



МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



**НГТУ
НЭТИ** | **Факультет прикладной
математики и информатики**

Кафедра теоретической и прикладной информатики
Лабораторная работа № 2
по дисциплине «Операционные системы и компьютерные сети»

КОМАНДНЫЕ СЦЕНАРИИ LINUX

Бригада 4	ТАДЖИБАЕВ ЗАВКИДДИН
Группа ПМ-25	НАЗИРОВ ДЖАСУР

Преподаватели	КОБЫЛЯНСКИЙ В. Г. СИВАК М. А.
---------------	----------------------------------

Новосибирск, 2024

Цель работы

Приобретение практических навыков работы с интерфейсом командной строки ОС Linux.

Ход работы

1. Посмотреть справку по команде **echo**, ознакомиться с ее ключами.

```
[pmi-b2704@students ~]$ help echo
```

```
echo: echo [-neE] [arg ...]
```

```
Write arguments to the standard output.
```

```
Display the ARGs on the standard output followed by a newline.
```

```
Options:
```

```
-n          do not append a newline
```

```
-e          enable interpretation of the following backslash  
escapes
```

```
-E          explicitly suppress interpretation of backslash  
escapes
```

```
`echo' interprets the following backslash-escaped characters:
```

```
\a          alert (bell)
```

```
\b          backspace
```

```
\c          suppress further output
```

```
\e          escape character
```

```
\f          form feed
```

```
\n          new line
```

```
\r          carriage return
```

```
\t          horizontal tab
```

```
\v          vertical tab
```

```
\\          backslash
```

```
\0nnn      the character whose ASCII code is NNN (octal).  NNN  
can be
```

```
0 to 3 octal digits
```

```
\xHH       the eight-bit character whose value is HH (hexadeci-  
mal).  HH
```

```
can be one or two hex digits
```

```
Exit Status:
```

```
Returns success unless a write error occurs.
```

```
[pmi-b2704@students ~]$ man echo
```

```
[pmi-b2704@students ~]$ man echo
```

```
ECHO(1)                                User Commands
```

```
ECHO(1)
```

```
NAME
```

```
echo - display a line of text
```

```
SYNOPSIS
```

```
echo [SHORT-OPTION]... [STRING]...
```

```
echo LONG-OPTION
```

DESCRIPTION

Echo the STRING(s) to standard output.

- n do not output the trailing newline
- e enable interpretation of backslash escapes
- E disable interpretation of backslash escapes (default)
- help display this help and exit
- version output version information and exit

2. Освоить способы создания сценария, описанные в разделе 2.6

Способ первый:

```
[pmi-b2704@students ~]$ echo -e "ls -l \nman echo" > echo_primer.sh
[pmi-b2704@students ~]$ cat echo_primer.sh
ls -l
man echo
```

Способ второй:

```
[pmi-b2704@students ~]$ cat >cat_primer.sh
ls - 1
man echo
[pmi-b2704@students ~]$ cat cat_primer.sh
ls - 1
man echo
```

Способ третий:

```
[pmi-b2704@students ~]$ vi vi_primer.sh
[pmi-b2704@students ~]$ cat vi_primer.sh
ls -1
man echo
```

3. Разработать сценарий в соответствии с вариантом задания, определяемым по номеру бригады в таблице 2.

Вариант №4

4	Разработать сценарий, выполняющий следующие действия с целыми числами из указанного пользователем диапазона: а) вывод на экран всех четных чисел диапазона; б) вывод на экран всех нечетных чисел диапазона; в) вывод на экран всех чисел диапазона, кратных введенному пользователем числу.	Действия выбирать из меню, границы диапазона передавать через параметры, каждое действие оформить в виде отдельной функции.
---	---	---

4. Занести в отчет исходный текст сценария, пример его запуска и результаты работы по каждому пункту меню с подтверждением скриншотами.

```
[pmi-b2704@students lab_2]$ cat script.sh
# Функция для вывода меню
print_menu() {
    echo "Меню:"
    echo "1. Вывести все четные числа в диапазоне"
    echo "2. Вывести все нечетные числа в диапазоне"
    echo "3. Вывести все числа в диапазоне, кратные указанному числу"
    echo "4. Выйти"
}

# Функция для вывода четных чисел в диапазоне
print_even_numbers() {
    for ((i = start; i <= end; i ++)); do
        if ((i % 2 == 0)); then
            echo -n "$i "
        fi
    done
}

# Функция для вывода нечетных чисел в диапазоне
print_odd_numbers() {
    for ((i = start; i <= end; i ++)); do
        if ((i % 2 != 0)); then
            echo -n "$i "
        fi
    done
}

# Функция для вывода чисел, кратных указанному числу
print_numbers_multiple_of() {
    for ((i = start; i <= end; i ++)); do
        if ((i % multiple == 0)); then
            echo -n "$i "
        fi
    done
}

# Проверяем, что переданы все необходимые аргументы через командную строку
if [[ $# -ge 2 ]]; then
    start=$1
    end=$2
# Если аргументов нет, запрашиваем ввод у пользователя
else
    read -p "Введите начало диапазона: " start
    read -p "Введите конец диапазона: " end
fi
```

```

# Проверяем, что введенные значения являются целыми числами
if ! [[ $start =~ ^[0-9]+$ ]] || ! [[ $end =~ ^[0-9]+$ ]]; then
    echo "Ошибка: Начало и конец диапазона должны быть целыми чис-
лами."
    exit 1
fi

# Если передан третий аргумент через командную строку, используем его
if [[ $# -eq 3 ]]; then
    multiple=$3
fi

# Выводим меню
print_menu
read -p "Выберите действие (1-4): " choice

# Обрабатываем выбор пользователя
case $choice in
    1)
        echo "Выбрано действие: Вывести все четные числа в диапазоне"
        print_even_numbers
        ;;
    2)
        echo "Выбрано действие: Вывести все нечетные числа в диапа-
зоне"
        print_odd_numbers
        ;;
    3)
        echo "Выбрано действие: Вывести все числа в диапазоне, крат-
ные указанному числу"
        # Если не указан третий аргумент, запрашиваем его у пользо-
вателя
        if [[ $# -ne 3 ]]; then
            read -p "Введите число, кратное которому будут выводиться
числа: " multiple
        fi
        print_numbers_multiple_of
        ;;
    4)
        echo "До свидания!"
        exit 0
        ;;
    *)
        echo "Ошибка: Некорректный выбор. Пожалуйста, выберите номер
действия из меню (1-4)."
```

exit 1

```

        ;;
    esac
[pmi-b2704@students lab_2]$ ./script.sh
Введите начало диапазона: 1
Введите конец диапазона: 9

```

Меню:

1. Вывести все четные числа в диапазоне
2. Вывести все нечетные числа в диапазоне
3. Вывести все числа в диапазоне, кратные указанному числу
4. Выйти

Выберите действие (1-4): 1

Выбрано действие: Вывести все четные числа в диапазоне

```
2 4 6 8 [pmi-b2704@students lab_2]$ ./script.sh
```

Введите начало диапазона: 11

Введите конец диапазона: 19

Меню:

1. Вывести все четные числа в диапазоне
2. Вывести все нечетные числа в диапазоне
3. Вывести все числа в диапазоне, кратные указанному числу
4. Выйти

Выберите действие (1-4): 2

Выбрано действие: Вывести все нечетные числа в диапазоне

```
11 13 15 17 19 [pmi-b2704@students lab_2]$ ./script.sh
```

Введите начало диапазона: 3

Введите конец диапазона: 39

Меню:

1. Вывести все четные числа в диапазоне
2. Вывести все нечетные числа в диапазоне
3. Вывести все числа в диапазоне, кратные указанному числу
4. Выйти

Выберите действие (1-4): 3

Выбрано действие: Вывести все числа в диапазоне, кратные указанному числу

Введите число, кратное которому будут выводиться числа: 6

```
6 12 18 24 30 36 [pmi-b2704@students lab_2]$
```

5. Вывести результат работы сценария в файл, сравнить содержимое файла с результатами вывода на терминал.

Вывод на терминал

```
[pmi-b2704@students lab_2]$ ./script.sh 1 23 7
```

Меню:

1. Вывести все четные числа в диапазоне
2. Вывести все нечетные числа в диапазоне
3. Вывести все числа в диапазоне, кратные указанному числу
4. Выйти

Выберите действие (1-4): 3

Выбрано действие: Вывести все числа в диапазоне, кратные указанному числу

```
7 14 21
```

Вывод в файл

```
[pmi-b2704@students lab_2]$ ./script.sh 1 23 7 > test.txt
```

Выберите действие (1-4): 3

```
[pmi-b2704@students lab_2]$ cat test.txt
```

Меню:

1. Вывести все четные числа в диапазоне
2. Вывести все нечетные числа в диапазоне
3. Вывести все числа в диапазоне, кратные указанному числу
4. Выйти

Выбрано действие: Вывести все числа в диапазоне, кратные указанному числу

7 14 21

Результаты совпадают.

6. Запустить сценарий с ключами -v и -x, сравнить результаты действия этих ключей.

Вывод с флагом -v

```
[pmi-b2704@students lab_2]$ sh -v script.sh 1 9
module () { eval ` /usr/bin/modulecmd bash $*`
}
# Функция для вывода меню
print_menu() {
    echo "Меню:"
    echo "1. Вывести все четные числа в диапазоне"
    echo "2. Вывести все нечетные числа в диапазоне"
    echo "3. Вывести все числа в диапазоне, кратные указанному числу"
    echo "4. Выйти"
}

# Функция для вывода четных чисел в диапазоне
print_even_numbers() {
    for ((i = start; i <= end; i ++)); do
        if ((i % 2 == 0)); then
            echo -n "$i "
        fi
    done
}

# Функция для вывода нечетных чисел в диапазоне
print_odd_numbers() {
    for ((i = start; i <= end; i ++)); do
        if ((i % 2 != 0)); then
            echo -n "$i "
        fi
    done
}

# Функция для вывода чисел, кратных указанному числу
print_numbers_multiple_of() {
    for ((i = start; i <= end; i ++)); do
        if ((i % multiple == 0)); then
            echo -n "$i "
        fi
    done
}
```

```

done
}

# Проверяем, что переданы все необходимые аргументы через командную строку
if [[ $# -ge 2 ]]; then
    start=$1
    end=$2
# Если аргументов нет, запрашиваем ввод у пользователя
else
    read -p "Введите начало диапазона: " start
    read -p "Введите конец диапазона: " end
fi

# Проверяем, что введенные значения являются целыми числами
if ! [[ $start =~ ^[0-9]+$ ]] || ! [[ $end =~ ^[0-9]+$ ]]; then
    echo "Ошибка: Начало и конец диапазона должны быть целыми числами."
    exit 1
fi

# Если передан третий аргумент через командную строку, используем его
if [[ $# -eq 3 ]]; then
    multiple=$3
fi

# Выводим меню
print_menu
Меню:
1. Вывести все четные числа в диапазоне
2. Вывести все нечетные числа в диапазоне
3. Вывести все числа в диапазоне, кратные указанному числу
4. Выйти
read -p "Выберите действие (1-4): " choice
Выберите действие (1-4):
-----
Выберите действие (1-4): 1

# Обрабатываем выбор пользователя
case $choice in
    1)
        echo "Выбрано действие: Вывести все четные числа в диапазоне"
        print_even_numbers
        ;;
    2)
        echo "Выбрано действие: Вывести все нечетные числа в диапазоне"
        print_odd_numbers
        ;;
    3)

```



```

        echo "Выбрано действие: Вывести все числа в диапазоне, крат-
ные указанному числу"
        # Если не указан третий аргумент, запрашиваем его у пользо-
вателя
        if [[ $# -ne 3 ]]; then
            read -p "Введите число, кратное которому будут выводиться
числа: " multiple
        fi
        print_numbers_multiple_of
        ;;
    4)
        echo "До свидания!"
        exit 0
        ;;
    *)
        echo "Ошибка: Некорректный выбор. Пожалуйста, выберите номер
действия из меню (1-4)."
```

exit 1

```

        ;;
esac
Выбрано действие: Вывести все четные числа в диапазоне
2 4 6 8
```

Вывод с флагом -x

```

[pmi-b2704@students lab_2]$ sh -x script.sh 1 9
+ [[ 2 -ge 2 ]]
+ start=1
+ end=9
+ [[ 1 =~ ^[0-9]+$ ]]
+ [[ 9 =~ ^[0-9]+$ ]]
+ [[ 2 -eq 3 ]]
+ print_menu
+ echo '$'\320\234\320\265\320\275\321\216:'
Меню:
+ echo '1. Вывести все четные числа в диапазоне'
1. Вывести все четные числа в диапазоне
+ echo '2. Вывести все нечетные числа в диапазоне'
2. Вывести все нечетные числа в диапазоне
+ echo '3. Вывести все числа в диапазоне, кратные указанному числу'
3. Вывести все числа в диапазоне, кратные указанному числу
+ echo '4. Выйти'
4. Выйти
+ read -p 'Выберите действие (1-4): ' choice
Выберите действие (1-4): 1
+ case $choice in
+ echo 'Выбрано действие: Вывести все четные числа в диапазоне'
Выбрано действие: Вывести все четные числа в диапазоне
+ print_even_numbers
+ (( i = start ))
```

```

+ (( i <= end ))
+ (( i % 2 == 0 ))
+ (( i ++ ))
+ (( i <= end ))
+ (( i % 2 == 0 ))
+ echo -n '2 '
2 + (( i ++ ))
+ (( i <= end ))
+ (( i % 2 == 0 ))
+ (( i ++ ))
+ (( i <= end ))
+ (( i % 2 == 0 ))
+ echo -n '4 '
4 + (( i ++ ))
+ (( i <= end ))
+ (( i % 2 == 0 ))
+ (( i ++ ))
+ (( i <= end ))
+ (( i % 2 == 0 ))
+ echo -n '6 '
6 + (( i ++ ))
+ (( i <= end ))
+ (( i % 2 == 0 ))
+ (( i ++ ))
+ (( i <= end ))
+ (( i % 2 == 0 ))
+ echo -n '8 '
8 + (( i ++ ))
+ (( i <= end ))
+ (( i % 2 == 0 ))
+ (( i ++ ))
+ (( i <= end ))

```

Сценарий с ключом “-v” выводит отдельно код и выполнение программы без детальной подстановки параметров.

Сценарий с ключом “-x” выводит построчно код и программу и выводит те операторы, которые выполняет программа, делая подстановку символом подчеркивания, где это необходимо и проверку на существование целевого файла.

Контрольные вопросы проработаны.