

"Отчёт по лабораторной работе №12"

"Дисциплина: Операционные системы"

"Горпинич Елена Михайловна"

Цель работы

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

Выполнение лабораторной работы

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, использовать его в течение некоторого времени $t_2 < t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Для данной задачи я создала файл и написала соответствующий скрипт.

```
emgorpinich@emgorpinich:~$ touch lab12_1.sh
emgorpinich@emgorpinich:~$ emacs &
[1] 2749
```

```
#!/bin/bash
t1=$1
t2=$2
s1=$(date +%s)
s2=$(date +%s)
((t=$s2-$s1))
while ((t < t1))
do
    echo "Ожидание"
    sleep 1
    s2=$(date +%s)
    ((t=$s2-$s1))
done
s1=$(date +%s)
s2=$(date +%s)
((t=$s2-$s1))
while ((t < t2))
do
    echo "Выполнение"
    sleep 1
    s2=$(date +%s)
    ((t=$s2-$s1))
done
```

Далее я проверила работу написанного скрипта, предварительно добавив право на исполнение файла. Скрипт работает корректно.

```
emgorpinich@emgorpinich:~$ chmod +x lab12_1.sh
emgorpinich@emgorpinich:~$ ./lab12_1.sh 3 6
Ожидание
Ожидание
Ожидание
Выполнение
Выполнение
Выполнение
Выполнение
Выполнение
Выполнение
```

После этого я изменила скрипт так, чтобы его можно было выполнять в нескольких терминалах и проверила его работу

```
#!/bin/bash
function ogidania
{
    s1=$(date +%s")
    s2=$(date +%s")
    ((t=s2-s1))
    while ((t < t1))
    do
        echo "Ожидание"
        sleep 1
        s2=$(date +%s")
        ((t=s2-s1))
    done
}
function vipolnenie
{
    s1=$(date +%s")
    s2=$(date +%s")
    ((t=s2-s1))
    while ((t < t2))
    do
        echo "Выполнение"
        sleep 1
        s2=$(date +%s")
        ((t=s2-s1))
    done
}
```

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


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`

```
emgorpinich@emgorpinich:~$ cd /usr/share/man/man1
emgorpinich@emgorpinich:/usr/share/man/man1$ ls
' [.1.gz'
7z.1.gz
7za.1.gz
7zr.1.gz
aa-enabled.1.gz
aa-exec.1.gz
aarch64-linux-gnu-addr2line.1.gz
aarch64-linux-gnu-ar.1.gz
aarch64-linux-gnu-as.1.gz
```

Для данной задачи я создала файл и написала соответствующий скрипт

```
emgorpinich@emgorpinich:~$ touch man.sh
emgorpinich@emgorpinich:~$ emacs &
[1] 3056
```

```
#!/bin/bash
c=$1
if [ -f /usr/share/man/man1/$c.1.gz ]
then
    gunzip -c /usr/share/man/man1/$1.1.gz | less
else
    echo "Справки по данной команде нет"
fi
```

Далее я проверила работу написанного скрипта , предварительно добавив право на исполнение файла. Скрипт работает корректно.

```
emgorpinich@emgorpinich:~$ chmod +x man.sh  
emgorpinich@emgorpinich:~$ ./man.sh ls  
emgorpinich@emgorpinich:~$ ./man.sh zip
```

```
emgorpinich@emgorpinich:~$ ./man.sh cd  
Справки по данной команде нет
```

```
.\" =====  
.\" Copyright (c) 1990-2008 Info-ZIP.  All rights reserved.  
.\"  
.\" See the accompanying file LICENSE, version 2007-Mar-4 or later  
.\" (the contents of which are also included in zip.h) for terms of use.  
.\" If, for some reason, all these files are missing, the Info-ZIP license  
.\" also may be found at:  ftp://ftp.info-zip.org/pub/infozip/license.html  
.\" =====  
.\"  
.\" zip.1 by Mark Adler, Jean-loup Gailly and  R. P. C. Rodgers  
.\" updated by E. Gordon for Zip 3.0 (8 May 2005, 24 December 2006,  
.\" 4 February 2007, 27 May 2007, 4 June 2007 by EG; 12 June 2007 by CS;  
.\" 30 August 2007, 27 April 2008, 25 May 2008, 27 May 2008 by EG,  
.\" 7 June 2008 by SMS and EG; 12 June 2008 by EG)  
.\"  
.TH ZIP 1 "16 June 2008 (v3.0)" Info-ZIP  
.SH NAME  
zip \- package and compress (archive) files  
.SH SYNOPSIS  
.B zip  
.RB [\- aABcdDeEfFghjklLmoqrRSTuvVwXyz!@$ ]  
[\-\-longoption ...]  
.RB [\- b " path"]
```

3. Используя встроенную переменную \$RANDOM, написала командный файл, генерирующий случайную последовательность букв латинского алфавита. Для данной задачи я создала файл: [random.sh](#) и написала соответствующий скрипт

```
emgorpinich@emgorpinich:~$ touch random.sh
emgorpinich@emgorpinich:~$ emacs &
[2] 3115
```

```
#!/bin/bash
k=$1
for (( i=0; i<$k; i++ ))
do
    (( char=$RANDOM%26+1 ))
    case $char in
        1) echo -n a;; 2) echo -n b;; 3) echo -n c;; 4) echo -n d;; 5) echo -n e;; 6) echo -n f;; 7) echo -n g;; 8) echo -n h;; 9) echo -n i;; 10) echo -n j;; 11) echo -n k;; 12) echo -n l;; 13) echo -n m;; 14) echo -n n;; 15) echo -n o;; 16) echo -n p;; 17) echo -n q;; 18) echo -n r;; 19) echo -n s;; 20) echo -n t;; 21) echo -n u;; 22) echo -n v;; 23) echo -n w;; 24) echo -n x;; 25) echo -n y;; 26) echo -n z;;
    esac
done
echo
```

Далее я проверила работу написанного скрипта, предварительно добавив право на исполнение файла. Скрипт работает корректно.

```
emgorpinich@emgorpinich:~$ chmod +x random.sh
emgorpinich@emgorpinich:~$ ./random.sh 4
jety
emgorpinich@emgorpinich:~$ ./random.sh 29
xeeuturunzmzkwtlwcouwrlmofvea
```

Вывод

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