

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

Danylenko Homework

4.12.1 Создать автоматический генератор паролей пользователей. На вход скрипта подать файл users.txt в котором содержится список пользователей:

1. user1

2. user2

...

20. user20

Автоматически сгенерировать пароли для всех пользователей, создать в скрипте пользователей linux-системы со сгенерированными паролями, так чтобы вы могли войти под именем каждого из пользователей с созданным для него паролем (тут предполагается работа с openssl), а также создать для каждого пользователя файл user*-login-password.txt, в который поместить имя пользователя и сгенерированный пароль. Например: user1 – uR44y6!#

```
#!/bin/bash
##Task 4.12.1
if [ $# -ne 1 ]
then
    echo "need file as argument"
    echo "line format:"
    echo "Num. Username"
    exit 1; fi
if [[ "$(id -u $(whoami))" != 0 ]]; then echo "Run script under root\sudo"; exit 1; fi
##variables
subdir=4121_danylenko
grp=utest
##creating subdir for pswd files
[ ! -d ${subdir} ] && mkdir ${subdir} &&\
chmod o-rwx ${subdir} &&\
echo "${subdir}/ created for pswd files, owner $(whoami), rights: o=-rwx"
##create-check utest group for all users
groupadd -f ${grp}
echo "users will be added to group ${grp}"
echo =====
for user in $(cut -d'.' -f2 ${1})
do
    pswd=$(openssl rand -base64 9)
    hash=$(openssl passwd -6 ${pswd})
    if sudo useradd -G ${grp} -s /bin/bash -d /home/${user} -m -p ${hash} ${user}
    then
        echo "$user - $pswd" > ${subdir}/${user}-login-password.txt
        echo "${subdir}/${user}-login-password.txt generated"
    else
        echo "Skiping... or do something else"
    fi
done
```

```

danylenko@VM2:~/tmp/scripts$ cat users.txt
1. User1
2. User2
3. User3
4. User4
5. User5
danylenko@VM2:~/tmp/scripts$ cat /etc/passwd | tail -n 3
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
utest:x:1001:1001::,/home/utest:/bin/bash
User1:x:1002:1002::/home/User1:/bin/bash
danylenko@VM2:~/tmp/scripts$ ./4121.sh users.txt
Run script under root\sudo
danylenko@VM2:~/tmp/scripts$ sudo ./4121.sh users.txt
4121_danylenko/ is created for pswd files, owner root, rights: o=-rwx
users will be added to group utest
=====
useradd: user 'User1' already exists
Skipping... or do something else
4121_danylenko/User2-login-password.txt generated
4121_danylenko/User3-login-password.txt generated
4121_danylenko/User4-login-password.txt generated
4121_danylenko/User5-login-password.txt generated
danylenko@VM2:~/tmp/scripts$ cat 4121_danylenko/User4-login-password.txt
cat: 4121_danylenko/User4-login-password.txt: Permission denied
danylenko@VM2:~/tmp/scripts$ sudo cat 4121_danylenko/User4-login-password.txt
User4 - +hqcNql7AzIv
danylenko@VM2:~/tmp/scripts$ su User4^C
danylenko@VM2:~/tmp/scripts$ su User4
Password:
User4@VM2:/home/danylenko/tmp/scripts$ id
uid=1005(User4) gid=1005(User4) groups=1005(User4),1001(utest)
User4@VM2:/home/danylenko/tmp/scripts$ █

```

4.12.2 Взять за основу проект <https://habr.com/ru/post/155201/> . Написать скрипт выполняющий следующее:

1. При первоначальном запуске – вычисление контрольных сумм и архивация проекта.
2. периодическая проверка проекта на предмет изменений.
3. Если проект изменился, то записать новую версию в новый архив и запустить перекомпиляцию проекта.


```
#!/bin/bash
#Task 4.12.2 MD5 SUMMS
arch_dir=~/.tmp/scripts/4122Project
proj_dir=~/.tmp/scripts/4122Project/mf
proj_name=mf
md5file=${arch_dir}/${proj_name}.md5
compiler=g++
prog=make
##### check programs asking to install #####
for pr in ${compiler} ${prog}
do
    if ! command -v ${pr} >/dev/null 2>&1
    then
        echo "##### need ${pr} to work"
        echo "##### to install with sudo (y\n)? : "
        read
        if [[ "${REPLY}" == "y" ]]
        then
            sudo apt-get install ${pr}
        fi;fi;done

### first launch if there is no md5 file #####
if [ ! -f ${md5file} ]
then
    echo "##### gener. md5 file... : ${md5file}"
    find ${proj_dir} -type f -exec md5sum {} \; > ${md5file}
    arch_file=${arch_dir}/${date '+%d%m%Y_%H%M%S'}_${proj_name}.tar.gz
    echo "##### Creating archive... ${arch_file}"
    tar -czf ${arch_file} ${proj_dir}/* ${md5file}
fi
##### SLEEP CYCLE 20s #####
```

```
##### SLEEP CYCLE 20s #####
while : ; do
    sleep 20
    #checking md5 sums
    if ! md5sum -c --status ${md5file}
    then
        echo "##### Change!!!: generating new md5 in archive"
        find ${proj_dir} -type f -exec md5sum {} \; > ${md5file}
        arch_file=${arch_dir}/${date '+%d%m%Y_%H%M%S'}_${proj_name}.tar.gz
        echo "##### Creating archive ${arch_file}"
        tar -czf ${arch_file} ${proj_dir}/* ${md5file}
        echo "##### recompiling Project: ${proj_name}"
        cd ${proj_dir}
        make -f Makefile-3
        echo "##### new md5 after compilation ${md5file}"
        find ${proj_dir} -type f -exec md5sum {} \; > ${md5file}
    #remove results: make -f Makefile-3 clean
    else
        echo "##### $(date '+%D %T'): nothing changed"
    fi
done
```

OUTPUT:

```
danylenko@VM2:~/tmp/scripts$ ./4122.sh
##### gener. md5 file... : /home/danylenko/tmp/scripts/4122Project/mf.md5
##### Creating archive... /home/danylenko/tmp/scripts/4122Project/12052020_230812_mf.tar.gz
tar: Removing leading '/' from member names
##### 05/12/20 23:08:32: nothing changed
##### 05/12/20 23:08:52: nothing changed
##### Change!!!: generating new md5 in archive
##### Creating archive /home/danylenko/tmp/scripts/4122Project/12052020_230912_mf.tar.gz
tar: Removing leading '/' from member names
##### recompiling Project: mf
g++ -c -Wall main.cpp
g++ -c -Wall factorial.cpp
g++ -c -Wall hello.cpp
g++ main.o factorial.o hello.o -o hello
##### new md5 after compilation /home/danylenko/tmp/scripts/4122Project/mf.md5
##### 05/12/20 23:09:33: nothing changed
##### Change!!!: generating new md5 in archive
##### Creating archive /home/danylenko/tmp/scripts/4122Project/12052020_230953_mf.tar.gz
tar: Removing leading '/' from member names
##### recompiling Project: mf
g++ -c -Wall main.cpp
g++ main.o factorial.o hello.o -o hello
##### new md5 after compilation /home/danylenko/tmp/scripts/4122Project/mf.md5
##### 05/12/20 23:10:14: nothing changed
##### 05/12/20 23:10:34: nothing changed
^C
danylenko@VM2:~/tmp/scripts$
```

```
danylenko@VM2:~/tmp/scripts/4122Project$ ls -lR
.:
total 24
-rw-r--r-- 1 danylenko danylenko 1310 tpa 12 23:08 12052020_230812_mf.tar.gz
-rw-r--r-- 1 danylenko danylenko 1304 tpa 12 23:09 12052020_230912_mf.tar.gz
-rw-r--r-- 1 danylenko danylenko 5966 tpa 12 23:09 12052020_230953_mf.tar.gz
drwxr-xr-x 2 danylenko danylenko 4096 tpa 12 23:09 mf
-rw-r--r-- 1 danylenko danylenko 1046 tpa 12 23:09 mf.md5

./mf:
total 60
-rw-r--r-- 1 danylenko danylenko 116 жов 18 2012 factorial.cpp
-rw-r--r-- 1 danylenko danylenko 1368 tpa 12 23:09 factorial.o
-rw-r--r-- 1 danylenko danylenko 42 жов 18 2012 functions.h
-rwxr-xr-x 1 danylenko danylenko 13328 tpa 12 23:09 hello
-rw-r--r-- 1 danylenko danylenko 115 жов 18 2012 hello.cpp
-rw-r--r-- 1 danylenko danylenko 2592 tpa 12 23:09 hello.o
-rw-r--r-- 1 danylenko danylenko 194 tpa 12 23:09 main.cpp
-rw-r--r-- 1 danylenko danylenko 3112 tpa 12 23:09 main.o
-rw-r--r-- 1 danylenko danylenko 52 жов 18 2012 Makefile-1
-rw-r--r-- 1 danylenko danylenko 236 жов 18 2012 Makefile-2
-rw-r--r-- 1 danylenko danylenko 661 жов 18 2012 Makefile-3
-rw-r--r-- 1 danylenko danylenko 250 жов 18 2012 Makefile-4
danylenko@VM2:~/tmp/scripts/4122Project$
```


4.12.3 Создать скрипт сбора статистики работы системы

На этапе инициализации:

```
danylenko@VM2:~/tmp/scripts/4123logs$ crontab -u danylenko -l | tail -n 5
*/2 * * * * ~/tmp/scripts/4123_mem_cron.sh >/dev/null 2>&1

*/5 * * * * ~/tmp/scripts/4123_logzip.sh >/dev/null 2>&1

*/2 * * * * ~/tmp/scripts/4123_main.sh >/dev/null 2>&1

danylenko@VM2:~/tmp/scripts/4123logs$ cat /var/log/syslog | grep CRON | tail -n 10
May 12 19:35:02 VM2 CRON[10742]: (danylenko) CMD (/tmp/scripts/4123_logzip.sh >/dev/null 2>&1)
May 12 19:36:01 VM2 CRON[10787]: (danylenko) CMD (/tmp/scripts/4123_mem_cron.sh >/dev/null 2>&1)
May 12 19:36:01 VM2 CRON[10788]: (danylenko) CMD (/tmp/scripts/4123_main.sh >/dev/null 2>&1)
May 12 19:38:01 VM2 CRON[10834]: (danylenko) CMD (/tmp/scripts/4123_main.sh >/dev/null 2>&1)
May 12 19:38:01 VM2 CRON[10835]: (danylenko) CMD (/tmp/scripts/4123_mem_cron.sh >/dev/null 2>&1)
May 12 19:40:01 VM2 CRON[10870]: (danylenko) CMD (/tmp/scripts/4123_mem_cron.sh >/dev/null 2>&1)
May 12 19:40:01 VM2 CRON[10871]: (danylenko) CMD (/tmp/scripts/4123_logzip.sh >/dev/null 2>&1)
May 12 19:40:01 VM2 CRON[10872]: (danylenko) CMD (/tmp/scripts/4123_main.sh >/dev/null 2>&1)
May 12 19:42:01 VM2 CRON[10959]: (danylenko) CMD (/tmp/scripts/4123_main.sh >/dev/null 2>&1)
May 12 19:42:01 VM2 CRON[10960]: (danylenko) CMD (/tmp/scripts/4123_mem_cron.sh >/dev/null 2>&1)
danylenko@VM2:~/tmp/scripts/4123logs$
```

Создайте задание для cron, согласно которому каждые 5 минут файл ~/memory/stat, а также логи, полученные основным скриптом, будут упаковываться в архив.

```
danylenko@VM2:~/tmp/scripts/4123logs$ cat ~/tmp/scripts/4123_logzip.sh
#!/bin/bash
#Cron task to archive set of files\directories
outfolder=~/tmp/scripts/4123Gzip
files=(\
~/memory/stat\
~/tmp/scripts/4123logs/vmstat\
~/tmp/scripts/4123logs/ld_avrg\
~/tmp/scripts/4123logs/mempages\
~/tmp/scripts/4123logs/hdd\
)
tar -czf $outfolder/${date '+%d%m%Y %H%M%S'} stats.tar.gz ${files[@]}
##removing log files each 5 mins in this homework to not leave long junk files
rm ${files[@]}
danylenko@VM2:~/tmp/scripts/4123logs$
```

```
danylenko@VM2:~/tmp/scripts/4123logs$ ls -l ../4123Gzip
total 20
-rw-rw-r-- 1 danylenko danylenko 1576 tpa 12 19:30 12052020_193001_stats.tar.gz
-rw-rw-r-- 1 danylenko danylenko 1531 tpa 12 19:35 12052020_193502_stats.tar.gz
-rw-rw-r-- 1 danylenko danylenko 1598 tpa 12 19:40 12052020_194001_stats.tar.gz
-rw-rw-r-- 1 danylenko danylenko 1741 tpa 12 19:45 12052020_194501_stats.tar.gz
-rw-rw-r-- 1 danylenko danylenko 1632 tpa 12 19:50 12052020_195001_stats.tar.gz
danylenko@VM2:~/tmp/scripts/4123logs$
```

Создайте задание для cron, согласно которому каждые 2 минуты в файл ~/memory/stat будет добавляться информация о текущем состоянии памяти, без учета размера подкачки и заголовка.

```
danylenko@VM2:~/tmp/scripts$ cat 4123_mem_cron.sh
#!/bin/bash
#cron task for memory stats
echo "$(date '+%D %T') $(free | grep Mem)" >> ~/memory/stat
danylenko@VM2:~/tmp/scripts$ cat ~/memory/stat
05/12/20 19:52:01 Mem:      1004824      634172      103500      31156      267152      198428
05/12/20 19:54:01 Mem:      1004824      635356      100988      31156      268480      197248
danylenko@VM2:~/tmp/scripts$
```

Основной этап выполнять каждые две минуты:

С помощью команды `vmstat`, в течении 30с с интервалом в 3с, собирайте статистику об использовании ресурсов системы. Посчитайте среднее количество переключений контекста ядра в секунду на заданном интервале времени. Информацию – в лог.

Получите информацию о средней загрузенности процессора в течении последних 15м. Информацию – в лог.

Опишите текущее состояние страниц памяти, доступных в вашей системе. Информацию – в лог.

Опишите текущее состояние разделов жестких дисков, доступных в вашей системе. Информацию – в лог.

```
GNU nano 2.9.3                                     4123_main.sh

#!/bin/bash
logdir=~/.tmp/scripts/4123logs

#vmstat stats, for 30s, skip 1st line with avr from uptime
sum=0
#sleep 1
for cs in $(vmstat -w 1 11 | tail -n 10 | awk '{print $12}')
do
    ((sum+=cs))
done;
echo "$(date '+%D %T') CS(-30/3s_avrg): $((sum/10))" >> $logdir/vmstat

#load_average - top
echo "$(date '+%D %T') load_average_15m: $(uptime | awk '{print $10}')" >> $logdir/ld_avrg

#memory state to log file
echo "===== " >> $logdir/mempages
echo "$(date '+%D %T') state of /proc/meminfo" >> $logdir/mempages
echo "===== " >> $logdir/mempages
cat /proc/meminfo >> $logdir/mempages

#disk state to log file
echo "===== " >> $logdir/hdd
echo "$(date '+%D %T') state of HDD volumes" >> $logdir/hdd
echo "===== " >> $logdir/hdd
df >> $logdir/hdd
```



```

danylenko@VM2:~/tmp/scripts/4123logs$ cat ld_avg
05/12/20 19:56:11 load_average_15m: 0,16
danylenko@VM2:~/tmp/scripts/4123logs$ cat vmstat
05/12/20 19:56:11 CS(-30/3s_avg): 706
danylenko@VM2:~/tmp/scripts/4123logs$ cat mempages
=====
05/12/20 19:56:11 state of /proc/meminfo
=====
MemTotal:      1004824 kB
MemFree:       102548 kB
MemAvailable:  199008 kB
Buffers:       41024 kB
Cached:        200636 kB
SwapCached:    10408 kB
Active:        403576 kB
Inactive:      360296 kB
Active(anon):  265260 kB
Inactive(anon): 288104 kB
Active(file):  138316 kB
Inactive(file): 72192 kB
Unevictable:   32 kB
Mlocked:       32 kB
SwapTotal:     483800 kB
SwapFree:      246488 kB
Dirty:         316 kB
Writeback:     0 kB
AnonPages:     519040 kB
Mapped:        137536 kB
Shmem:         31152 kB
KReclaimable:  27012 kB
Slab:          56992 kB
SReclaimable:  27012 kB
SUnreclaim:    29980 kB
KernelStack:   6784 kB
danylenko@VM2:~/tmp/scripts/4123logs$ cat hdd
=====
05/12/20 19:56:11 state of HDD volumes
=====

```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
udev	478072	0	478072	0%	/dev
tmpfs	100484	1372	99112	2%	/run
/dev/sda1	10253588	6336088	3376932	66%	/
tmpfs	502412	0	502412	0%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	502412	0	502412	0%	/sys/fs/cgroup
/dev/loop0	3840	3840	0	100%	/snap/gnome-system-monitor/127
/dev/loop1	4480	4480	0	100%	/snap/gnome-calculator/704
/dev/loop2	164096	164096	0	100%	/snap/gnome-3-28-1804/116
/dev/loop3	1024	1024	0	100%	/snap/gnome-logs/93
/dev/loop4	56064	56064	0	100%	/snap/core18/1668
/dev/loop5	46080	46080	0	100%	/snap/gtk-common-themes/1440
/dev/loop6	49536	49536	0	100%	/snap/gtk-common-themes/1474
/dev/loop7	3840	3840	0	100%	/snap/gnome-system-monitor/135
/dev/loop8	93568	93568	0	100%	/snap/core/8689
/dev/loop9	15104	15104	0	100%	/snap/gnome-characters/399
/dev/loop10	15104	15104	0	100%	/snap/gnome-characters/495
/dev/loop11	56320	56320	0	100%	/snap/core18/1705
/dev/loop12	91264	91264	0	100%	/snap/core/8268
/dev/loop13	4352	4352	0	100%	/snap/gnome-calculator/544
/dev/loop14	1024	1024	0	100%	/snap/gnome-logs/81
tmpfs	100480	52	100428	1%	/run/user/1000
tmpfs	100480	0	100480	0%	/run/user/1001
scripts	218669212	145333936	73335276	67%	/media/sf_scripts

```

danylenko@VM2:~/tmp/scripts/4123logs$

```