# 2. ЛАБОРАТОРНЫЕ РАБОТЫ

# ЛАБОРАТОРНАЯ РАБОТА №1 УПРАВЛЕНИЕ ОС LINUX, ИНТЕРПРЕТАТОР BASH

Цель работы – изучить основные объекты, команды, типы данных и операторы управления интерпретатора BASH; создать скрипт-файл.

**Теоретическая часть**

***Bash*** - это ***sh***-совместимый интерпретатор командного языка, выполняющий команды, прочитанные со стандартного входного потока или из файла. ***Скрипт-файл*** – это обычный текстовый файл, содержащий последовательность команд ***bash***, для которого установлены права на выполнение. Пример скрипта, выводящего содержимое текущего каталога на консоль и в файл:

***#!/bin/bash***

***dir***

***dir > 1.txt***

Следующие переменные используются командным интерпретатором.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***$0, $1, $2, $3…*** | | | Значения аргументов командной строки при запуске скрипта. Где ***$0***-имя самого файла скрипта, ***$1***- первый аргумент, ***$2***- второй аргумент, и т.д. | |  |
| ***$@*** | | | Все аргументы командной строки начиная с 1-го | |  |
| ***$?*** | Код возврата последней команды | | | | |
| ***$\**** | Все аргументы командной строки начиная с 1-го | | | | |
| ***$#***­­ | | Число переданных аргументов | |  |  |

Пример простого скрипта, выводящего на консоль и в файл содержимое каталога, где имя каталога передаётся скрипту в качестве аргументов при запуске:

Запуск скрипта: ***./home/stud 1.sh***

Скрипт:

***#!/bin/bash***

***dir $1***

***dir $1 > 1.txt***

Можно создать собственную переменную и присвоить ей значение:

***A=121***

***A=”121”***

***let A=121***

***let “A=А+1”***

Вывод значения на консоль: ***echo $A***

Организация циклов:

1. ***for var1 in list***

***do***

***…***

***done***

2. ***while exp***

***…***

***end***

3. ***until exp // аналог do-while***

***do***

***…***

***done***

Проверка условия: ***test[expr]***

где ***expr***: а) для строк: ***S1 = S2*** ***S1***содержит ***S2***

***S1 != S2******S1***не содержит ***S2***

***-n S1*** если длина ***S1 >0***

***-z S1*** если длина ***S1 =0***

б) целые ***i1***и ***i2***

***i1 – ge i2***

***i1 – gt i2***

***i1 – ie i2***

***i1 – et i2***

***i1 – nt i2***

в) файлы

***-d name\_file*** является ли файл каталогом

***-f name\_file*** является ли файл обычным файлом

***-r name\_file*** доступен ли файл для чтения

***-s name\_file*** имеет ли файл ненулевую длину

***-w name\_file*** доступен ли файл для записи

***-x name\_file*** является ли файл исполняемым

г) логически операции

***!exp*** логическое отрицание (не)

***exp1 –a exp2*** умножение условий (и)

***exp1 –o exp2*** сложение условий (или)

Проверка условия: ***if [expr ]***

***then com 1*** Если условие ***expr=true*** то команда

***…***

***(elif expr2***

***…***

***)***

***else***

***…***

***fi***

Проверка нескольких условий: ***case string1 in***

***str 1)***

***…***

***; ;***

***str 2)***

***…***

***; ;***

***str 3)***

***…***

***; ;***

***\*)***  // default

***…***

***; ;***

***esac***

Функция пользователя: ***fname2 (arg1,arg2...argN)***

***{  
commands  
}***

***if [ -d “/home” ]***

***then***

***echo "Yes"***

***else***

***echo "No"***

***fi***

**Общие правила выполнения работы**

1. Все параметры для скрипта задаются как аргументы командой строки
2. Вывод результатов должен быть по формату:

***полный путь к каталогу(файлу) остальные параметры через пробел.***

1. Все размеры каталогов(файлов) только в байтах

Примеры вывода результатов:

***/home/student/1.txt 128 76***

***/home/student/2.txt 4576 35***

***/home/student/3.txt 128 76***

***/home/student/4.txt 128 76***

***/home/student/lab1/1.txt 128 76***

***/home/student/lab2/1.txt 128 76***

***/home/student/lab3/1.txt 128 76***

***…***

***3278***

**Не выводить никаких дополнительных слов типа размер, байт и т.д.**

1. Должен быть контроль ошибок в скрипте. Все сообщения (системные) об ошибках необходимо дополнить именем скрипта в начале сообщения.

Пример вывода:

***> 1.sh: (системное сообщение об ошибке)***

1. Все скрипты проверяются на каталогах: ***/etc, / usr***

## Варианты индивидуальных заданий

1. Написать скрипт для поиска файлов заданного размера в заданном каталоге (имя каталога задаётся пользователем в качестве третьего аргумента командной строки). Диапазон (мин.- мах.) размеров файлов задаётся пользователем в качестве первого и второго аргумента командной строки. Вывести на консоль, используя цикл ***for in***, первые найденные 20 файлов в виде: полный путь, имя файла, его размер.
2. Написать скрипт с использованием цикла ***for in***, выводящий на консоль полный путь с именем файла, размеры и права доступа для всех файлов, кроме файлов с заданным расширением (второй аргумент командной строки, в заданном каталоге и всех его подкаталогах (имя каталога задается пользователем в качестве первого аргумента командной строки). На консоль в конце выводится общее число просмотренных файлов.
3. Написать скрипт для поиска заданной пользователем строки во всех файлах заданного каталога и всех его подкаталогов (строка и имя каталога задаются пользователем в качестве первого и второго аргумента командной строки). На консоль выводятся полный путь и имена файлов, в содержимом которых присутствует заданная строка, и их размер. Если к какому-либо каталогу нет доступа, необходимо вывести соответствующее сообщение и продолжить выполнение.
4. Написать скрипт поиска одинаковых по содержимому файлов в двух каталогов и их подкаталогах, например, ***Dir1*** и ***Dir2***. Пользователь задаёт имена ***Dir1*** и ***Dir2*** в качестве первого и второго аргумента командной строки. В результате работы программы файлы, имеющиеся в ***Dir1***, сравниваются с файлами в ***Dir2*** по их содержимому. На экран выводятся число просмотренных файлов и результаты сравнения.
5. Написать скрипт, находящий в заданном каталоге и всех его подкаталогах каждый второй файл, владельцем которых является заданный пользователь. Имя владельца и каталог задаются пользователем в качестве первого и второго аргумента командной строки. Скрипт выводит результаты в виде: полный путь, имя файла, его размер. В конце выводится общее число просмотренных файлов.
6. Написать скрипт, находящий в заданном каталоге (первый аргумент командной строки) и всех его подкаталогах все файлы заданного размера. Диапазон (мин.- мах.) размеров файлов задаётся пользователем в качестве второго и третьего аргумента командной строки. Скрипт выводит результаты поиска в файл (четвёртый аргумент командной строки) в виде: полный путь, имя файла, его размер. На консоль выводится общее число просмотренных файлов.
7. Написать скрипт, подсчитывающий суммарный размер файлов в заданном каталоге и отдельно во всех его подкаталогах (имя каталога задаётся пользователем в качестве аргумента командной строки). Скрипт выводит результаты подсчёта в файл (второй аргумент командной строки) в виде подкаталог (полный путь), суммарный размер файлов, число файлов в данном подкаталоге.
8. Написать скрипт, находящий в заданном каталоге (первый аргумент командной строки) и всех его подкаталогах все файлы заданного расширения и создающий для каждого найденного файла жесткую ссылку в заданном каталоге (второй аргумент командной строки). Расширение файла — это третий аргумент командной строки.
9. Написать скрипт, находящий на заданной вложенности глубине (аргумент 2 командной строки) все каталоги и подкаталоги начиная с заданного каталога (аргумент 1 командной строки). Скрипт выводит результаты в файл (третий аргумент командной строки) в виде полный путь, количество файлов в каталоге. На консоль выводится общее число просмотренных каталогов.
10. Написать скрипт, находящий все дубликаты (с одинаковым содержимым) файлов в заданном диапазоне размеров от ***N1*** до ***N2*** (***N1, N2*** задаются в аргументах командной строки), начиная с исходного каталога и ниже. Имя исходного каталога задаётся пользователем в качестве первого аргумента командной строки.
11. Написать скрипт, считающий для заданного каталога и всех его подкаталогов суммарный размер и количество файлов в заданном диапазоне размеров файлов (имя каталога задаётся пользователем в качестве первого аргумента командной строки, 2,3 аргументы командной строки диапазон размеров). Подсчет ведется для каждого подкаталога отдельно. Результаты выводятся на консоль в виде: каталог (полный путь), количество файлов, суммарный размер.
12. Написать скрипт, считающий для заданного каталога (имя каталога задаётся пользователем в качестве первого аргумента командной строки) и всех его подкаталогов количество файлов, имеющих в своем имени три символа в любом месте имени (второй аргумент командной строки) и суммарный размер таких файлов. Подсчет ведется для каждого подкаталога отдельно. Результаты выводятся на консоль в виде: каталог (полный путь), количество файлов, суммарный размер.
13. Написать скрипт, считающий для заданного каталога (имя каталога задаётся пользователем в качестве первого аргумента командной строки) и всех его подкаталогов количество файлов, имеющих длину имени в ***N*** символов (второй аргумент командной строки) и суммарный размер таких файлов. Подсчет ведется для каждого подкаталога отдельно. Результаты выводятся на консоль в виде: (каталог полный путь), количество файлов, суммарный размер.
14. Написать скрипт, считающий для заданного каталога (имя каталога задаётся пользователем в качестве первого аргумента командной строки) и всех его подкаталогов количество файлов, расположенных на заданном устройстве(***device***) (второй аргумент командной строки) и суммарный размер таких файлов для каждого устройства. Подсчет ведется для каждого подкаталога отдельно. Результаты выводятся на консоль в виде: (каталог полный путь), устройство, количество файлов, суммарный размер.
15. Написать скрипт, находящий в каталоге /proc все процессы, запущенные заданным пользователем (***uid*** пользователя задаётся в качестве первого аргумента командной строки). Вывести количество открытых этим процессом файлов. Результаты выводятся на консоль в виде: ***pid*** процесса, количество файлов. Нельзя использовать команды ps, top и им подобные. Информация о ***uid*** пользователя для запущенного процесса (программы) находится в файле /proc/pid/status (в качестве ***uid*** пользователя можно брать первое значение в строке:” ***Uid: \*\*\* \*\*\* \*\*\* \*\*\*“***). Файлы, открытые процессом (программой) –в /proc/pid/fd/.
16. Написать скрипт, считающий для заданного каталога (имя каталога задаётся пользователем в качестве первого аргумента командной строки) и всех его подкаталогов количество файлов, расположенных на устройстве (***device: sda1-sda8***) и суммарный размер таких файлов для каждого устройства. Подсчет ведется для каждого подкаталога отдельно. Результаты выводятся на консоль в виде каталог полный путь, устройство, количество файлов, суммарный размер.
17. Написать скрипт, считающий для заданного пользователя (имя пользователя задаётся в качестве первого аргумента командной строки) во всех подкаталогах, принадлежащих этому пользователю количество файлов, расположенных на заданном устройстве (***device***) (второй аргумент командной строки) и суммарный размер таких файлов. Подсчет ведется для каждого подкаталога отдельно. Результаты выводятся на консоль в виде: каталог (полный путь), устройство, количество файлов, суммарный размер.
18. Написать скрипт, находящий 2 подкаталога (имя начального каталога задаётся пользователем в качестве первого аргумента командной строки) с максимальным суммарным размером файлов и максимальным количеством файлов, Подсчет ведется для каждого подкаталога отдельно. Результаты выводятся на консоль в виде: каталог (полный путь), количество файлов, суммарный размер.
19. Написать скрипт, находящий в каталоге /proc заданный запущенный процесс(программу) по его имени (имя процесса задаётся пользователем в качестве первого аргумента командной строки). Вывести количество открытых этим процессом файлов. Результаты выводятся на консоль в виде: имя процесса, pid, количество файлов. Нельзя использовать команды ps, top и им подобные. Строка с командой запуска процесса (программы) находится в /proc/pid/cmdline. Файлы, открытые процессом (программой) –в /proc/pid/fd/.