



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение  
высшего образования*

*«МИРЭА — Российский технологический университет»*

***РТУ МИРЭА***

---

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации  
информационных технологий (МОСИТ)

**Отчет по практической работе № 1**

**по дисциплине**

**«Технология разработки программных приложений»**

**Тема: «Основные команды Git»**

Выполнил студент группы ИКБО-06-21      Бондарь А.Р.

Принял      Туманова М.Б.

Практическая работа выполнена    «\_»\_\_\_\_\_2023г.    *(подпись студента)*

«Зачтено»    «\_»\_\_\_\_\_2023 г.    *(подпись руководителя)*

Москва 2023

# Содержание

Цель практической работы	3
1 Базовые Bash скрипты	4

# Цель практической работы

Данная практическая работа посвящена написанию базовых bash-скриптов.

По сути своей Bash-скриптов представляют из себя ни что иное как последовательность команд командной строки, объединенных в один файл для решения какой-либо задачи.

# Глава 1

## Базовые Bash скрипты

Сценарий, который выводит дату, время, список зарегистрированных пользователей, и uptime системы и сохраняет эту информацию в файл

Данный сценарий принимает параметром имя файла в который будет сохраняться информация. Сам сценарий будет состоять из четырех команд:

- **date** - выводит текущую дату и время;
- **cat** - выводит содержимое файла переданного ей параметром. В скрипте параметром передается файл `/etc/passwd`, где храниться список всех пользователей;
- **cut** - обрезает строки, переданные ей на вход;
- **uptime** - возвращает информацию как долго система была запущена.

Код скрипта показан в листинге 1.1.

Листинг 1.1. Код скрипта

```
#!/bin/sh

date > "$1"
cat /etc/passwd | cut -d ':' -f 1 >> "$1"
uptime >> "$1"
echo " " >> "$1"
```

## Сценарий, который выводит содержимое любого каталога или сообщение о том, что его не существует

Для реализации данного скрипта используется управляющая конструкция `if-then-else`. В условии используется команда `test`, ее сокращенная форма с квадратными скобками, и флаг `-d` для проверки файла, является ли он директорией.

Код скрипта показан в листинге 1.6.

### Листинг 1.2. Проверка на директорию

```
#!/bin/sh

if [ -d "$1" ]; then
    ls "$1"
else
    echo "it_is_not_dir"
fi
```

## Сценарий, который с помощью цикла прочитает файл и выведет его содержимое

Чтобы построчно прочитать файл использовался цикл `while` и в качестве условия выступал результат команды `read line`, которая читает строки из стандартного потока. Чтобы чтение происходило из файла, перенаправили стандартный поток ввода в файл, переданный параметром.

Код скрипта показан в листинге 1.3.

### Листинг 1.3. Вывод содержимого файла

```
#!/bin/sh

while read line
do
    echo "$line"
done < "$1"
```

Сценарий, который с помощью цикла выведет список файлов и директорий из текущего каталога, укажет, что есть файл, а что директория

Листинг 1.4. Проверка на директорию

```
#!/bin/sh

ls "$1" | while read file
do
    echo -n "$file"
    if [ -f "$file" ]; then
        echo "_is_a_file"
    else
        echo "_is_a_dir"
    fi
done;
```

Сценарий, который подсчитает объем диска, занимаемого директорией

Листинг 1.5. Проверка на директорию

```
#!/bin/sh

du -sh "$1"
```

Сценарий, который выведет список всех исполняемых файлов в директории, для которых у текущего пользователя есть права на исполнение

Листинг 1.6. Проверка на директорию

```
#!/bin/sh

ls "$1" | while read file
```

```
do
    if [ -x "$file" ]; then
        echo "$file"
    fi
done;
```