

# Лекция №6: Bash и пакетные менеджеры

## Bash

### Что это такое?

- Bash — сценарии оболочки bash.

### Зачем?

- Простая автоматизация действий и скриптов.
- Доступен где есть оболочка.
- Простая интеграция с другими приложениями.

Bash-скрипт — это файл, содержащий последовательность команд, которые выполняются программой bash построчно. Он позволяет выполнять ряд действий, таких как переход к определённому каталогу, создание папки и запуск процесса с помощью командной строки.

Сохранив эти команды в сценарии (скрипты), вы можете повторять одну и ту же последовательность шагов несколько раз и выполнять их, запустив сценарий.

## Hello World на bash

Пример скрипта:

```
1 #!/bin/bash
2 echo "Hello, World!"
```

Шебанг `#!/bin/sh` не обязателен — он может быть изменён под нужную программу, например:

```
1 #!/usr/bin/env python
```

Такой шебанг использует утилиту `env`, которая ищет исполняемый файл в переменной окружения `PATH`. Плюс — гибкость пути. Минус — первый попавшийся файл может не подходить.

## Переменные

Присваивание переменных:

```
1 name="Alex"
2 echo $name
3 echo ${name}
```

## Особые переменные

- `$@` — все аргументы скрипта.
- `$i` — *i*-й аргумент скрипта (нумерация с 1).

## Литералы

Оболочка выполняет подстановку переменных, шаблонов и т.д. перед выполнением команды. Кавычки позволяют создавать литералы — строки, которые не анализируются оболочкой.

- Одинарные кавычки (`'`): всё внутри интерпретируется буквально.

Пример:

```
1 grep 'r.*t' /etc/passwd
```

## Математические операции

Используется конструкция `$((...))`:

```
1 result=$((5 + 3))
2 echo $result \# 8
```

## Управление потоком

```
1 if [ "$1" = "hi" ]; then
2     echo "Hi!"
3 else
4     echo "Bye!"
5 fi
```

[ — это отдельная команда (синоним `test`), а не часть синтаксиса.

## Логические выражения

### Операторы для числового сравнения

- `-eq` — равно
- `-ne` — не равно
- `-lt` — меньше
- `-le` — меньше или равно
- `-gt` — больше
- `-ge` — больше или равно

## Операторы проверки прав доступа

- `-r` — файл доступен на чтение
- `-w` — файл доступен на запись
- `-x` — файл исполняемый

## Операторы проверки типов файлов

- `-f` — обычный файл
- `-d` — каталог

## Циклы

Пример цикла `for`:

```
1 for i in {1..5}; do
2     echo "Iteration $i"
3 done
```

Пример цикла `while`:

```
1 count=1
2 while [ $count -le 5 ]; do
3     echo $count
4     count=$((count + 1))
5 done
```

## Функции

```
1 greet() {
2     echo "Hi, $1!"
3 }
4 greet "Ann"
```

## Пример использования `awk`

Пример фильтрации данных:

```
1 ps aux | awk '{print $1, $2, $11}' | head -5
```

## Установка пакетов

### Что такое пакет?

- Пакет — набор команд и зависимостей для установки ПО.
- `.deb` — Debian и производные (Ubuntu, Mint и др.)
- `.rpm` — Red Hat и производные (CentOS, Fedora, OpenSUSE)
- `.apk` — Android
- `.ebuild` — Gentoo

## Пакетные менеджеры

- APT
- RPM
- YUM
- DNF (Dandified YUM)
- Pacman

## Что такое репозиторий?

Репозиторий (от англ. *repository* — хранилище) — место, где хранятся и поддерживаются данные, чаще всего в виде файлов, доступных для распространения по сети.

Официальные репозитории есть у большинства дистрибутивов Linux. Также можно добавлять сторонние.