

Федеральное государственное бюджетное образовательное учреждение высшего образования.  
«Национально исследовательский университет «Московский энергетический институт»  
Кафедра ВМСС

## Лабораторная работа №2

# ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ КОМАНДНОГО ИНТЕРПРЕТАТОРА SHELL.

Курс: МУЛЬТИЗАДАЧНЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ

Группа: А-07м-23

Выполнили: Балашов С.А.,

Кретов Н.В., Рогов Д.Р.

Проверил: Орлов Д.А.

Москва 2023 г.

2. Создайте в вашем домашнем каталоге подкаталог lab4, перейдите в него.

```
lab4 — -zsh — 80x5
~/lab4 — -zsh
Last login: Sat Nov 11 16:14:06 on tty??
(base) milord@MacBook-Air-2 ~ % mkdir lab4
(base) milord@MacBook-Air-2 ~ % cd lab4
(base) milord@MacBook-Air-2 lab4 %
```

3. Сохранить в виде ascii- файла F1.var значение внутренних переменных shell и номера процессов в системе.

```
lab4 — -zsh — 80x5
~/lab4 — -zsh
(base) milord@MacBook-Air-2 lab4 % touch F1.var
(base) milord@MacBook-Air-2 lab4 % echo -e "$HOME\n$PATH\n$PS1\n$PS2\n$TERM\n$MAIL\n" >> F1.var
(base) milord@MacBook-Air-2 lab4 % ps -e | awk '{print $1}' >> F1.var
(base) milord@MacBook-Air-2 lab4 %
```

```
lab4 — -zsh — 80x41
~/lab4 — -zsh
(base) milord@MacBook-Air-2 lab4 % cat F1.var
/Users/milord
/usr/local/mysql/bin:/opt/homebrew/opt/openssl@3/bin:/Users/milord/opt/anaconda3
/bin:/Users/milord/opt/anaconda3/condabin:/usr/local/bin:/System/Cryptexes/App/u
sr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.
system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system
/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/u
sr/appleinternal/bin:/opt/X11/bin:/Library/Apple/usr/bin:/Library/TeX/texbin:/Ap
plications/VMware Fusion.app/Contents/Public:/Library/Frameworks/Mono.framework/
Versions/Current/Commands:/Users/milord/Downloads/platform-tools/
(base) %n@m %m %1~ %#
%_>
xterm-256color

PID
1
122
124
126
127
128
131
134
136
137
142
147
153
154
158
161
162
163
164
165
166
167
169
171
172
```

4. Переопределить значения переменных PS1, PS2, сохранить в ascii- файле F2.var значения внутренних переменных.

```
lab4 — -zsh — 80x11
~/lab4 — -zsh
Last login: Sun Nov 12 13:41:00 on ttys006
(base) milord@MacBook-Air-2 lab4 % export PS1="[%T %W]
[%/ %M %n] $ "
[13:44 11/12/23]
[/Users/milord/lab4 MacBook-Air-2 milord] $ PS2="continue-> "
[13:45 11/12/23]
[/Users/milord/lab4 MacBook-Air-2 milord] $ echo

[13:45 11/12/23]
[/Users/milord/lab4 MacBook-Air-2 milord] $
```

```
lab4 — -zsh — 80x26
~/lab4 — -zsh
[13:46 11/12/23]
[/Users/milord/lab4 MacBook-Air-2 milord] $ touch F2.var
[13:46 11/12/23]
[/Users/milord/lab4 MacBook-Air-2 milord] $ echo -e "$HOME\n$PATH\n$PS1\n$PS2\n$TERM\n$MAIL\n" >> F2.var
[13:46 11/12/23]
[/Users/milord/lab4 MacBook-Air-2 milord] $ cat F2.var
/Users/milord
/usr/local/mysql/bin:/opt/homebrew/opt/openssl@3/bin:/Users/milord/opt/anaconda3/bin:/Users/milord/opt/anaconda3/condabin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/X11/bin:/Library/Apple/usr/bin:/Library/TeX/texbin:/Applications/VMware Fusion.app/Contents/Public:/Library/Frameworks/Mono.framework/Versions/Current/Commands:/Users/milord/Downloads/platform-tools/
[%T %W]
[%/ %M %n] $
continue->
xterm-256color

[13:46 11/12/23]
[/Users/milord/lab4 MacBook-Air-2 milord] $
```

5. Вернуться к исходному значению внутренних переменных shell.

```
lab4 — -zsh — 80x5
~/lab4 — -zsh
50868
[13:51 11/12/23]
[/Users/milord/lab4 MacBook-Air-2 milord] $ export PS1="(base) %n@m %1~ %#"
(base) milord@MacBook-Air-2 lab4 %PS2="%_>"
(base) milord@MacBook-Air-2 lab4 %
```

6. В каталоге lab4 получить копию файла /etc/skel/local.profile с именем .profile, определить в нем значения переменных DIGIT\_FOR\_LAB, MYOWNNAME, TIMEZONE, MYGROUP, DAY\_OF\_WEEK\_TODAY, а также измените на действующие значения внутренних переменных интерпретатора PATH и HOME. Стартуйте модифицированную таким образом командную оболочку. Убедитесь в результате.

```
milord — -zsh — 80x11
~ — -zsh
(base) milord@MacBook-Air-2 ~ % cp ~/.zshrc .profile
(base) milord@MacBook-Air-2 ~ % cd ~
(base) milord@MacBook-Air-2 ~ % echo DIGIT_FOR_LAB=2 >> .zshrc
echo MYOWNNAME=savvadimakolya >> .zshrc
echo TIMEZONE=MOSCOW >> .zshrc
echo MYGROUP=A-07m-23 >> .zshrc
echo DAY_OF_WEEK_TODAY=7 >> .zshrc
echo PATH=$PATH >> .zshrc
echo HOME=$HOME >> .zshrc
(base) milord@MacBook-Air-2 ~ %
```

```
milord — -zsh — 80x11
~ — -zsh
(base) milord@MacBook-Air-2 ~ % echo $MYGROUP
A-07m-23
(base) milord@MacBook-Air-2 ~ %
```

7. Напишите программу на языке shell - "скрипт", которая по вводимой вами произвольной последовательности десятичных чисел осуществляла бы вычисление промежуточных сумм и выводила конечный результат на экран и в файл sh.result. Используйте в программе значение переменной интерпретатора S#. Если номер Вашей бригады четный - постарайтесь обойтись оператором for, если нечетный - конструкцией if.

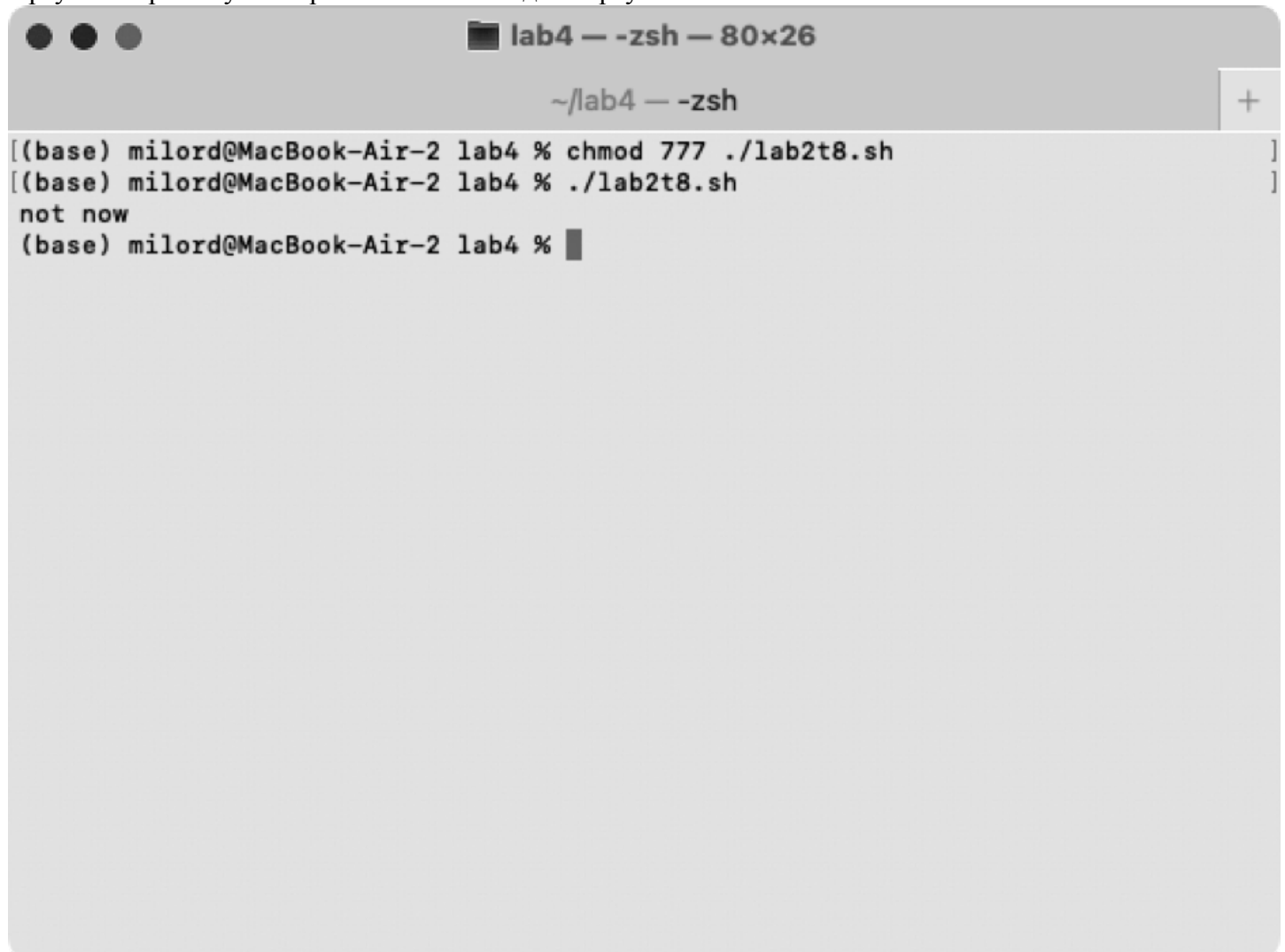
```
lab4 — -zsh — 80x26
~/lab4 — -zsh
[(base) milord@MacBook-Air-2 lab4 % ./lab2t7.sh ]
SUM=0
+
3
SUM=3
+
4
SUM=7
+
9
SUM=16
+
0
SUM=16
+
^C
(base) milord@MacBook-Air-2 lab4 %
```

```
#!/bin/zsh
sum=0
while true
do
    for i in $*
    do
        sum=$((sum+i))
        shift
    done
    echo SUM=$sum
    echo SUM=$sum >> sh.result
    echo "+"
    read N
    sum=$((sum+N))
done
```

8. Напишите скрипт, который выводил бы через каждые десять секунд на экран некоторое сообщение только в том случае, если наступит определенное время в определенный день.

Режим запуска - фоновый. Уточнение задания - см. табл.4.1

Аргумент при запуске скрипта – значения для переустановки PS1



```
#!/bin/bash
day=3
hour=17
count=0
while true
do
    currentDay="$(date +%d) "
    if [ $day -eq $currentDay ]
    then
        currentHour="$(date +%H) "
        if [ $hour -eq $currentHour ]
        then
            ps1=$1
            echo "Message$count=$ps1"
            count="$( (count+1) )"
        else
            echo "not now"
            exit 0
        fi
    else
        echo "not now"
        exit 0
    fi
    sleep 10s
done
```

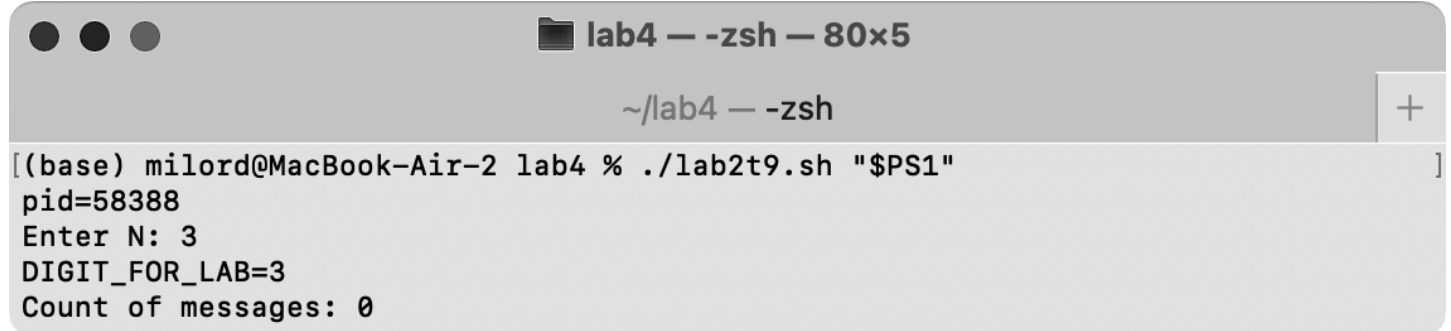
9. Напишите скрипт, который проверял бы наличие процесса, связанного со скриптом по пункту номер 8) и останавливал бы его, если количество сообщений, выведенных на экран скриптом по пункту номер 8) превысит некоторое число.

Необходимое число повторов - в переменной DIGIT\_FOR\_LAB

Значение DIGIT\_FOR\_LAB установить на N больше, чем количество повторов по скрипту п.8), установка N - в диалоговом режиме.

Значение DIGIT\_FOR\_LAB не может быть отрицательным, нулевым и больше 5

Первое сообщение после старта скрипта по п.9) - вывод значения текущего числа повторов сообщений, выведенных скриптом по п.8).



```
lab4 — -zsh — 80x5
~/lab4 — -zsh
[(base) milord@MacBook-Air-2 lab4 % ./lab2t9.sh "$PS1"
pid=58388
Enter N: 3
DIGIT_FOR_LAB=3
Count of messages: 0]
```

```
#!/bin/bash
process="$(ps -e | grep script_time)"
pid="$( echo $process | cut -d' ' -f1 )"
count=0
echo "pid=$pid"
if [ "$pid" == "" ]
then
    echo "process not started" exit
fi
read -p "Enter N: " N
DIGIT_FOR_LAB=$((DIGIT_FOR_LAB+N))
echo "DIGIT_FOR_LAB=$DIGIT_FOR_LAB"
if [ $DIGIT_FOR_LAB -le 0 -o $DIGIT_FOR_LAB -gt 5 ]
then
    echo "DIGIT_FOR_LAB=$DIGIT_FOR_LAB < 0 or DIGIT_FOR_LAB=$DIGIT_FOR_LAB > 5"
exit
fi
while [ $count -lt $DIGIT_FOR_LAB ]
do
    sudo timeout 10s strace -e trace=write -fp$pid 2>>log | grep -o '".\+[^"]"'
    count="$(grep -c Message log)"
    echo "Count of messages: $count"
done
echo "count of messages greater then DIGIT_FOR_LAB"
kill $pid
rm log
```

10. Напишите скрипт без использования диалога с оператором на языке shell, который через несколько минут (5-10) закроет все процессы, связанные с пунктами 7 - 9. Если номер Вашей бригады четный - постарайтесь обойтись конструкцией if, иначе - конструкцией for.

Примечание: перед запуском скрипта покажите его текст преподавателю и только после получения разрешения - запустите на исполнение. Подсказка к выполнению: кое-о-чем Вы позаботились в начале лабораторного задания; это "кое-что" позволит Вам выполнить задание.

```
#!/bin/bash
process1="$(ps -e | grep script_sum)"
pid1="$( echo $process1 | cut -d' ' -f1 )"
if [ "$pid1" == "" ]
then
    kill $pid1
    echo 'process script_sum finished'
fi
```

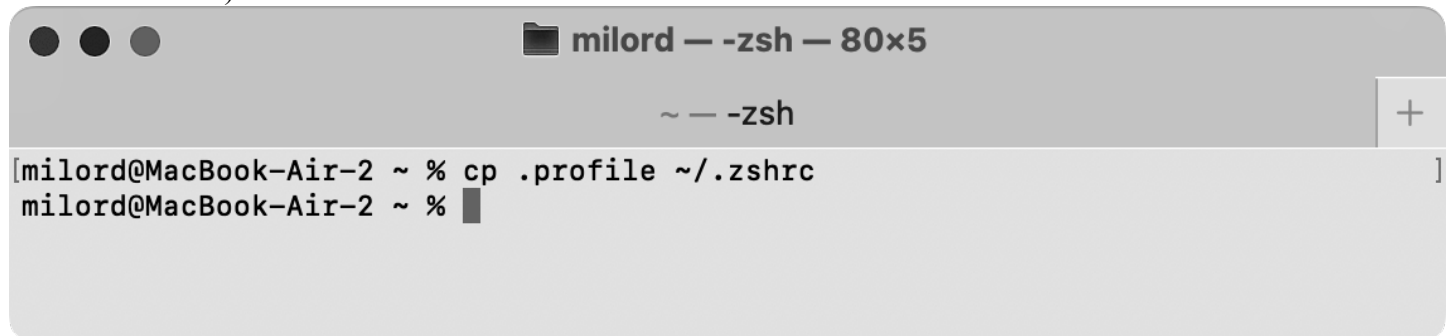


```

process2="$(ps -e | grep script_time)"
pid2="$( echo $process2 | cut -d' ' -f1 )"
if [ "$pid2"="" ]
then
    kill $pid2
    echo 'process script_time finished'
fi
process3="$(ps -e | grep script_proc)"
pid3="$( echo $process3 | cut -d' ' -f1 )"
if [ "$pid3"="" ]
then
    kill $pid3
    echo 'process script_proc finished'
fi

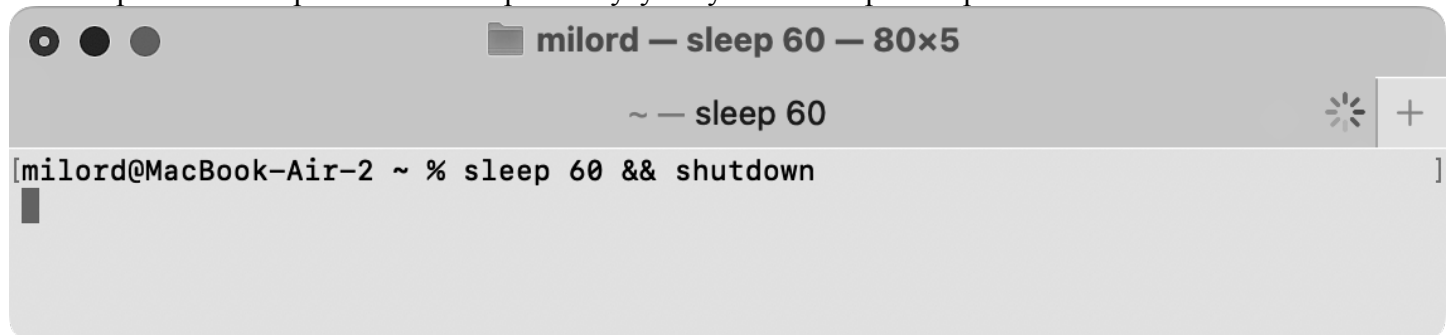
```

11. После запуска скрипта п.10 вернитесь в оболочку shell которая была в Вашем распоряжении до выполнения п.6).



A terminal window titled "milord — -zsh — 80x5" with a sub-title "~ — -zsh". The prompt is "milord@MacBook-Air-2 ~ %". The user enters the command "cp .profile ~/.zshrc" and the prompt returns to "milord@MacBook-Air-2 ~ %".

12. Завершите сеанс работы с ОС через минуту запуска некоторого скрипта.



A terminal window titled "milord — sleep 60 — 80x5" with a sub-title "~ — sleep 60". The prompt is "milord@MacBook-Air-2 ~ %". The user enters the command "sleep 60 && shutdown" and the prompt returns to "milord@MacBook-Air-2 ~ %".