

EPAM University Programs  
DevOps external course  
Module 4 Linux Essentials with Bash

TASK 4.11

All scripts are available on my Git in 4.11\_task Dir

4.11.1 Найти и вывести пути до файлов из каталога /usr (включая подкаталоги), размер которых больше 20 Мб.

`find /usr -type f -size +20M -exec ls -hsSr {} +`

```
192.168.1.99 - PuTTY
root@SmartHome:~# find /usr -type f -size +20M -exec ls -hsSr {} +
26M /usr/lib/arm-linux-gnueabi/libcudata.so.63.1
30M /usr/libexec/docker/cli-plugins/docker-app
32M /usr/bin/containerd
33M /usr/libexec/docker/cli-plugins/docker-buildx
57M /usr/bin/docker
66M /usr/bin/dockerd
root@SmartHome:~# uname -a
Linux SmartHome 4.19.97+ #1294 Thu Jan 30 13:10:54 GMT 2020 armv6l GNU/Linux
root@SmartHome:~#
```

4.11.2 Подсчитать количество файлов, количество скрытых файлов в домашнем каталоге текущего пользователя и вывести результат в формате:

Домашний каталог пользователя  
<User>  
содержит обычных файлов:  
XX  
скрытых файлов:  
YY

`number_of_files.sh`

basic script commands

```
192.168.1.99 - PuTTY
root@SmartHome:~# find ~ -type f -not -iname ".*" | wc -l
1115
root@SmartHome:~# find ~ -type f -iname ".*" | wc -l
16
root@SmartHome:~# find ~ -type f | wc -l
1131
root@SmartHome:~#
```

Script result:

```
192.168.1.99 - PuTTY
root@SmartHome:~# /tmp/4.11_task/number_of_files.sh
Homedir /root of user:
root
consist of
regular files:
1115
Hidden files:
16
root@SmartHome:~#
```

4.11.3 Вывести на экран дату, время, список зарегистрировавшихся пользователей, uptime системы.

Simple command - no script needed:

date && w && uptime

Result:

```
192.168.1.99 - PuTTY
root@SmartHome:~# date && w && uptime
Sat 9 May 17:23:57 EEST 2020
17:23:58 up 7:32, 2 users, load average: 0.07, 0.23, 0.25
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
root      pts/0    192.168.1.131   16:52   48.00s 0.95s  0.95s -bash
root      pts/1    192.168.1.131   17:07    6.00s 0.67s  0.07s w
17:23:58 up 7:32, 2 users, load average: 0.07, 0.23, 0.25
root@SmartHome:~#
```

4.11.4 Вывести количество процессов, запущенных от имени текущего и от имени пользователя root в формате:

Процессов пользователя:

<User>

XX

Процессов пользователя root:

YY

processes.sh

basic script commands:

```
pi@SmartHome: ~
root      pts/0    192.168.1.131   16:52   48.00s 0.95s  0.95s -bash
root      pts/1    192.168.1.131   17:07    6.00s 0.67s  0.07s w
17:23:58 up 7:32, 2 users, load average: 0.07, 0.23, 0.25
root@SmartHome:~# su - pi
pi@SmartHome:~$ ps -aux | grep -c ^$USER
4
pi@SmartHome:~$ ps -aux | grep -c ^root
113
pi@SmartHome:~$
```

### Script result:

```
pi@SmartHome: ~  
pi@SmartHome:~ $ /tmp/4.11_task/processes.sh  
Number of processes for user:  
pi  
4  
Number of processes for user root:  
111  
pi@SmartHome:~ $
```

4.11.5 Найти и вывести 5 процессов, потребляющих больше всего памяти в системе.

### Simple command - no script needed:

`ps aux --sort -pmem | head -5`

### Result:

```
pi@SmartHome:~ $ ps aux --sort -pmem | head -5  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root      3339  4.6  15.6 148512 76896 ?        Ssl   09:54   22:01 python3 -m homeassistant --config /config  
root       544  0.7   7.4 993724 36740 ?        Ssl   09:51    3:46 /usr/bin/dockerd -H fd:// --containerd=/run/cont  
ainerd/containerd.sock  
root      1468  1.5   6.3 88444 31020 ?        Ssl   09:53    7:19 python3 -m supervisor  
root      1039  0.0   4.9 897920 24504 ?        Sl    09:52    0:19 docker start --attach hassio_supervisor  
pi@SmartHome:~ $
```

4.11.6 Разработать сценарий, который вызывает команду `grep` и принимает следующие аргументы:

- текст, который нужно найти;
- файл, в котором нужно найти этот текст;
- максимальное количество строк, которое нужно вывести на экран.

Вывод команды `grep` отсортировать и пронумеровать. Сделать для скрипта псевдоним и запустить через псевдоним

`find_text_in_file.sh`

basic script commands:

`cat $2 | grep $1 | sort -h | head -$3 | cat -n -`

### Result:

```
192.168.1.99 - PuTTY
root@Smarthome:~/4.11_task# head -5 bmp180.h
#ifndef _BMP180_
#define _BMP180_

//i2c address
#define BMP180_Address 0x77
root@Smarthome:~/4.11_task# ./find_text_in_file.sh define bmp180.h 3 44
Usage: FT "text" File_name Num_of_string
root@Smarthome:~/4.11_task# ./find_text_in_file.sh define bmp180.h 4
1  #define _BMP180_
2  #define BMP180_Address 0x77
3  #define BMP180_CAL_AC1          0xAA  //Calibration data (16 bits)
4  #define BMP180_CAL_AC2          0xAC  //Calibration data (16 bits)
root@Smarthome:~/4.11_task#
```

Create alias:

<https://opensource.com/article/19/7/bash-aliases>

command for temporary alias:

alias FT='/root/4.11\_task/find\_text\_in\_file.sh'

result:

```
192.168.1.99 - PuTTY
root@Smarthome:~/4.11_task# alias FT='/root/4.11_task/find_text_in_file.sh'
root@Smarthome:~/4.11_task# FT 1 2 3 4
Usage: FT "text" File_name Num_of_string
root@Smarthome:~/4.11_task# FT define bmp180.h 2
1  #define _BMP180_
2  #define BMP180_Address 0x77
root@Smarthome:~/4.11_task#
```

4.11.7 Запустить 4.11.6 в ограниченном режиме для каталога /home/user

Script: find\_text\_in\_file\_home\_restr.sh

Added #!/bin/bash -r

To enable restricted mode

and change main script to avoid to search not in dir /home/\$USER:

cat /home/\$USER/\$2 | grep \$1 | sort -h | head -\$3 | cat -n -

Result:

```
pi@SmartHome: ~  
pi@SmartHome:~ $ head -3 /usr/b  
bin/      bmp180.h  
pi@SmartHome:~ $ head -3 /usr/bmp180.h  
#ifndef _BMP180_  
#define _BMP180_  
  
pi@SmartHome:~ $ head -3 ~/bmp180.h  
#ifndef _BMP180_  
#define _BMP180_  
  
pi@SmartHome:~ $ ./find_text_in_file.sh AC /usr/bmp180.h 3  
cat: /home/pi//usr/bmp180.h: No such file or directory  
pi@SmartHome:~ $ ./find_text_in_file.sh AC bmp180.h 3  
1  #define BMP180_CAL_AC1      0xAA  //Calibration data (16 bits)  
2  #define BMP180_CAL_AC2      0xAC  //Calibration data (16 bits)  
3  #define BMP180_CAL_AC3      0xAE  //Calibration data (16 bits)  
pi@SmartHome:~ $
```