

1. Изучение сценариев bash

Лысенко Артем, группа 7.1

Исходный код всех скриптов доступен по ссылке:

<https://github.com/quicklybly/operating-systems-workshop/tree/master/lab1>

1. Вывод всех аргументов в обратном порядке

count=\$# - определение числа аргументов

((count--)) - операция декремента

Код скрипта:

```
#!/bin/bash
```

```
count=$#
```

```
while [[ $count -gt 0 ]]
```

```
do
```

```
  echo "${!count}"
```

```
  ((count--))
```

```
done
```

2. Создание файла

Валидация числа аргументов + проверка того, что файл с таким именем еще не существует

```
if [[ $# -eq 0 ]]
```

```
then
```

```
  echo "Usage: file_maker <name> [<mode>]"
```

```
  exit 2
```

```
elif [[ -e $1 ]]
```

```
then
```

```
  echo "File $(pwd)/$1 already exists"
```

```
  exit 2
```

fi

При некорректном параметре <mode>, программа отправляет результат chmod в /dev/null, печатает сообщение об ошибке и удаляет созданный файл.

Полный код скрипта.

```
#!/bin/bash
```

```
if [[ $# -eq 0 ]]
```

```
then
```

```
    echo "Usage: file_maker <name> [<mode>]"
```

```
    exit 2
```

```
elif [[ -e $1 ]]
```

```
then
```

```
    echo "File $(pwd)/$1 already exists"
```

```
    exit 2
```

```
fi
```

```
touch "$1"
```

```
if [[ $# -ge 2 ]]
```

```
then
```

```
    if ! chmod "$2" "$1" 2> /dev/null;
```

```
    then
```

```
        echo "Invalid mode $2, consider using chmod style"
```

```
        rm "$1"
```

```
        exit 2
```

```
    fi
```

```
fi
```

3. Калькулятор арифметических действий

Валидация числа переданных параметров

```
if [[ ! $# -eq 3 ]]
then
  echo "Usage: dummy_calculator <A> <+|-|*|/> <B>"
  exit 2
fi
```

Если в качестве второго параметра передается /, то программа проверяет, что третьим параметром передан не 0. В случае, если переданы не +, -, *, / Программа печатает сообщение об ошибке.

Полный код скрипта:

```
#!/bin/bash
```

```
if [[ ! $# -eq 3 ]]
```

```
then
```

```
echo "Usage: dummy_calculator <A> <+|-|*|/> <B>"
```

```
exit 2
```

```
fi
```

```
# never use expr in production code, use (( )) instead
```

```
case "$2" in
```

```
'+'|'|'-'|'*)
```

```
echo "$((expr "$1" "$2" "$3"))"
```

```
;;
```

```
'/')
```

```
if [[ $3 -eq 0 ]]
```

```
then
```

```
echo "Division by 0 not allowed"
```

```
exit 2
```

```
else
```

```
echo "$((expr "$1" "$2" "$3"))"
```

```
fi
```

```
;;
```

```
*)
```

```
echo "Unknown operator: $2, consider using <+|-|*|/>"
```

```
exit 2
```

```
;;
```

```
esac
```

