

**Министерство науки и образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"Московский институт электронной техники"
(МИЭТ)**

Отчет по лабораторной работе № 1

Выполнил: студент ПМ - 31

Мартынова Мария Олеговна

2023 г.

Задание 1

- Запустите скрипт background.sh в фоновом режиме.
- Создайте текстовый файл и выведите его содержимое на экран терминала.
- Очистите окно терминала.
- Посчитайте количество символов в файле.

Решение

- & в shell

Shell (или «*шелл*», «*командная оболочка*») — это не только командный интерпретатор, который обеспечивает интерфейс взаимодействия между пользователем и ядром операционной системы, но и своеобразный язык программирования, в котором присутствуют такие конструкции, как операторы условного ветвления, циклы, переменные и многое другое.

Операционная система (ОС) запускает командную оболочку для каждого пользователя, когда тот входит в систему или открывает окно терминала. Первым что пользователь увидит в окне терминала, будет **приглашение оболочки**.

Использование & один раз в конце команды или скрипта – это запуск команд в фоновом режиме.

```
Console  Shell  Markdown
~/oslab2019$ lab1/src/background.sh &
[1] 46
~/oslab2019$ 1 sec
2 sec
3 sec
4 sec
5 sec
6 sec
7 sec
8 sec
9 sec
10 sec
Done!
█
```

- Как работает команда `cat`

С помощью команды `cat` вы можете быстро создать файл и поместить в него текст. Для этого используйте оператор `>` redirect, чтобы перенаправить текст в файл.

```
cat > imiafaila.txt
```

Файл создан, и вы можете начать заполнять его текстом. Если вы хотите добавить несколько строк, просто нажимайте **Enter** в конце каждой строки. Когда вы закончите, нажмите **CTRL+D**, чтобы выйти из файла.

```
~/oslab2019$ cat > task1_2.txt
Living is harmful!
It is said on the BGD.
Take care!
~/oslab2019$ cat task1_2.txt
Living is harmful!
It is said on the BGD.
Take care!
~/oslab2019$ █
```

- Как работает команда `clear`

Команда `clear` очищает окно терминала. На самом деле вывод предыдущих команд в терминале никуда не удаляется, просто строка ввода команд перемещается на первую строку терминала, а под ней вы получаете чистое пространство.

сочетание клавиш **CTRL+L** может выполнять функцию, аналогичную данной команде.

Console Shell Markdown

```
5 sec
6 sec
7 sec
8 sec
9 sec
10 sec
Done!
cat > Task1_2.txt
living is harmful!
it is said on the BGD
take care![1]+ Done
lab1/src/background.sh
~/oslab2019$ cat Task1_2.txt
living is harmful!
it is said on the BGD
~/oslab2019$ rm Task1_2.txt
~/oslab2019$ cat > task1_2.txt
Living is harmful!
It is said on the BGD.
Take care!
~/oslab2019$ cat task1_2.txt
Living is harmful!
It is said on the BGD.
Take care!
~/oslab2019$ clear
```

Console Shell Markdown

```
~/oslab2019$
```

- Как работает команда `wc`

`-m` `--count` Показать количество символов в объекте

Console Shell Markdown

```
~/oslab2019$ wc task1_2.txt -m
53 task1_2.txt
~/oslab2019$
```

Задание 2

- С помощью команды `grep`, используя редирект (`>`), запишите в файл `with_cake.txt` все строки из файла `cake_rhymes.txt`, в которых есть слово `cake`.
- Сделайте команду `rm` "тихой", используя редирект в `/dev/null`.

Решение

```
~/oslab2019$ cd lab1/src
~/.../lab1/src$ grep cake cake_rhymes.txt > with_cake.txt
~/.../lab1/src$ cat with_cake.txt
```

✓ lab1

✓ src

>_ background.sh

≡ cake_rhymes.txt

>_ hello.sh

≡ with_cake.txt

✓ text

```
rm name.txt > /dev/null
```

```
m$ cat cake_rhymes.txt | grep cake > with_cake.txt
```

Задание 3

- Сделайте файл `hello.sh` исполняемым, выполните его.
- Напишите `bash` скрипт, который выводит текущий путь, текущую дату и время, а также содержимое переменной окружения `PATH`.

Решение

ОСНОВНЫЕ ПРАВА ДОСТУПА К ФАЙЛАМ В LINUX

Изначально каждый файл имел три параметра доступа. Вот они:

- **Чтение** - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем;
 - **Запись** - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги;
 - **Выполнение** - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу.
-
- **0** - никаких прав;
 - **1** - только выполнение;
 - **2** - только запись;
 - **3** - выполнение и запись;
 - **4** - только чтение;
 - **5** - чтение и выполнение;
 - **6** - чтение и запись;
 - **7** - чтение запись и выполнение.

[Команда `chmod` Linux - Losst](#)

```
~/oslab2019$ cd lab1/src
~/../lab1/src$ chmod 555 hello.sh
~/../lab1/src$ ./hello.sh
Hello, world!
~/../lab1/src$
```

lab1/src/task3_2.sh ×

```
1  #!/bin/sh
2
3  D=$(date +%Y-%m-%d)
4  T=$(date +%H:%M:%S)
5
6  pwd
7  echo "$D" "$T"
8  echo $PATH
9
```



```
~/oslab2019$ cd lab1/src
~/.../lab1/src$ chmod 555 task3_2.sh
~/.../lab1/src$ ./task3_2.sh
/home/runner/oslab2019/lab1/src
2022-02-13 13:19:38
/nix/store/sh92m3hjnsxl9z684aaxcwdzsa65i0hi-pid1-0.0.0-04e4f31/bin:/
nix/store/22jj08nss8zvbkgk52ikj09ssg6rw84rv-silver-searcher-2.2.0/bin
:/nix/store/aw2a9ws5lwqxv0kz63srvkaqpcmfspg0-bash/bin:/nix/store/176
gh50y24c0lx2bnnmsvf9wazb73php-coreutils-9.0/bin:/nix/store/zaamc32wi
jqka1mg8w2gr1f6bgvs7yyz-vscode-cpptools-dap-1.3.1/bin:/nix/store/ann
2w0blx5vrc8mw1hwnvb37gry6ci9a-replit-python-dap-wrapper-1.0.0/bin:/n
ix/store/4ddsa52p3k2mckfi7ri6wa6ilqwaib5f-node-dap/bin:/nix/store/hh
zckydwi442l85smbd512xlpw2qarxv-fluxbox/bin:/nix/store/mviq50cwshsyjji
q33g3y89vdg42bkrcv-prybar-nodejs-0.0.0-fc9ff1e/bin:/nix/store/mav425
1if1ahjq5vkwpycyvnm459c4-pulseaudio-14.2/bin:/nix/store/2py1rvdx7y
7fbi0yffw7dq2bacd15605-rfbproxy-0.0.0-b81eb2f/bin:/nix/store/0b2bvvi
ijkvq2nvm04dbq31wlq7h742m-tigervnc-1.12.0/bin:/nix/store/7v07bmyfjq0
jdnx6xwxlib05na8kksdf-xwininfo-1.1.4/bin:/nix/store/gfvyxpnr8ji58pdh
y7ns4zqy57z34aqm-replbox/bin:/home/runner/.nix-profile/bin:/usr/loca
l/go/bin:/opt/virtualenvs/python3/bin:/usr/GNUstep/System/Tools:/usr
/GNUstep/Local/Tools:/home/runner/.apt/usr/bin:/usr/local/sbin:/usr/
local/bin:/usr/sbin:/usr/bin:/sbin:/bin
~/.../lab1/src$
```

Задание 4

- Напишите скрипт `average.sh`, который выводит количество и среднее арифметическое его входных аргументов.
- С помощью `bash` и `dev/random` создайте файл `numbers.txt` из 150 случайных чисел.
- "Скормите" скрипту `average.sh` значения из файла `numbers.txt`.

Решение

Console Shell Markdown

```
~/.../lab1/src$ chmod 555 task4_1.sh
~/.../lab1/src$ ./task4_1.sh
Parameters do not found
~/.../lab1/src$ ./task4_1.sh 2 4 6
Avarege: 4
Quantity: 3
~/.../lab1/src$
```

lab1/src/task4_1.sh ×

```
1  #!/bin/sh
2
3  if [ $# -gt 0 ] # если количество
   параметров больше 0, то
4  then
5      sum=0
6      for param in "$@" # @-содержит все что
   передалось, можно перебирать
7      do
8          sum=$((sum+$param))
9      done
10
11     resul=$((sum/$#)) # #-количество
   параметров
12
13     echo "Avarege: $resul"
14     echo "Quantity: $#"
```

```
15 else
16     echo "Parameters do not found"
17 fi
```

```
~/.../lab1/src$ touch task4_2.sh
~/.../lab1/src$ chmod 555 task4_2.sh
~/.../lab1/src$ ./task4_2.sh
```

lab1/src/task4_2.sh ×

```
1  #!/bin/sh
2
3  od -A n /dev/null > numbers.txt
4  k=1
5  while [ $k -le 150 ]
6  do
7      echo "$(od -A n -i -N 1 /dev/random)"
      >> numbers.txt
8      k=$((k+1))
9  done
```

lab1/src/numbers.txt ×

1					90
2					19
3					65
4					155
5					210

lab1/src/numbers.txt ×

144					254
145					213
146					208
147					83
148					157
149					66
150					48
151					

```
~/.../lab1/src$ ./average.sh $(cat numbers.txt)
Avarege: 128
Quantity: 150
```