-k keyfile]
        [-{oO} logfile] [-p pattern] [-P prompt] [-t tag]
        [-T tagsfile] [-x tab,...] [-y lines] [-[z] lines]
        [-# shift] [+[[+]cmd] [--] [filename]...
    (See the OPTIONS section for alternate option syntax with long option names.)
```

Рис. 7: Результат

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

```
amezhova@dk6n55 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ touch lab12_2.sh
amezhova@dk6n55 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ chmod +x lab12_2.sh
```

Рис. 8: Создание файла


```
File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help
#!/bin/bash
M=8
c=1
d=1
echo
echo "8 random words"
while (($c≠(($M+1))))
do
    echo $(for((i=1;i<8;i++)); do printf '$s' "${RANDOM:0:1}"; done) | tr '[0-9]' '[a-z]'
    echo $d
    ((c+=1))
    ((d+=1))
done
```

Рис. 9: Написание кода

```
amezhova@dk6n55 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12_2.sh
8 random words
$$$$$$$$$$$$
1
$$$$$$$$$$$$
2
$$$$$$$$$$$$
```

- В ходе выполнения Лабораторной работы №12, я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.