

EPAM University Programs  
DevOps external course  
Module 4 Linux Essentials with Bash  
TASK 4.11

Danylenko Homework

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

```
danylenko@VM2:/tmp/scripts$ cat 4111.sh > 4111.log
danylenko@VM2:/tmp/scripts$ ./4111.sh >> 4111.log
danylenko@VM2:/tmp/scripts$ cat 4111.log
#!/bin/bash
##Task 4.11.1
echo
echo =====
echo "Task 4.11.1. Files bigger than 20Mb in /usr:"
find /usr -type f -size +20M

=====
Task 4.11.1. Files bigger than 20Mb in /usr:
/usr/lib/gcc/x86_64-linux-gnu/7/cc1
/usr/lib/firefox/libxul.so
/usr/lib/firefox/browser/omni.ja
/usr/lib/firefox/omni.ja
/usr/lib/thunderbird/libxul.so
/usr/lib/thunderbird/omni.ja
/usr/lib/libreoffice/program/libmergedlo.so
/usr/lib/x86_64-linux-gnu/libcudata.so.60.2
/usr/lib/x86_64-linux-gnu/libLLVM-9.so.1
/usr/lib/x86_64-linux-gnu/libLLVM-6.0.so.1
/usr/lib/x86_64-linux-gnu/libwebkit2gtk-4.0.so.37.39.5
/usr/lib/x86_64-linux-gnu/libmozjs-52.so.0.0.0
/usr/share/fonts/opentype/noto/NotoSerifCJK-Regular.ttc
/usr/share/fonts/opentype/noto/NotoSerifCJK-Bold.ttc
danylenko@VM2:/tmp/scripts$
```

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
```

```

danylenko@VM2:/tmp/scripts$ cat ./4112.sh > 4112.log
danylenko@VM2:/tmp/scripts$ ./4112.sh >> 4112.log
danylenko@VM2:/tmp/scripts$ cat 4112.log
#!/bin/bash
##Task 4.11.2
echo
echo =====
echo "Task 4.11.2."
echo "User's Home directory:"
echo $HOME
echo "Contains usual files:"
find $HOME -type f | rev | cut -d/ -f1 | rev | grep -ve "^\" | wc -l
echo "Hidden files:"
find $HOME -type f | rev | cut -d/ -f1 | rev | grep -e "^\" | wc -l

=====
Task 4.11.2.
User's Home directory:
/home/danylenko
Contains usual files:
434
Hidden files:
21

```

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

```

danylenko@VM2:~/tmp/scripts$ cat ./4113.sh > 4113.log
danylenko@VM2:~/tmp/scripts$ ./4113.sh >> 4113.log
danylenko@VM2:~/tmp/scripts$ cat 4113.log
#!/bin/bash
##Task 4.11.2
echo
echo =====
echo "Task 4.11.3"
echo "Date/Time: "$(date '+%D %T')
echo
echo "logon users:"
who
echo
echo "System uptime"
uptime

=====
Task 4.11.3
Date/Time: 05/10/20 16:05:25

logon users:
danylenko :0          2020-05-10 15:51 (:0)
utest     tty2        2020-05-10 16:02

System uptime
16:05:25 up 14 min,  2 users,  load average: 0,04, 0,12, 0,15
danylenko@VM2:~/tmp/scripts$

```

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

```
danylenko@VM2:~/tmp/scripts$ cat ./4114.sh > 4114.log
danylenko@VM2:~/tmp/scripts$ ./4114.sh >> 4114.log
danylenko@VM2:~/tmp/scripts$ cat 4114.log
#!/bin/bash
##Task 4.11.4
echo
echo =====
echo "Task 4.11.4"
echo "Processes of user:"
echo $USER
top -bn 1 -u $USER | tail -n +8 | wc -l
echo "Processes of user: root"
top -bn 1 -u root | tail -n +8 | wc -l

=====
Task 4.11.4
Processes of user:
danylenko
74
Processes of user: root
99
```

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

```
danylenko@VM2:~/tmp/scripts$ cat 4115.sh >> 4115.log
danylenko@VM2:~/tmp/scripts$ ./4115.sh >> 4115.log
danylenko@VM2:~/tmp/scripts$ cat 4115.log
#!/bin/bash
##Task 4.11.5
echo
echo =====
echo "Task 4.11.5"
echo "5 Processes with biggest memory use:"
top -bn 1 -o %MEM | tail -n +7 | head -n 6

=====
Task 4.11.5
5 Processes with biggest memory use:
  PID USER      PR  NI   VIRT   RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 1173 danylen+  20   0 2984952 201616 63372 S   7,7  20,1   1:01.72 gnome-shell
 1589 danylen+  20   0 1055000 134452 14524 S   0,0  13,4   0:05.06 gnome-softw+
 1002 danylen+  20   0 405044  59820 27368 S   0,0   6,0   0:22.82 Xorg
 1479 danylen+  20   0 893664  50280 10360 S   0,0   5,0   0:00.54 evolution-c+
 1492 danylen+  20   0 932512  47464  8420 S   0,0   4,7   0:00.48 evolution-c+
danylenko@VM2:~/tmp/scripts$
```



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

```
danylenko@VM2:~/tmp/scripts$ cat ./4116.sh > 4116.log
danylenko@VM2:~/tmp/scripts$ alias gpnum=~/.tmp/scripts/4116.sh
danylenko@VM2:~/tmp/scripts$ gpnum test >> 4116.log
danylenko@VM2:~/tmp/scripts$ gpnum echo 4116.sh >> 4116.log
danylenko@VM2:~/tmp/scripts$ gpnum echo 4116.sh 3 >> 4116.log
danylenko@VM2:~/tmp/scripts$ cat 4116.log
#!/bin/bash
##Task 4.11.6
echo =====
if [ $# -lt 2 ];then
    echo "Input at least 2 arguments:"
    echo "$0 text file [#rows]"
    exit 1
fi
text=$1;file=$2;n=1
if [ -z $3 ]; then maxrows=0; else maxrows=$3; fi
echo "4.11.6 output:"
grep "$text" $file | sort -r | while read -r row
do
    echo "$n: $row"
    if [ $n -eq $maxrows ]; then break; fi
    (( n++ ))
done; exit 0
=====
Input at least 2 arguments:
/home/danylenko/tmp/scripts/4116.sh text file [#rows]
=====
4.11.6 output:
1: echo "$n: $row"
2: echo "Input at least 2 arguments:"
3: echo "4.11.6 output:"
4: echo "$0 text file [#rows]"
5: echo =====
=====
4.11.6 output:
1: echo "$n: $row"
2: echo "Input at least 2 arguments:"
3: echo "4.11.6 output:"
danylenko@VM2:~/tmp/scripts$
```

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

```
danylenko@VM2:~/tmp/scripts$ cat 4117.sh > 4117.log
danylenko@VM2:~/tmp/scripts$ cd ~
danylenko@VM2:~$ alias gpR=~/.tmp/scripts/4117.sh
danylenko@VM2:~$ gpR echo ~/.tmp/scripts/4117.sh >> ~/.tmp/scripts/4117.log
/home/danylenko/tmp/scripts/4117.sh: line 4: cd: restricted
danylenko@VM2:~$ cat ~/.tmp/scripts/4117.log
#!/bin/bash -r
##Task 4.11.7
echo =====
cd .. || echo "script works in restricted mode in directory: $(pwd)"

if [ $# -lt 2 ];then
    echo "Input at least 2 arguments:"
    echo "$0 text file [#rows]"
    exit 1
fi
text=$1;file=$2;n=1
if [ -z $3 ]; then maxrows=0; else maxrows=$3; fi
echo "4.11.7 output:"
grep "$text" $file | sort -r | while read -r row
do
    echo "$n: $row"
    if [ $n -eq $maxrows ]; then break; fi
    (( n++ ))
done; exit 0
=====
script works in restricted mode in directory: /home/danylenko
4.11.7 output:
1: echo "$n: $row"
2: echo "Input at least 2 arguments:"
3: echo "4.11.7 output:"
4: echo "$0 text file [#rows]"
5: echo =====
6: cd .. || echo "script works in restricted mode in directory: $(pwd)"
danylenko@VM2:~$
```