

Отчёт по лабораторной работе №13

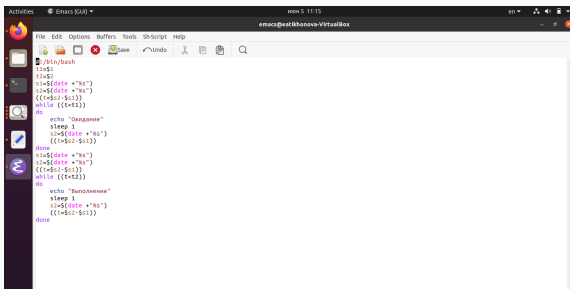
Тихонова Екатерина Андреевна

19 мая 2021

РУДН, Москва, Россия

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

Выписала 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: `sem.sh` и написала соответствующий скрипт



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

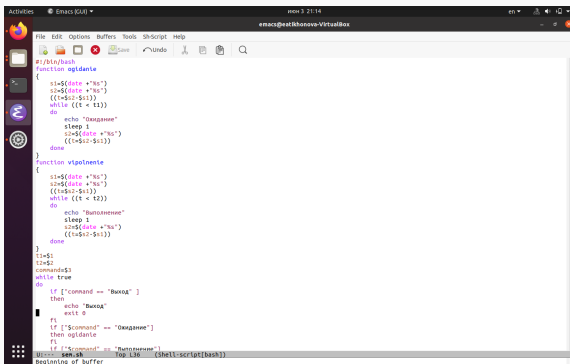
Далее я проверила работу написанного скрипта(команда«./sem.sh47»), предварительно добавив право на исполнение файла (команда «chmod+xsem.sh») Скрипт работает корректно.

```
eatikhonova@eatikhonova-VirtualBox:~$ chmod +x sem.sh
eatikhonova@eatikhonova-VirtualBox:~$ ./sem.sh 4 7
Ожидание
Ожидание
Ожидание
Ожидание
Выполнено
Выполнено
Выполнено
Выполнено
Выполнено
Выполнено
Выполнено
eatikhonova@eatikhonova-VirtualBox:~$ import ~/work/2020-2021/os-intro/laboratory/lab13/image/2.png
```

Рис. 2: Проверяем работу

Изменила скрипт

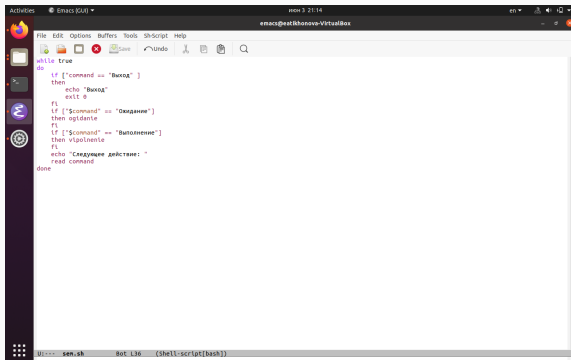
После этого я изменила скрипт так, чтобы его можно было выполнять в нескольких терминалах и проверила его работу (например, команда «./sem.sh 2 3 Ожидание > /dev/pts/1 &»)



```
#!/bin/bash
function ogidante
{
    s1=$(date +%s)
    s2=$(date +%s)
    ((t=s2-s1))
    while ((t < t1))
    do
        echo "Ожидание"
        sleep 1
        s2=$(date +%s)
        ((t=s2-s1))
    done
}
function vipolnente
{
    s1=$(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" == "Buxxa" ]
    then
        echo "Buxxa"
        exit 0
    fi
    if [ "$command" == "Ожидание" ]
    then
        ogidante
    fi
    if [ "$command" == "Выполнение" ]
    then
        vipolnente
    fi
done
Ujccw sem.sh Top 136 (shell:script(bash))
Beginning of buffer
```

Рис. 3: Изменила скрипт

Проверяем работу



The screenshot shows an Emacs editor window titled "Emacs [GUI]" with a menu bar (File, Edit, Options, Buffers, Tools, Sh-script, Help) and a toolbar. The main text area contains a shell script with a while loop. The status bar at the bottom indicates the file is "sem.sh" and the buffer is "Buf_L36 (shell-script[batch])".

```
while true
do
  if [ "$command" == "Выход" ]
  then
    echo "Выход"
    exit 0
  fi
  if [ "$command" == "Ожидание" ]
  then
    ogidanie
  fi
  if [ "$command" == "Выполнение" ]
  then
    vipolnenie
  fi
  echo "Следующее действие: "
  read command
done
```

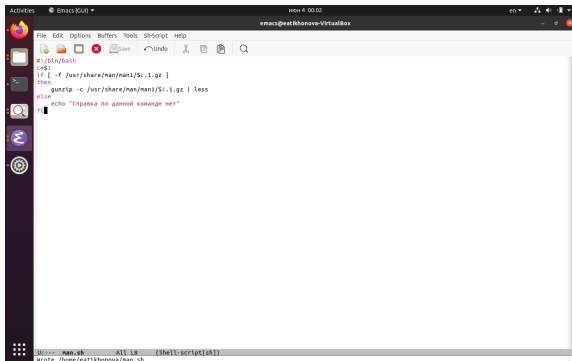
Рис. 4: Проверяем работу

```
Следующее действие:  
Выход  
Выход  
eatikhonova@eatikhonova-VirtualBox:~$ ./sem.sh 2 5 Выполнение > /dev/pts/2 &  
[2] 8146  
eatikhonova@eatikhonova-VirtualBox:~$ bash: /dev/pts/2: Permission denied  
Выход  
Выход: command not found  
[2]+ Exit 1                      ./sem.sh 2 5 Выполнение > /dev/pts/2
```

Рис. 5: Проверяем работу

Пишем скрипт

Для данной задачи я создала файл: man.sh и написала соответствующий скрипт



```
File Edit Options Buffers Tools Shell-script Help
# /bin/bash
c=$1
if [ -f /usr/share/man/man1/$c.1.gz ]
then
    gunzip -c /usr/share/man/man1/$c.1.gz | less
else
    echo "Справка по данной команде нет"
fi
```

UJ>>> man.sh All 18 (shell:script[sh])
Wrote /home/eatikhonova/man.sh

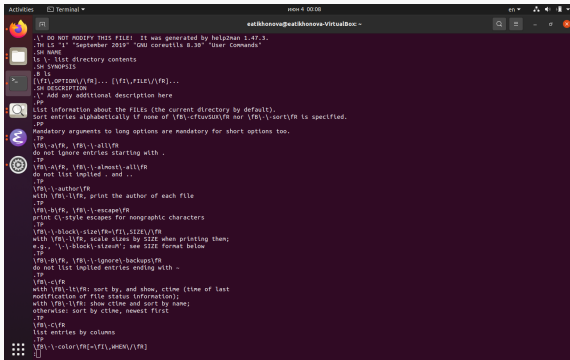
Рис. 7: Пишем скрипт

Далее я проверила работу написанного скрипта(команды«./man.shls»и «./man.shmkdir»), предварительно добавив право на исполнение файла (команда «chmod+xman.sh») Скрипт работает корректно.

```
eatikhonova@eatikhonova-VirtualBox:~$ chmod +x man.sh  
eatikhonova@eatikhonova-VirtualBox:~$ ./man.sh ls  
eatikhonova@eatikhonova-VirtualBox:~$ ./man.sh mkdir
```

Рис. 8: Проверяем работу

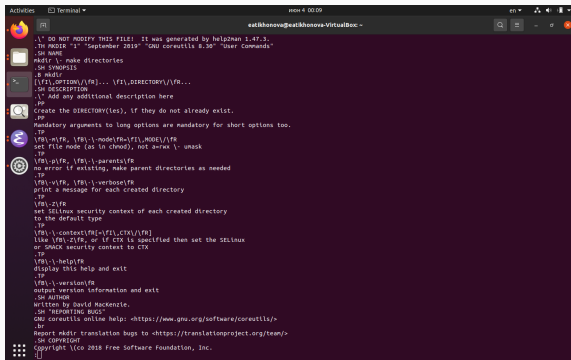
Проверяем работу



```
Activities [ Terminal ] mon 4 00:08
eatkhonova@eatkhonova-VirtualBox: ~
.\ DO NOT MODIFY THIS FILE! It was generated by helpman 1.47.3.
.TM LS "1" "September 2019" "GNU coreutils 8.30" "User Commands"
.SM NAME
ls \- list directory contents
.SM SYNOPSIS
.B ls
[(F1),OPTION\[/F\]]... [(F2),FILE\[/F\]]...
.SM DESCRIPTION
.\ Add any additional description here
.FM
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of /Fb/-cfuvSUx/Fn nor /Fb/-/-sort/Fn is specified.
.FM
Mandatory arguments to long options are mandatory for short options too.
.TP
/Fb/-a\[/F\], /Fb/-\-all\[/F\]
do not ignore entries starting with .
.TP
/Fb/-A\[/F\], /Fb/-\-almost\-all\[/F\]
do not list implied . and ..
.TP
/Fb/-l\[/F\], /Fb/-\-author\[/F\]
with /Fb/-l\[/F\], print the author of each file
.TP
/Fb/-b\[/F\], /Fb/-\-escape\[/F\]
print C-style escapes for nongraphic characters
.TP
/Fb/-b\[/F\], /Fb/-\-block\-size\[/F\], /Fb/-\-size\[/F\]
with /Fb/-l\[/F\], scale sizes by SIZE when printing them;
e.g., '\-block\-size\[/F\]'; see SIZE format below
.TP
/Fb/-B\[/F\], /Fb/-\-ignore\-backups\[/F\]
do not list implied entries ending with ~
.TP
/Fb/-c\[/F\]
with /Fb/-l\[/F\]: sort by, and show, ctime (time of last
modification of file status information);
with /Fb/-l\[/F\]: show ctime and sort by name;
otherwise: sort by ctime, newest first
.TP
/Fb/-C\[/F\]
list entries by columns
.TP
/Fb/-\-color\[/F\][=F1,WHEN\[/F\]]
```

Рис. 9: Проверяем работу

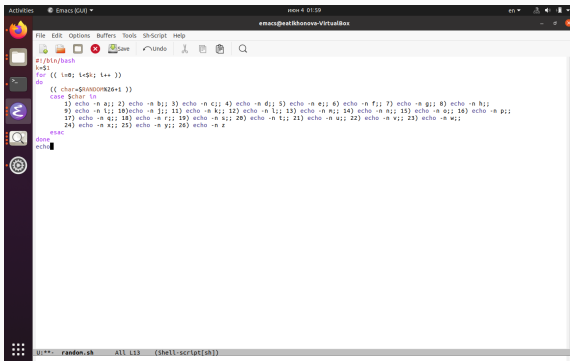
Проверяем работу



```
#!/bin/sh
# DO NOT MODIFY THIS FILE! It was generated by help2man 1.47.3.
# The mkdir "1" "September 2019" "GNU coreutils 8.30" "User Commands"
#
# NAME
# mkdir - make directories
#
# SYNOPSIS
# B mkdir
# [(-i),OPTION]\[FR]... \[FI],DIRECTORY\[\[FR]...
#
# DESCRIPTION
# \[ Add any additional description here
#
# Create the DIRECTORY(ies), if they do not already exist.
#
# Mandatory arguments to long options are mandatory for short options too.
#
# -m, --mode=FR, \[FR]\[-mode=FR]\[FI],MODE\[/FR
# set file mode (as in chmod), not a-w-x \[- umask
#
# -p, --parents, \[FR]\[-parent\[/FR
# no error if existing, make parent directories as needed
#
# -v, --verbose, \[FR]\[-verbose\[/FR
# print a message for each created directory
#
# -Z, --context=Z\[/FR
# set SELinux security context of each created directory
# to the default type
#
# \[FR]\[-context=FR]\[FI],CTX\[/FR]
# like \[FR]\[-Z\[/FR, or if CTX is specified then set the SELinux
# or SMACK security context to CTX
#
# -h, --help, \[FR]\[-help\[/FR
# display this help and exit
#
# -V, --version, \[FR]\[-version\[/FR
# output version information and exit
#
# AUTHOR
# Written by David Mackenzie.
#
# "REPORTING BUGS"
# GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
#
# Report mkdir translation bugs to <https://translationproject.org/team/>
#
# Copyright (C) 2018 Free Software Foundation, Inc.
#
```

Рис. 10: Проверяем работу

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



```
#!/bin/bash
i=$1
for (( i=0; i<$i; 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
```

Рис. 11: Пишем скрипт

Далее я проверила работу написанного скрипта(команды «./random.sh7»и «./random.sh15»), предварительно добавив право на исполнение файла (команда «chmod+xrandom.sh») Скрипт работает корректно.

```
eatikhonova@eatikhonova-VirtualBox:~$ chmod +x random.sh
eatikhonova@eatikhonova-VirtualBox:~$ ./random.sh 15
rtwuakpednjlfw
eatikhonova@eatikhonova-VirtualBox:~$ ./random.sh 7
ohaxosy
```

Рис. 12: Проверяем работу

В ходе выполнения данной лабораторной работы я изучил основы программирования в оболочке ОС UNIX, а также научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Спасибо за внимание!