**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.

**Формулировка задания:**

**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.

**Листинг:**

**first.sh:**

#!/bin/bash

if [[ $(whoami) == "beer-h4nter" || $(whoami) == "root" ]]; then

echo "ТЫ ДУРАК?!?!??!?!?"

exit 1

fi

USER\_HOME="/home/$(whoami)"

if [[ $2 == "\*" && $3 == "-" ]]; then

if [[ $# -lt 4 ]]; then

echo "Не указаны исключённые файлы."

exit 1

fi

excluded\_files=("${@:4}")

excluded\_files+=("${1}")

for excluded\_file in "${excluded\_files[@]}"; do

if [[ ! -f "${USER\_HOME}/${excluded\_file}" ]]; then

echo "Файл ${excluded\_file} не существует."

exit 1

fi

done

echo "Удаление всех файлов в ${USER\_HOME} кроме ${excluded\_files[@]}."

for file in "${USER\_HOME}"/\*; do

if ! [[ -f $file ]]; then

continue

fi

is\_excluded="false"

for excluded\_file in "${excluded\_files[@]}"; do

if [[ ${excluded\_file} == $(basename ${file}) ]]; then

is\_excluded="true"

continue

fi

done

if ! $is\_excluded; then

rm -f "${file}"

fi

done

echo "Все файлы удалены."

elif [[ "${#}" -eq 2 && -f $2 ]]; then

if [[ "${1}" != "${2}" ]]; then

echo "Удаление файла ${2}..."

rm -f "${2}"

echo "Файл ${2} удалён."

else

echo "Файл не удалён".

fi

else

echo "Неправильный формат команды."

fi

**second.sh:**

#!/bin/bash

if [[ ${#} -ne 1 ]]; then

echo "Пользователь не был указан"

exit 1

fi

USER\_DATA=$(cat /etc/passwd | grep "${1}")

IFS=':' read -r -a USER\_DIR <<< "$USER\_DATA"

echo "Домашняя дикертория пользователя ${1}:"

echo "$(ls -a /${USER\_DIR[5]})"

**third.sh:**

#!/bin/bash

echo "Топ 5 процессов по нагрузке на процессор:"

echo "$(ps -eo pid,comm,%cpu,%mem --sort=-%cpu | head -6)"

echo ""

echo "Топ 5 процессов по занимаемой памяти:"

echo "$(ps -eo pid,comm,%cpu,%mem --sort=-%mem | head -6)"

fourth.sh:

#!/bin/sh

if [[ ${#} -eq 1 ]]; then

echo "Необходимое количество аргументов: 1"

exit 1

fi

ps --ppid $1

**fiveth.sh:**

#!/bin/sh

TEMP\_DIR="/home/beer-h4nter/"

CURRENT\_MONTH=$(date +"%Y-%m")

ONE\_WEEK\_AGO=$(date -d '7 days ago' +%s)

find "$TEMP\_DIR" -type f -newermt "$CURRENT\_MONTH-01" ! -newermt "$CURRENT\_MONTH-31" | while read -r file; do

last\_modified=$(stat -c %Y "$file")

if [[ $last\_modified -le $ONE\_WEEK\_AGO ]]; then

if grep -q "test" "$file"; then

sed -i 's/test/tset/g' "$file"

echo "Modified: $file"

fi

fi

done

**sixth.sh:**

#!/bin/sh

DIR="/var/log"

for file in "$DIR"/\*; do

if [[ $(stat --format="%s" ${file}) -ge 51200 ]]; then

echo $file

fi

done

**seventh.sh:**

#!/bin/bash

if [ "$#" -lt 2 ]; then

echo "Количество аргументов должно быть больше 1."

exit 1

fi

SERVER="$1"

shift

COMMAND\_LIST=""

for COMMAND in "$@"; do

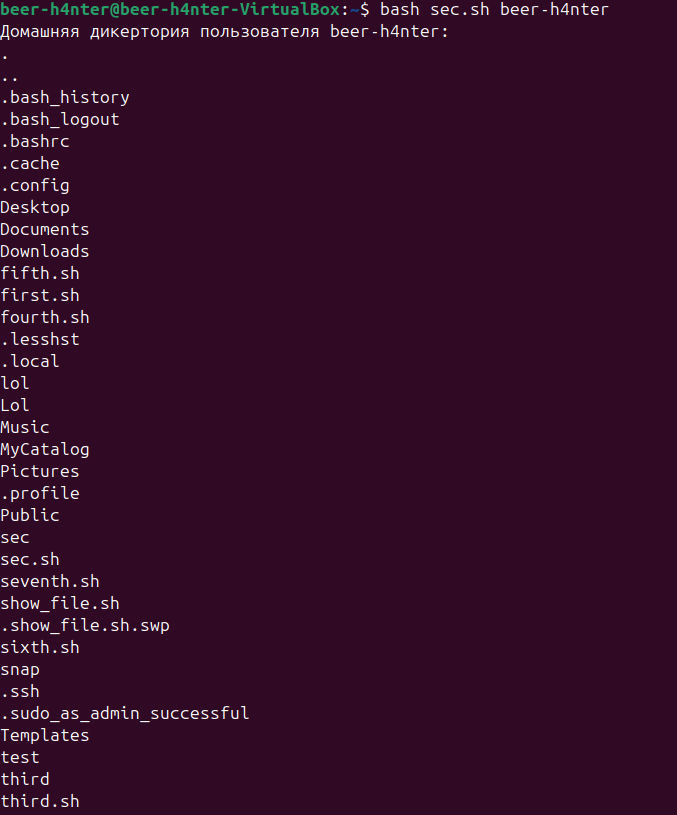
COMMAND\_LIST+="${COMMAND}; "

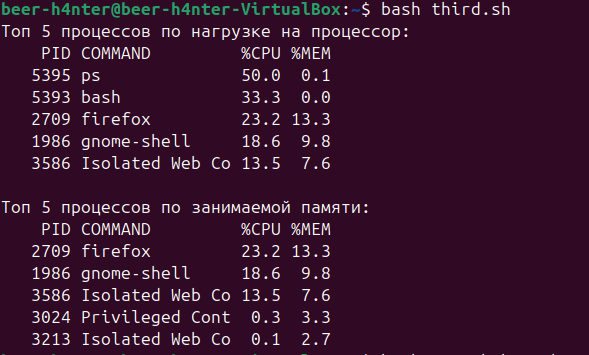
done

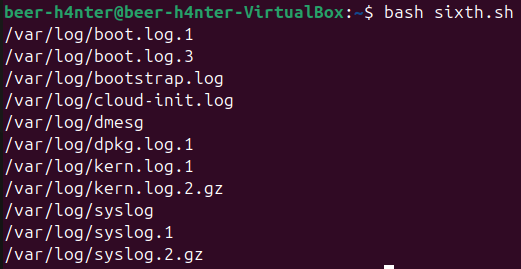
echo "Выполнение команды на сервере: $SERVER"

ssh "$SERVER" "$COMMAND\_LIST"

**Результаты выполнения команды:**

****

****

****

**Вывод:** в ходе лабораторной работы были реализованы 7 скриптов с использованием bash.