

Презентация к лабораторной работе №12

Ермолаев А.М.

Презентация к лабораторной работе №12

Цель работы

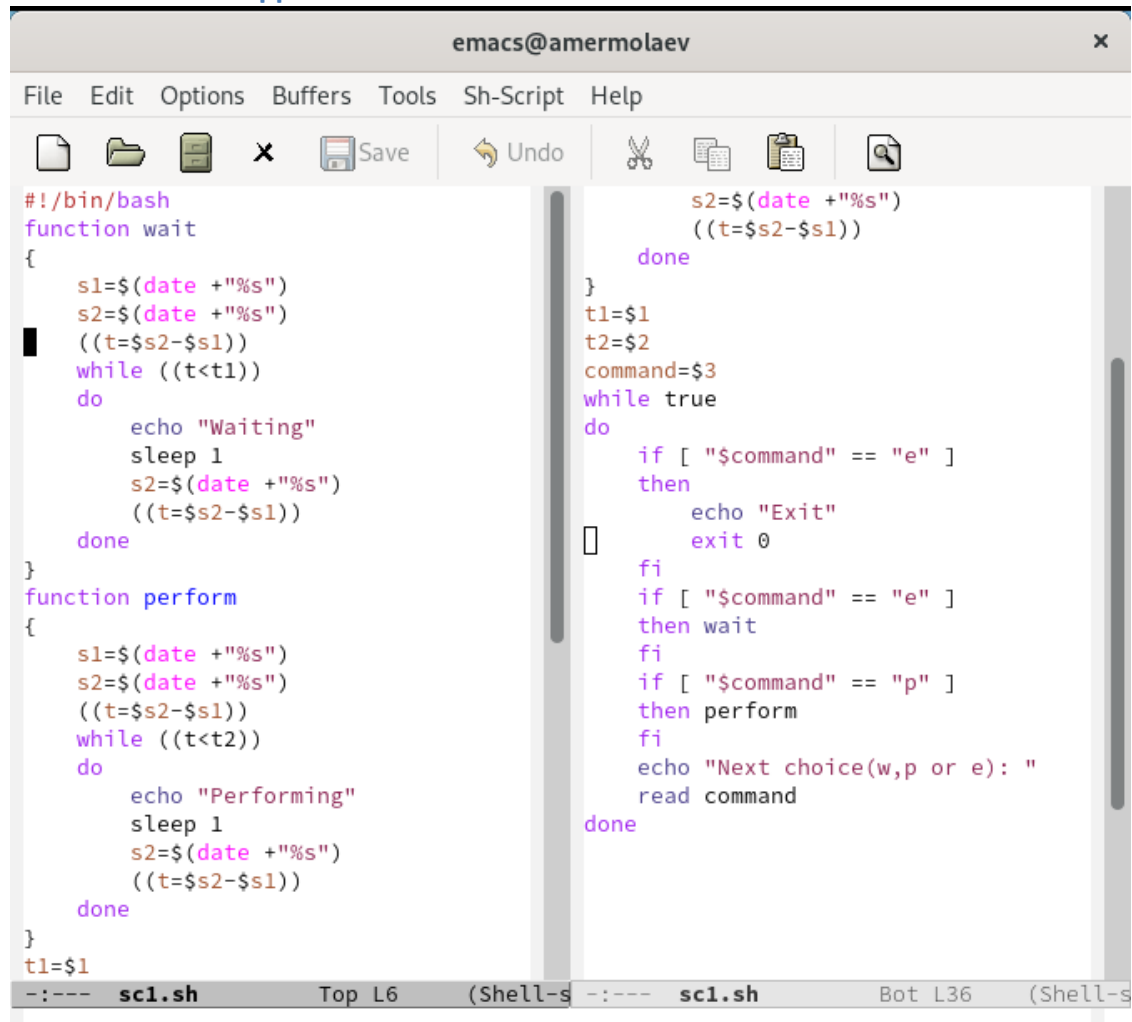
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.



цель

Выполнение работы

Выполнение задания 1



```
#!/bin/bash
function wait
{
    s1=$(date +%s)
    s2=$(date +%s)
    ((t=s2-s1))
    while ((t<t1))
    do
        echo "Waiting"
        sleep 1
        s2=$(date +%s)
        ((t=s2-s1))
    done
}
function perform
{
    s1=$(date +%s)
    s2=$(date +%s)
    ((t=s2-s1))
    while ((t<t2))
    do
        echo "Performing"
        sleep 1
        s2=$(date +%s)
        ((t=s2-s1))
    done
}
t1=$1
s2=$(date +%s)
((t=s2-s1))
done
}
t1=$1
t2=$2
command=$3
while true
do
    if [ "$command" == "e" ]
    then
        echo "Exit"
        exit 0
    fi
    if [ "$command" == "e" ]
    then wait
    fi
    if [ "$command" == "p" ]
    then perform
    fi
    echo "Next choice(w,p or e): "
    read command
done
```

листинг первой программы

```
amermolaev@amermolaev:~  
[amermolaev@amermolaev ~]$ ./sc1.sh 2 4 /dev/pts/1  
Next choice(w,p or e):  
w  
Next choice(w,p or e):  
p  
Performing  
Performing  
Performing  
Performing  
Next choice(w,p or e):  
e  
Exit  
[amermolaev@amermolaev ~]$
```

исполнение первой программы

Выполнение второго задания

```
amermolaev@amermolaev:/usr/share/man/man1  
[amermolaev@amermolaev man1]$ cd /usr/share/man/man1  
[amermolaev@amermolaev 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
```

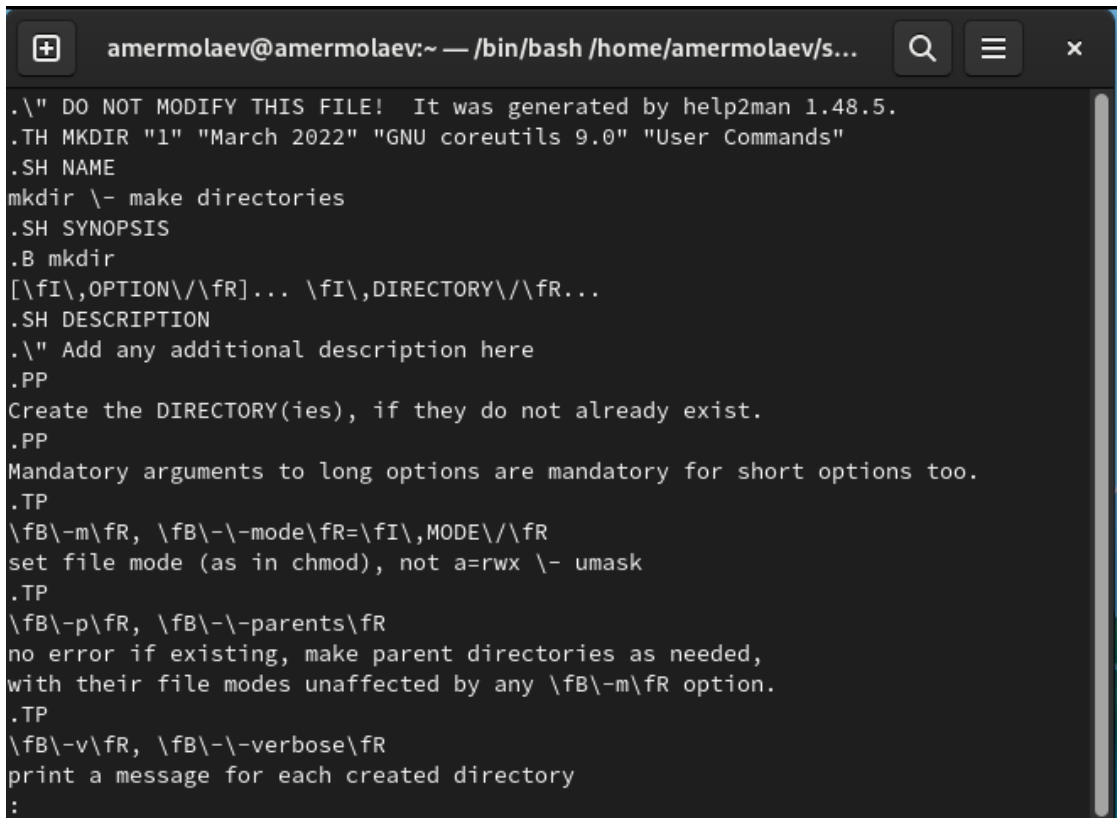
просмотр содержимого интересующего каталога

fi
The Emacs interface includes a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a toolbar with icons for file operations. The status bar at the bottom shows 'U:--- sc2.sh All L7 (Shell-script[bash])'." data-bbox="135 68 819 357"/>

```
#!/bin/bash
a=$1
if [ -f /usr/share/man/man1/$a.1.gz ]
then
gunzip -c /usr/share/man/man1/$1.1.gz | less
else
echo "No such file"
fi
```

U:--- sc2.sh All L7 (Shell-script[bash])

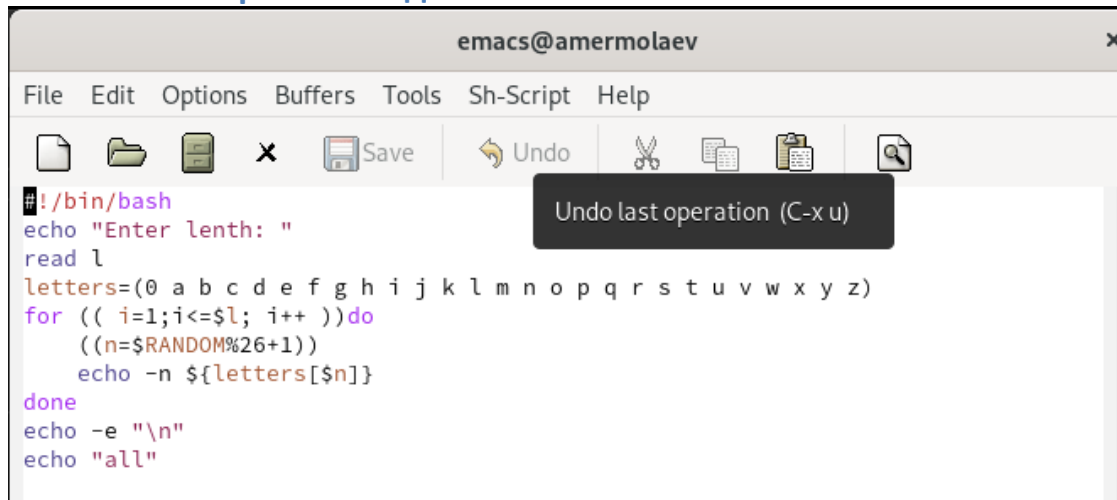
листинг второй программы



```
amermolaev@amermolaev:~ — /bin/bash /home/amermolaev/s...
.\" DO NOT MODIFY THIS FILE! It was generated by help2man 1.48.5.
.TH MKDIR "1" "March 2022" "GNU coreutils 9.0" "User Commands"
.SH NAME
mkdir \- make directories
.SH SYNOPSIS
.B mkdir
[\\fI\\,OPTION\\/\fR]... \\fI\\,DIRECTORY\\/\fR...
.SH DESCRIPTION
.\" Add any additional description here
.PP
Create the DIRECTORY(ies), if they do not already exist.
.PP
Mandatory arguments to long options are mandatory for short options too.
.TP
\\fB\\-m\\fR, \\fB\\-\\-mode\\fR=\\fI\\,MODE\\/\fR
set file mode (as in chmod), not a=rwx \\- umask
.TP
\\fB\\-p\\fR, \\fB\\-\\-parents\\fR
no error if existing, make parent directories as needed,
with their file modes unaffected by any \\fB\\-m\\fR option.
.TP
\\fB\\-v\\fR, \\fB\\-\\-verbose\\fR
print a message for each created directory
:
```

исполнение второй программы

Выполнение третьего задания



```
#!/bin/bash
echo "Enter lenth: "
read l
letters=(0 a b c d e f g h i j k l m n o p q r s t u v w x y z)
for (( i=1;i<=$l; i++ ))do
    ((n=$RANDOM%26+1))
    echo -n ${letters[$n]}
done
echo -e "\n"
echo "all"
```

листинг третьей программы



```
[amermolaev@amermolaev ~]$ ./sc4.sh
Enter lenth:
800
mtnczejorchmojcgxdudxarrirlptvehzvgrgphkfjllenlspajjmlwqcctfxoiyqrxypbtyooaryjm
vtyedovgeiqfpphgkidbggprhnwrntzespvaokeizhvfdlmvbxicouvklcqwhncqkesbfilktfkwczu
grkkvdgmluqtuqtpgubazusjljfhvboyxsymqlrxmhxbocstycfkttnofyqdwmyghwngezbiukzvw
bawbveqtscsgblthlrdjvpktwyedifnqrtldlcrtybnumjtbcoqcmhdfsprhetrzpbttkwwzcuvjkmv
yrkcqbivhdegxhgefgwugrsvijlashzpsudumoalfbzjluckjlrxwvclmkxixawtcdmsehuepcocz
pthlimcylfkicpflrqpujdfafqkrvglliycxhxciguvxdkyluvakdfrbcflirpvuyyexllulaugtter
valmsvzajfxmewgsjcbtkmmvzvxfujrwdzzbejbxbbehakdcqxsnsfbqflplvvsphptceblaztiafcpl
chhxxzeubpojrfojdzprpebeqlthkrzitufasqreaizvwxfddeejwzeadelndwvwagyjzudtizpqkhrq
ablwsztifwkgatbctqewccyexwawmxufsfdneyigchitudnrbfhvfxfphjapuddcjt看exevlbopkvoir
oqwqpvgbplrcnphcflpswuigylbnoupyzvwoqmpijtokpdsusktykqdrpdefaomqibgpkujdchkpj

all
[amermolaev@amermolaev ~]$
```

исполнение второй программы

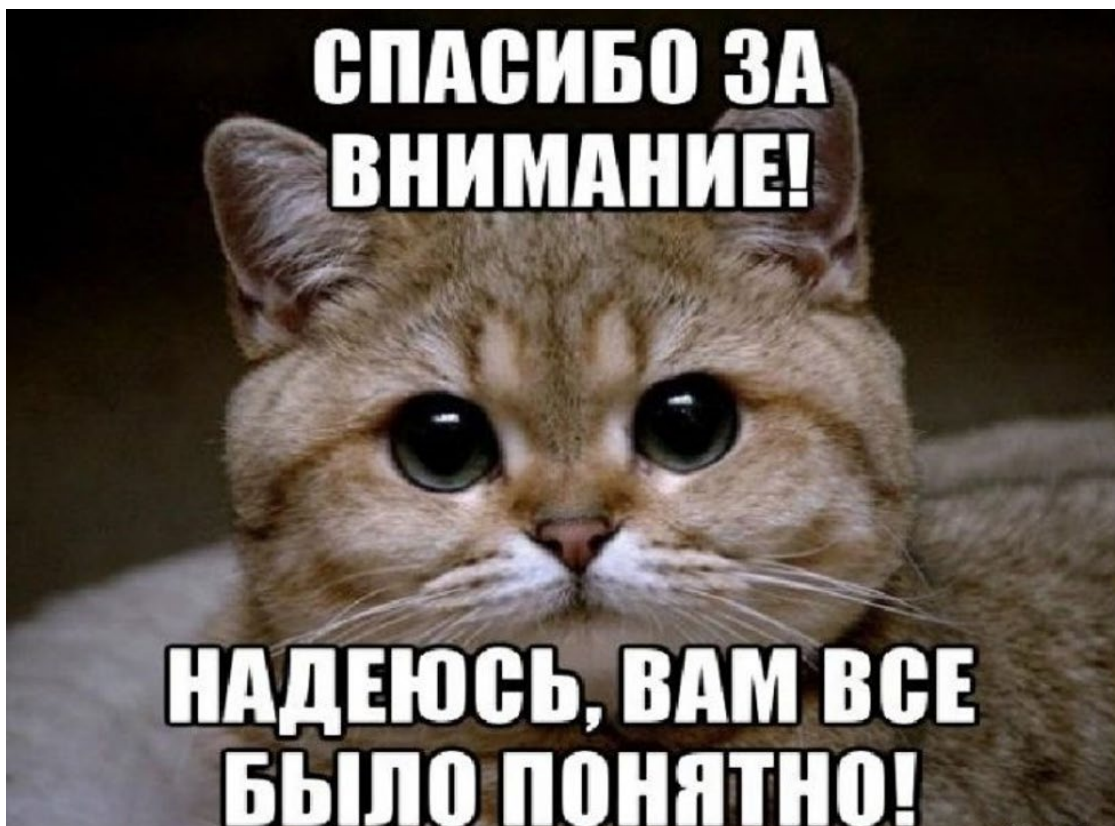
Вывод

В рамках выполнения работы я изучил основы программирования в оболочке ОС UNIX и научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.



conclusion

Финал



final