

Отчет к лабораторной работе №12

Common information

discipline: Операционные системы

group: НПМбд-01-21

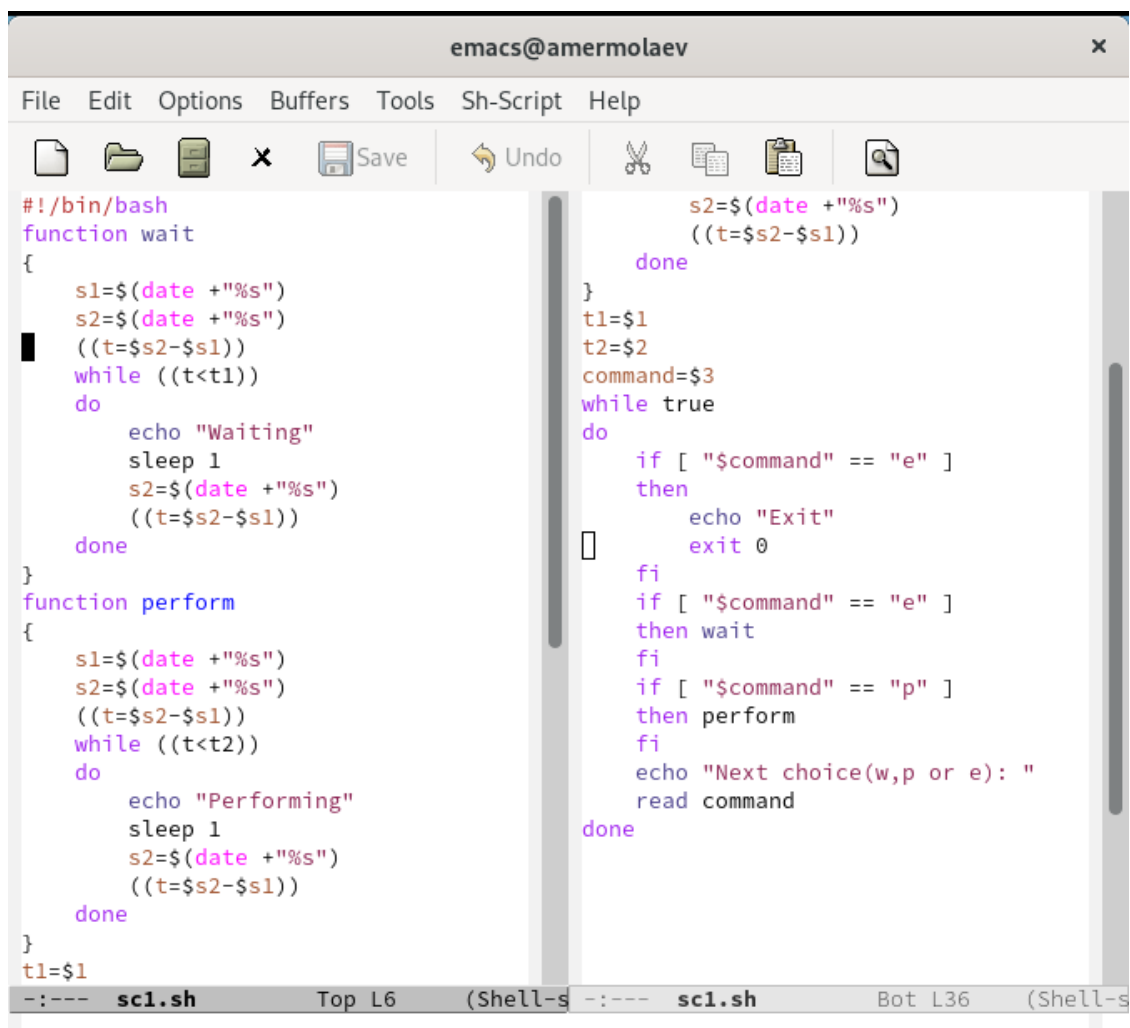
author: Ермолаев А.М.

Цель работы

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

Выполнение работы

Напишем 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). Запустим 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. Доработаем программу так, чтобы имелась возможность взаимодействия трёх и более процессов.



```
emacs@amermolaev
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/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
-:--- sc1.sh Top L6 (Shell-s -:--- sc1.sh Bot L36 (Shell-s
```

листинг первой программы

Предоставим файлу право на исполнение командой `chmod +x sc1.sh` и проверим корректность работы.

```
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

Реализуем команду man с помощью командного файла. Для этого изучим 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.

```
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

```
emacs@amermolaev
File Edit Options Buffers Tools Sh-Script Help
[Icons: File, Folder, Disk, X, Save, Undo, Cut, Copy, Paste, Find]
#!/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

Проверим кооректность работы программы для команды mkdir.

```
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

Используя встроенную переменную \$RANDOM, напомним командный файл, генерирующий случайную последовательность букв латинского алфавита. Будем иметь в виду, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

```
emacs@amermolaev
File Edit Options Buffers Tools Sh-Script Help
[Icons] Save Undo [Icons]
#!/bin/bash
echo "Enter length: "
read l
letters=(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 lenh:
800
mtnczejorchmojcgxdudxarirlptvehzvgrgphkfjqlenlspajjmlwqcctfxoiyqrxypbtyooaryjm
vtyedovgeiqfpphgkidbggprhnwrntzespvaokeizhvfdlmbxlcouvklcqwhncqkesbfilktfkwczu
grkkvdgmoluqtuqtpgubazusjljfhvboyxsymqlrxmhxbocstycfkttnofyqdwmyghwngezbiukzvw
bawbveqtsccsgblthlrdjvpktwyedifnqrdtclrtymbxnumjtbcoqcmhdfsprhetrzpbttkwwzcuvjkmv
yrkcqbivhdegxhgefugrsvijlashzpsudumoalfibzjluckjlrwxvclmkxixawtcdmsehuepcocz
pthlimcylfkicpflrqpujdfafqkrvglliyxhxciguvxdkyluvakdfrbcflirpvuyyexllulaugtter
valmsvzajfxmewgsjcbtkmmvzvxfujrwdzzbejxbbehakdcqxsnsfbqflplvvsphpteblaztiafcpl
chhxxzeubpojrfojdzprpebeqlthkrzitufasqreaizvwxfdeejezadeleldvwwagyjzudtizpqkhrq
ablwszti fwkgatbctqewccyexwawmxufsfndnyigchitudnrbvf hvfxphjapuddcjt wexevlbopkvoir
oqwqpvgbplrcnphcflpswui gylnoupyzvwoqmpij tokpdsusktykqdrpdefaomqibgpkujdchkpj

all
amermolaev@amermolaev ~]$
```

исполнение второй программы

Ответы на контрольные вопросы

Вопрос 1

В данной строке отсутствуют пробелы после первой скобки и перед последней скобкой. К тому же, не лишним было бы обрамление \$1 в ", так как данная переменная может содержать пробелы.

Вопрос 2

Самый простой способ объединить две или более строковые переменные — записать их одну за другой:

```
VAR1="Hello,"
VAR2=" World"
VAR3="$VAR1$VAR2"
echo "$VAR3"
```

Вы также можете объединить одну или несколько переменных с литеральными строками:

```
VAR1="Hello,"
VAR2="${VAR1}World"
echo "$VAR2"
```

Другой способ объединения строк в bash — добавление переменных или литеральных строк к переменной с помощью оператора +=:

```
VAR=""
for ELEMENT in 'Hydrogen' 'Helium' 'Lithium' 'Beryllium'; do
    VAR+="${ELEMENT} "
done
```

```
echo "$VAR"
```

Вопрос 3

Команда seq выводит последовательность целых или действительных чисел, подходящую для передачи в другие программы.

Команда seq может пригодиться в различных других командах и циклах для генерации последовательности чисел.

Общий синтаксис команды «seq»:

```
seq [options] specification
```

Например, чтобы просто напечатать последовательность чисел с 1 до 4, можно воспользоваться командой seq 4.

В качестве альтернативы ее можно реализовать при помощи цикла for:

```
for (( i = $1; i <= $2; i++ ))do
    echo $i
done
```

Вопрос 4

Результатом инструкции $\$(10/3)$ будет 3, так как происходит целочисленное деление.

Вопрос 5

Отличия командной оболочки zsh от bash:

- В zsh более быстрое автодополнение для cdc помощью Tab
- В zsh существует калькулятор zcalc, способный выполнять вычисления внутри терминала
- В zsh поддерживаются числа с плавающей запятой
- В zsh поддерживаются структуры данных «хэш»
- В zsh поддерживается раскрытие полного пути на основе неполных данных
- В zsh поддерживается замена части пути
- В zsh есть возможность отображать разделенный экран, такой же как разделенный экран vim

Вопрос 6

Синтаксис данной конструкции корректен.

Вопрос 7

Преимущества bash:

- Установлен по умолчанию в большинстве дистрибутивах Linux, MacOS
- Удобное перенаправление ввода/вывода
- Автоматизация некоторых действий с файловыми системами Linux
- Работа с серверами

Недостатки bash:

- Отсутствие дополнительных библиотек
- Не является кроссплатформенным языком
- Утилиты при выполнении скрипта запускают свои процессы, которые, в свою очередь, отражаются на скорости выполнения этого скрипта.

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.