

lab12

26 May, 2022 Moscow, Russia

Цель работы

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

Задача

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 `$RANDOM`, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что `$RANDOM` выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

Выполнение лабораторной работы

Задание 1

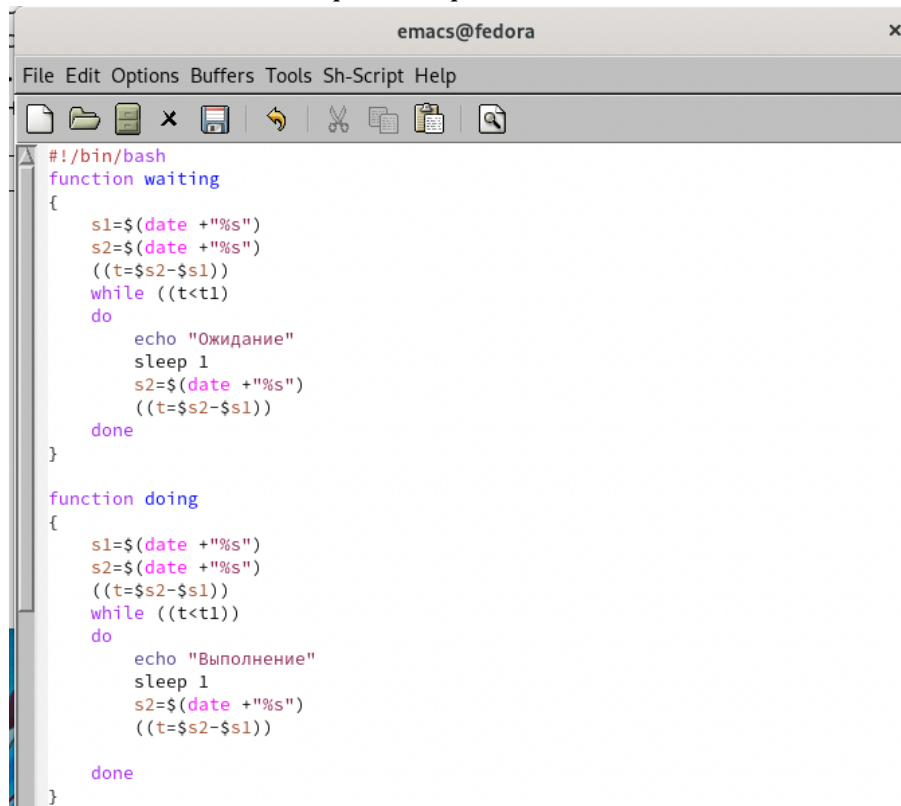
1. Я создала файла `prog1.sh`

```
epmitichkina@fedora:~  
[epmitichkina@fedora ~]$ touch prog1.sh  
[epmitichkina@fedora ~]$ emacs &  
[1] 2374
```

{#fig:001

width=70%}

2. Написала код в редакторе emacs



```
emacs@fedora  
File Edit Options Buffers Tools Sh-Script Help  
#!/bin/bash  
function waiting  
{  
    s1=$(date +%s)  
    s2=$(date +%s)  
    ((t=s2-s1))  
    while ((t<t1))  
    do  
        echo "Ожидание"  
        sleep 1  
        s2=$(date +%s)  
        ((t=s2-s1))  
    done  
}  
  
function doing  
{  
    s1=$(date +%s)  
    s2=$(date +%s)  
    ((t=s2-s1))  
    while ((t<t1))  
    do  
        echo "Выполнение"  
        sleep 1  
        s2=$(date +%s)  
        ((t=s2-s1))  
    done  
}
```

prog1.sh

```

done
}
t1=$1
t2=$2
command=$3
while true
do
    if [ "$command" == "Выход" ]
    then
        echo "Выход"
        exit 0
    fi
    if [ "$command" == "Ожидание" ]
    then waiting
    fi
    if [ "$command" == "Выполнение" ]
    then doing
    fi
    echo "Следующее действие:"
    read command
done

```

prog1.sh

3. Предоставила право на выполнение и проверка файла

```

[epmitichkina@fedora ~]$ chmod +x prog1.sh
[epmitichkina@fedora ~]$ ./prog1.sh 3 5 Ожидание > /dev/tty1 &
[1] 2664
bash: /dev/tty1: Отказано в доступе
[1]+  Выход 1          ./prog1.sh 3 5 Ожидание > /dev/tty1
[epmitichkina@fedora ~]$ ./prog1.sh 3 5 Ожидание > /dev/tty2 &
[1] 2669
[epmitichkina@fedora ~]$ ./prog1.sh 3 5 Выполнение > /dev/tty2 &
[2] 2687

[1]+  Остановлен      ./prog1.sh 3 5 Ожидание > /dev/tty2
[epmitichkina@fedora ~]$

```

Предоставление прав и проверка

Задание 2

1. Я проверила содержимое /usr/share/man/man1

```
epmitichkina@fedora:/usr/share/man/man1
[epmitichkina@fedora man1]$ cd
[epmitichkina@fedora ~]$ cd /usr/share/man/man1
[epmitichkina@fedora man1]$ ls
.:.1.gz
' [.1.gz'
ab.1.gz
abrt.1.gz
abrt-action-analyze-backtrace.1.gz
abrt-action-analyze-c.1.gz
abrt-action-analyze-ccpp-local.1.gz
abrt-action-analyze-core.1.gz
abrt-action-analyze-java.1.gz
abrt-action-analyze-oops.1.gz
abrt-action-analyze-python.1.gz
abrt-action-analyze-vmcore.1.gz
abrt-action-analyze-vulnerability.1.gz
abrt-action-analyze-xorg.1.gz
abrt-action-check-oops-for-hw-error.1.gz
abrt-action-find-bodhi-update.1.gz
abrt-action-generate-backtrace.1.gz
abrt-action-generate-core-backtrace.1.gz
abrt-action-install-debuginfo.1.gz
abrt-action-list-dsos.1.gz
abrt-action-notify.1.gz
abrt-action-perform-ccpp-analysis.1.gz
abrt-action-save-package-data.1.gz
abrt-action-trim-files.1.gz
abrt-applet.1.gz
abrt-auto-reporting.1.gz
abrt-bodhi.1.gz
abrt-cli.1.gz
```

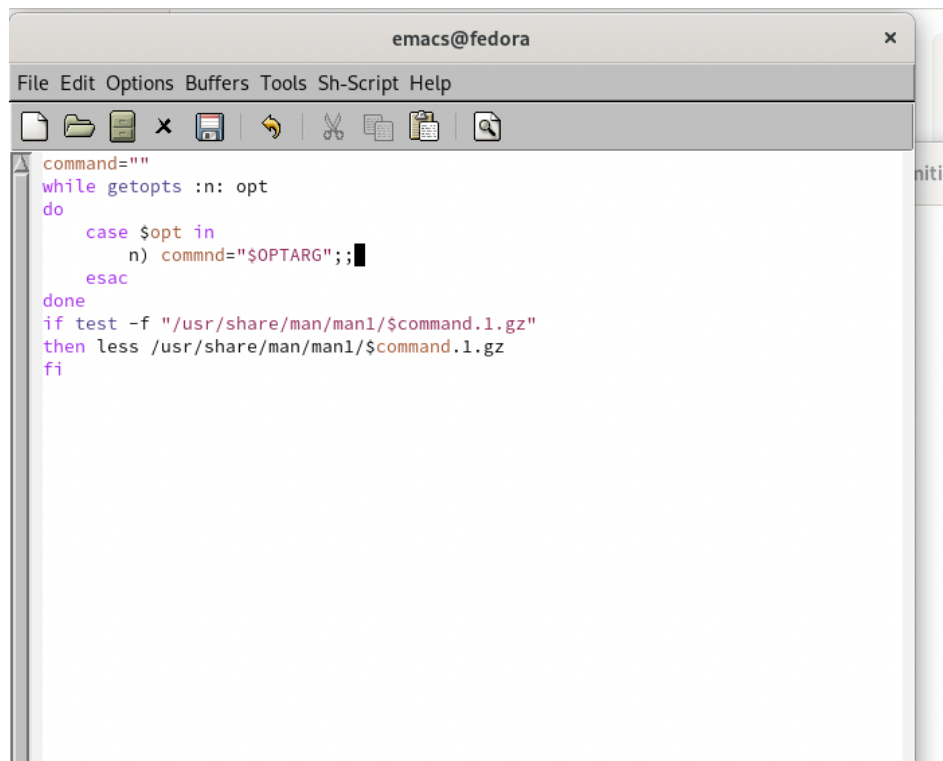
содержимое /usr/share/man/man1

2. Я создала файл prog2.sh

```
[epmitichkina@fedora ~]$ touch prog2.sh
[epmitichkina@fedora ~]$ emacs &
[2] 2876
[epmitichkina@fedora ~]$ touch prog2.sh
[1]-  Завершён      emacs
[2]+  Завершён      emacs
[epmitichkina@fedora ~]$ emacs &
[1] 2914
[epmitichkina@fedora ~]$ chmod +x prog2.sh
[epmitichkina@fedora ~]$ ./prog2.sh -n mc
[epmitichkina@fedora ~]$ ./prog2.sh -n rm
```

создание prog2.sh

2. Написала код в редакторе emacs



```
command=""
while getopts :n: opt
do
    case $opt in
        n) commnd="$OPTARG";;
    esac
done
if test -f "/usr/share/man/man1/$command.1.gz"
then less /usr/share/man/man1/$command.1.gz
fi
```

prog2.sh

3. Предоставила право на выполнение и проверка файла

```
[epmitichkina@fedora ~]$ chmod +x prog2.sh
[epmitichkina@fedora ~]$ ./prog2.sh -n mc
[epmitichkina@fedora ~]$ ./prog2.sh -n rm
[epmitichkina@fedora ~]$ ./prog2.sh -n vo
```

Предоставление прав и проверка

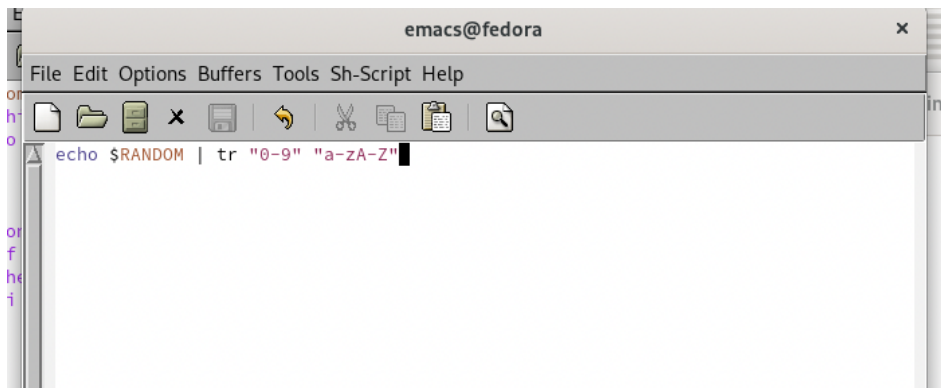
Задание 3

1. Я создала файл prog3.sh

```
[epmitichkina@fedora ~]$ ./prog2.sh -n shshgs
[epmitichkina@fedora ~]$ touch prog3.sh
[epmitichkina@fedora ~]$ emacs &
[2] 2987
```

создание prog3.sh

2. Написала код в редакторе emacs



prog3.sh

3. Предоставила право на выполнение и проверка файла

```
[epmitichkina@fedora ~]$ ./prog3.sh
bcbgi
[epmitichkina@fedora ~]$ ./prog3.sh
ccdig
[epmitichkina@fedora ~]$ ./prog3.sh
cgbbe
[epmitichkina@fedora ~]$
```

Предоставление прав и проверка

Выводы

В результате работы изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.