

Лабораторная работа №12

Легиньких Галина - группа НФИбд-02-21

20.05.2022

Программирование в командном процессоре ОС UNIX. Расширенное программирование

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

1. Написала командный файл, реализующий упрощённый механизм семафоров. Командный файл в течение некоторого времени t_1 дожидается освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует 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.(рис. 1)(рис. 2)

```
lackfile="./lacking.file"

exec {fn}>"$lackfile"
if test -f "$lackfile"
then
    while [ 1!=0 ]
    do
        if flock -n ${fn}
        then
            echo "file was locked"
            sleep 4
            echo "unlocking"
            flock -u ${fn}

        else
            echo "file already locked"
            sleep 3
            fi
    done
fi
~
~
~
```

Figure 1: Скрипт 1

```
galeginjkikh@dk6n39 ~ $ vi lab12_1
galeginjkikh@dk6n39 ~ $ chmod 777 lab12_1
galeginjkikh@dk6n39 ~ $ ./lab12_1
file was locked
unlocking
file already locked
file already locked
file already locked
file already locked
file already locked
file already locked
file already locked
file already locked
file already locked
file already locked
ccfile already locked

file already locked
file already locked
file already locked
a
file already locked
file already locked
cfile already locked
```

Figure 2: Работа скрипт 1

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`. (рис. 3)(рис. 4)

```
command=""

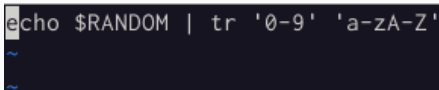
while getopts :n: opt
do
case $opt in
n)command="$OPTARG";;
esac
done

echo "$command"

~
~
~
~
~
~
~
~
```

Figure 3: Скрипт 2

3. Используя встроенную переменную \$RANDOM, написала командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтя, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.(рис. 5)

A screenshot of a terminal window with a dark background. The command 'echo \$RANDOM | tr '0-9' 'a-zA-Z'' is entered at the prompt. Below the command, there are two lines, each starting with a blue tilde '~' character, representing the output of the command.

```
echo $RANDOM | tr '0-9' 'a-zA-Z'
~
~
```

Figure 5: Скрипт 3

Для наглядности запустила скрипт несколько раз, чтобы увидеть, что генерируются разные случайные пароли.(рис. 6)

```
galeginjkikh@dk6n39 ~ $ touch lab12_3
galeginjkikh@dk6n39 ~ $ vi lab12_3
galeginjkikh@dk6n39 ~ $ chmod 777 lab12_3
galeginjkikh@dk6n39 ~ $ ./lab12_3
ecja
galeginjkikh@dk6n39 ~ $ ./lab12_3
bjggj
galeginjkikh@dk6n39 ~ $ ./lab12_3
bjcif
galeginjkikh@dk6n39 ~ $
```

Figure 6: Работа скрипт 3

В ходе лабораторной работы я изучила основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.