**Лабораторная работа №7. Bash-скрипты**

**Целью** лабораторной работы является получение практических навыков по написанию Bash-скриптов для ОС Linux.

**Задачи**:

1. Самостоятельно изучить синтаксис и важнейшие структуры Bash-скриптов.

2. Научиться применять Bash-скрипты для администрирования системы.

3. Закрепить полученные в ходе выполнения лабораторной работы навыки.

**Краткие теоретические сведения**

**Bash** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***B****ourne****a****gain****sh****ell*, [каламбур](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BB%D0%B0%D0%BC%D0%B1%D1%83%D1%80) *«Born again» shell* — «возрождённый» shell) — усовершенствованная и модернизированная вариация [командной оболочки](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BE%D0%BB%D0%BE%D1%87%D0%BA%D0%B0_UNIX) [Bourne shell](https://ru.wikipedia.org/wiki/Bourne_shell" \o "Bourne shell). Одна из наиболее популярных современных разновидностей командной оболочки [UNIX](https://ru.wikipedia.org/wiki/UNIX). Особенно популярна в среде [Linux](https://ru.wikipedia.org/wiki/Linux" \o "Linux), где она часто используется в качестве предустановленной командной оболочки.

Представляет собой командный процессор, работающий, как правило, в интерактивном режиме в текстовом окне. Bash также может читать команды из файла, который называется [*скриптом*](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) (или *сценарием*). Как и все Unix-оболочки, он поддерживает автодополнение имён файлов и каталогов, подстановку вывода результата команд, переменные, контроль над порядком выполнения, операторы ветвления и цикла. Ключевые слова, синтаксис и другие основные особенности языка были заимствованы из [sh](https://ru.wikipedia.org/wiki/Bourne_shell" \o "Bourne shell). Другие функции, например, история, были скопированы из [csh](https://ru.wikipedia.org/wiki/Csh" \o "Csh) и [ksh](https://ru.wikipedia.org/wiki/Ksh" \o "Ksh). Bash в основном соответствует стандарту [POSIX](https://ru.wikipedia.org/wiki/POSIX), но с рядом расширений[[4]](https://ru.wikipedia.org/wiki/Bash#cite_note-4).

Название «bash» является [акронимом](https://ru.wikipedia.org/wiki/%D0%90%D0%BA%D1%80%D0%BE%D0%BD%D0%B8%D0%BC) от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Bourne-again-shell* («ещё-одна-командная-оболочка-Борна») и представляет собой игру слов: Bourne-shell — одна из популярных разновидностей командной оболочки для UNIX (sh), автором которой является [Стивен Борн](https://ru.wikipedia.org/wiki/%D0%91%D0%BE%D1%80%D0%BD,_%D0%A1%D1%82%D0%B8%D0%B2%D0%B5%D0%BD) ([1978](https://ru.wikipedia.org/wiki/1978)), усовершенствована в [1987 году](https://ru.wikipedia.org/wiki/1987_%D0%B3%D0%BE%D0%B4) [Брайаном Фоксом](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D0%BA%D1%81,_%D0%91%D1%80%D0%B0%D0%B9%D0%B0%D0%BD). Фамилия Bourne (Борн) перекликается с английским словом *born*, означающим «родившийся», отсюда: рождённая-вновь-командная оболочка.

В сентябре 2014 года в bash была обнаружена широко эксплуатируемая уязвимость [Bashdoor](https://ru.wikipedia.org/wiki/Bashdoor" \o "Bashdoor).

**Отличия в синтаксисе**

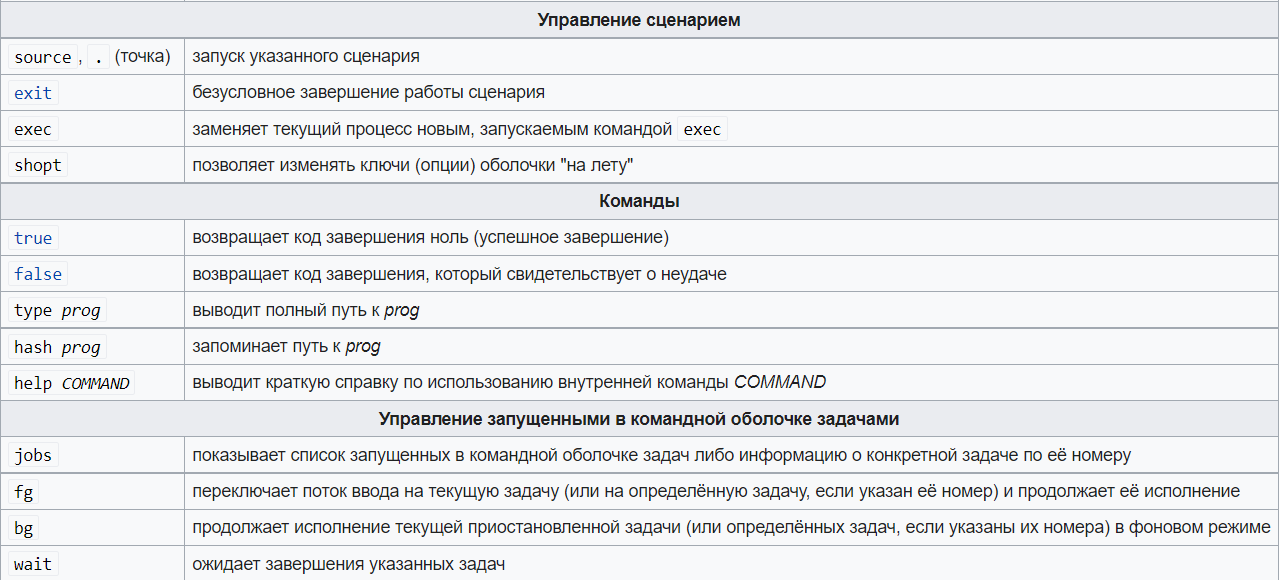
Подавляющее большинство важных скриптов командного процессора Bourne может выполняться без изменения в bash, за исключением тех, которые ссылаются на специальные переменные Bourne или используют встроенные команды Bourne. Синтаксис команд Bash включает идеи, заимствованные у Korn shell ([ksh](https://ru.wikipedia.org/wiki/Ksh" \o "Ksh)) и C shell ([csh](https://ru.wikipedia.org/wiki/Csh" \o "Csh)), такие как редактирование командной строки, история команд, стек каталогов, переменные $RANDOM и $PPID, синтаксис замены команды $(…). Когда Bash используется как интерактивный командный процессор, он поддерживает автозавершение имён программ, файлов, переменных и т. п. с помощью клавиши Tab ↹.

**Внутренние команды**

Интерпретатор bash имеет множество встроенных команд, часть из которых имеет аналогичные исполняемые файлы в операционной системе. Однако следует обратить внимание, что чаще всего для встроенных команд отсутствуют [man-страницы](https://ru.wikipedia.org/wiki/Man" \o "Man), а при попытке просмотра справки по встроенной команде на самом деле будет выдаваться справка по исполняемому файлу. Исполняемый файл и встроенная команда могут различаться параметрами. Информация по встроенным командам расписана в справочной странице bash:

man bash





**Скрипты**

В простейшем случае, [скрипт](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) — простой список команд, записанный в файл. Командный процессор должен знать, что он должен этот файл обработать, а не просто прочесть его содержимое. Для этого служит специальная конструкция, называемая [shebang](https://ru.wikipedia.org/wiki/Shebang" \o "Shebang): #!. Символ # задаёт комментарий, но в данном случае shebang означает, что после этого спецсимвола находится путь к интерпретатору для исполнения сценария.

**Синтаксис**

Синтаксис команд bash — это расширенный синтаксис команд Bourne shell. Окончательная спецификация синтаксиса команд bash есть в Bash Reference Manual, распространяемом проектом [GNU](https://ru.wikipedia.org/wiki/GNU).

«Hello world»

*#!/usr/bin/env bash*

echo 'Hello World!'

Этот скрипт содержит только две строки. Первая строка сообщает системе о том, какая программа используется для запуска файла. Вторая строка — это единственное действие, которое выполняется этим скриптом, он, собственно, печатает «Hello world!» в терминале.

**Запуск скрипта**

Для того, чтобы скрипт стал исполняемым, могут быть использованы следующие команды:

chmod +rx scriptname *# выдача прав на чтение/исполнение любому пользователю*

chmod u+rx scriptname *# выдача прав на чтение/исполнение только "владельцу" скрипта*

Из соображений безопасности путь к текущему каталогу . не включён в переменную окружения $PATH. Поэтому для запуска скрипта необходимо явно указывать путь к текущему каталогу, в котором находится скрипт:

./scriptname

Кроме того, передать такой файл на исполнение [интерпретатору](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D1%80%D0%B5%D1%82%D0%B0%D1%82%D0%BE%D1%80_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B9_%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8) Bash можно и явно, используя команду bash:

bash scriptname

В этом случае не требуется ни установка прав доступа, ни использование последовательности #! в коде.

**Перенаправление ввода-вывода**

В bash есть встроенные [файловые дескрипторы](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B4%D0%B5%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%80): 0 (stdin), 1 (stdout), 2 (stderr).

* stdin — стандартный ввод — то, что набирает пользователь в консоли;
* stdout — стандартный вывод программы;
* stderr — стандартный вывод ошибок.

Для операций с этими и пользовательскими дескрипторами существуют специальные символы: > (перенаправление вывода), < (перенаправление ввода). Символы &, - могут предварять номер дескриптора; например, 2>&1 — перенаправление дескриптора 2 (stderr) в дескриптор 1 (stdout).

*Более подробно теоретическая информация расписана в файлах в приложенном к работе архиве*

**Задание:**

**Задача 1**

Написать скрипт на Bash, который должен принимать в качестве аргумента любую строку и удалять из /home/”user” файл с именем, соответствующим переданной строке без необходимости отвечать утвердительно на вопросы системы. Так же, скрипту можно передать параметр «\*» (удалит все файлы) и «-» кроме файлов имена который начинаются с параметра после «-». Т.е

$del.sh file # удалит файл file

$del.sh \* -file # удалит все, кроме файла file

**Задача 2**

Написать скрипт, который из /etc/passwd выведет домашнюю директорию для пользователя заданного в параметре скрипта, пример файла /etc/passwd:

sync:x:4:65534:sync:/bin:/bin/sync

games:x:5:60:games:/usr/games:/bin/sh

man:x:6:12:man:/var/cache/man:/bin/sh

lp:x:7:7:lp:/var/spool/lpd:/bin/sh

mail:x:8:8:mail:/var/mail:/bin/sh

news:x:9:9:news:/var/spool/news:/bin/sh

uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh

proxy:x:13:13:proxy:/bin:/bin/sh

www-data:x:33:33:www-data:/var/www:/bin/sh

backup:x:34:34:backup:/var/backups:/bin/sh

list:x:38:38:Mailing List Manager:/var/list:/bin/sh irc:x:39:39:ircd:/var/run/ircd:/bin/sh

gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh nobody:x:65534:65534:nobody:/nonexistent:/bin/sh

libuuid:x:100:101::/var/lib/libuuid:/bin/sh

syslog:x:101:103::/home/syslog:/bin/false

messagebus:x:102:105::/var/run/dbus:/bin/false

**Задача 3**

Написать скрипт, который выведет top 5 процессов, которые потребляют больше всего памяти и процессора в системе.

**Задача 4**

Написать скрипт, который выведет всех потомков процесса по его PID.

**Задача 5**

Написать скрипт, который для всех файлов во временной директории, которые были созданы за текущий месяц, и не изменялись в течение недели при нахождении в них строки test поменяет ее на tset.

**Задача 6**

Написать скрипт, которые выведет на экран все файлы, размер которых больше 50 Мб в директории /var/log

**Задача 7**

Написать скрипт, который в качестве первого параметра принимает команду, которую нужно выполнить, а в качестве остальных параметров адреса серверов, на которых эта команда должна выполниться.