

EPAM University Programs  
DevOps external course  
Module 4 Linux Essentials with Bash  
TASK 4.12

**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
# Create file with users
if [ ! -f "users.txt" ]
then
    for (( i=1; i<21; i++ ))
    {
        echo "$i. user$i" >> users.txt
    }
else echo
fi
# Generate passwords and create a file with passwords
if [ ! -f "passwords.txt" ]
then
    for (( i=1; i<21; i++ ))
    {
        openssl rand -base64 8 >> passwords.txt
    }
else echo
fi
# Merge two files into one file "userspass.txt"
paste users.txt passwords.txt > userspass.txt
# Delete the first column of a file userspass.txt
# and create temporary file
awk '{print $2,$3}' userspass.txt > temp
# Create files for users with pass and login
for (( i=1; i<21; i++ ))
{
    sed $i!d temp > user$i-login-password.txt
}
# Delete temporary files
rm temp passwords.txt
```

*This script creates a file with users names, if it needs, then generates passwords for each user and saves this information to the file userspass.txt, then creates files with login and password for each user.*

```
#!/bin/bash
# Create users with information from file userspass.txt
for (( i=1; i<21; i++ )) {
    sudo useradd -p $(openssl passwd -6 $(sed $i!d userspass.txt | awk
'{print $3}'))\
    -s /bin/bash $(sed $i!d userspass.txt | awk '{print $2}')
}
```

*This script creates users to log in use information from file userspass.txt with login and password for each user.*

*Information from /etc/passwd*

```
bobrov:x:1000:1000:bobrov,,,:/home/bobrov:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
sshd:x:122:65534::/run/sshd:/usr/sbin/nologin
utest:x:1001:1001::/home/utest:/bin/bash
guest:x:1002:1002::/home/guest:/bin/bash
user1:x:1003:1003::/home/user1:/bin/bash
user2:x:1004:1004::/home/user2:/bin/bash
user3:x:1005:1005::/home/user3:/bin/bash
user4:x:1006:1006::/home/user4:/bin/bash
user5:x:1007:1007::/home/user5:/bin/bash
user6:x:1008:1008::/home/user6:/bin/bash
user7:x:1009:1009::/home/user7:/bin/bash
user8:x:1010:1010::/home/user8:/bin/bash
user9:x:1011:1011::/home/user9:/bin/bash
user10:x:1012:1012::/home/user10:/bin/bash
user11:x:1013:1013::/home/user11:/bin/bash
user12:x:1014:1014::/home/user12:/bin/bash
user13:x:1015:1015::/home/user13:/bin/bash
user14:x:1016:1016::/home/user14:/bin/bash
user15:x:1017:1017::/home/user15:/bin/bash
user16:x:1018:1018::/home/user16:/bin/bash
user17:x:1019:1019::/home/user17:/bin/bash
user18:x:1020:1020::/home/user18:/bin/bash
user19:x:1021:1021::/home/user19:/bin/bash
user20:x:1022:1022::/home/user20:/bin/bash
bobrov@bobrov-VirtualBox:~/Documents/homework412$
```

*Test change  
user to the  
newly created*

```
bobrov@bobrov-VirtualBox:~/Documents/homework412$ su user1
Password:
user1@bobrov-VirtualBox:/home/bobrov/Documents/homework412$ su user12
Password:
user12@bobrov-VirtualBox:/home/bobrov/Documents/homework412$ su bobrov
Password:
bobrov@bobrov-VirtualBox:~/Documents/homework412$
```

## Information from /etc/shadow

```
bobrov:$6$VMVnWkC1$G0vNpX8virziUqyj0RDaUGTPIYJsyZB8D8aCq.6YpIyycC.tapkdYbwqUB9IxfZgwSwaD.KvBiFXMevXp3s/:18356:0:99999:7:::
vboxadd:!:18356:~::~:
sshd:!:18363:0:99999:7:::
utest:$6$wybFqJrZ$qiG0coPtVig5cB1N.lnx.kIwYvG0gKiu.pZhAwkrtZpe7B9j1thdPtpJ4WnIFax6IxzrNzYxXt83q6ZQLQ.z01:18376:0:99999:7:::
guest:$6$nJf09egC$Tz0Rm2b9MCCUnEp0mJX1.f31av99813pEFNnS1AxCf0v4a5ZOG1pUKVNmftjRegU8PtKBE55fIdcL3TvtBY1:18377:0:99999:7:::
user1:$6$wS4neXZRFGD0RqzJ$RvoQ33LVkDbRIHmTXKjGWEKUkSxV1JGOMVgIFbH0ZpbsjM92b2A1sD.KLodZZbxEGy8ZDqiZLA1LG3bI7aoL.:18393:0:99999:7:::
user2:$6$jbIFMEWGb2Y1P1w6$5x6J/SyLVqJkGfK8jIUn9qV2k6XX9EV7YqkJWtASQFV7UQNfJJ1iWggoudCGsHRe706JFSasnqqxmHwx1v5Fp1:18393:0:99999:7:::
user3:$6$0jFzPe.y9e7DL5f$I5qh3PXfZwQZJX.obQBFXGPFuLF0ucEqN8Xo.XWaj6AU/WpSxwf6fdXLv50R2uGQfKD6uysgf7q5gy.WoUBKy/:18393:0:99999:7:::
user4:$6$GrUwGchtL4ZyCOY.$8byR1jCAIyZDVocfNGtkz32fjAGupdhKAyCyYVE331h5seCaZDqIXXGNmSFkR0Dqk1kBlZ12ydiQIV4hmhe2.:18393:0:99999:7:::
user5:$6$PvewN1qwIX1SJak$h6uoRV4/pzB0k0Uxi6xZVuhd9XydE0FwsnRpqVE4mIxmPgKfLNPDFZFdbCRzzmFKCAvkiIyJZHQlCBib0aRL0:18393:0:99999:7:::
user6:$6$GW.op58aP0cEmc02$/Yn1NyTBHMHhNat0L9nk1f90psZPXLr9/vCWKXoFyfpbd1xgHn2DRBKs0ZUabhKJpXDqdZPGdwrY2nWzsFPW0:18393:0:99999:7:::
user7:$6$PMo2wS1VfuLRWBHM$K7I1976PGItfSGGcf8ySwZ4HDV4u/wTOC3rIpG5Tr8w6.cNwER72oNsGJ0G0EonvxaH0rCYfj7RFNj73/2qsM1:18393:0:99999:7:::
user8:$6$ea7vFltg47b81ys4$S.e0hAaTXGdnOgWb0V0QCpaqhgicEiyzMgKHyoY0xvtaAomR72kp.Mj90bsVjIMxCwApQVhXhXtE5TTIQmFSN1:18393:0:99999:7:::
user9:$6$RU6hh70swNHkBbt$PwdrnVzhh153ZrgwT.xcbALCSt.mOXVa0HxcApmARBqgxvYQ178ZfASX/gQNPkKzu9A07DI11whuo1H1sbqVf1:18393:0:99999:7:::
user10:$6$0h5ivsxi05PIy89p$K0RrkicWEMeg5a4sr3FmZUuM/1/rDwCKyqQNdMqGx1NC8tMJjwErThnZ/44wegkNBubgw8E8eqLlt0iFkeD0:18393:0:99999:7:::
user11:$6$AnAz2okrb1pai44t$WxTVJR/xf0LgvwhP528mFUG6zRy1YKngfGX6Vi4d0/Tj5tgdXS/fuc9APinjk2V.ABFc9xjg4jW7HyC7hAjoJ.:18393:0:99999:7:::
user12:$6$HLJpOhHS.JYY7wD.$4SjymM8Ezss0DCGLW/no4PJPG/eHcChAhIAK1GqG3nIXfLgKpFTNdDvWiy8384bc15nraRxpntx3QIGxsGwq.:18393:0:99999:7:::
user13:$6$XDpQ2cewCr6Los0F$0bSdGee8FHkQQuiHuoQzttvHksbPGWlyV0glTjVgXT0d7XEJznjVUHKmyHDdwwNRdJwzCG.vT3suF2IUHKMk.:18393:0:99999:7:::
user14:$6$WWLxlgKwaLQ718HC$36DI7qLWb6gTj8k0AK.f/5JRR8Pqh5yxmYOU2iIoXJk9p8.Vv1LiuRM/g7I72a88KqVv92TDMvnEkB.eWtpFM.:18393:0:99999:7:::
user15:$6$5oGuNEUJPCJwhqLz$ru1A40cv48R40roIGAxz.LxuFx10fGb1iD.ek7GbbpbF8rnA0q9TeP5fy6xswZN4yoV77LkYvU7NmPdI0ngl0:18393:0:99999:7:::
user16:$6$3/nYddACitUrw/t.$HpGdvqYqt64d90EQHw3T0BGuD22Pro4P9pWtF1GWNZAuF3PmwEdToKy3NdCT7aQSurqSPa3hVtWQBdmpqor.:18393:0:99999:7:::
user17:$6$Cki6Q0qn3siRGIgo$rhEHikmgQDf6TVQb.Eit3vRqzfy1EHSrcg6nTkwxE8TLlk1XhJgRcbw5tCRmpcZsBtpKekp3ywbQ3PfHqGmzB/:18393:0:99999:7:::
user18:$6$fWrWQEEeWB/T8cXL$1NiL0v5q1JWPhTThmMZRv0WnbVNZUD1cQ1mcBv9DRyDFiyjLiheeSVKfQqW/JW..1w9oxw0wG9THUco15gkie1:18393:0:99999:7:::
user19:$6$Q.55KTQT7kT7oUE8$dHtAlUMj5XxLHY12K3is51zRWHAN64LQQT0N.RaNM.Dex5EUW0Jgy54ItIKGduzLYP0szAePbYKrc2wHv5cn1:18393:0:99999:7:::
user20:$6$83zkFE6nRV.NT/P/$hC8cWIAu6heXfAOZoo5UZASFA$NnfCg2JJy4Qg1zokX105s4KCtcgi0bJ9b0fSqV8nKNdjcm/Iw/70yD2cLzn/:18393:0:99999:7:::
bobrov@bobrov-VirtualBox:~/Documents/homework412$
```

**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.

### Script file - scr4122

```
#!/bin/bash
#Calculate checksums of project files. The command is executed from a
file Makefile.
make md5f
#Perform project archiving. The command is executed from a file Makefile.
make tarf
#Define variables for readability.
name1=main.cpp
name2=hello.cpp
name3=factorial.cpp
name4=functions.h
name5=project
#Run an endless loop in which the project files are checked,
#if they were changed, then archived into a new archive and compiling
files.
#Archiving and compilation in a loop are performed from a file Makefile.
#The cycle repeats every minute.
```

```
while :
do
    datef=$(date +%m%d%H%M)
    if [[ -N $name1 || -N $name2 || -N $name3 || -N $name4 ]] ; then
        make tarf
        cp $name5.tar.gz $name5$datef.tar.gz
        make
    fi
    sleep 60
done
```

## Makefile content

```
CC=g++

all: hello tarf md5f

hello:    main.o factorial.o hello.o
    $(CC) main.o factorial.o hello.o -o hello

main.o:   main.cpp
    $(CC) -c main.cpp

factorial.o: factorial.cpp
    $(CC) -c factorial.cpp

hello.o:  hello.cpp
    $(CC) -c hello.cpp

md5f: factorial.cpp.md5 functions.h.md5 hello.cpp.md5 main.cpp.md5

factorial.cpp.md5: factorial.cpp
    md5sum factorial.cpp > factorial.cpp.md5

functions.h.md5: functions.h
    md5sum functions.h > functions.h.md5

hello.cpp.md5: hello.cpp
    md5sum hello.cpp > hello.cpp.md5

main.cpp.md5: main.cpp
    md5sum main.cpp > main.cpp.md5

tarf: factorial.cpp functions.h hello.cpp main.cpp
    tar -zvcf project.tar.gz factorial.cpp functions.h hello.cpp
main.cpp

clean:
    rm hello main.o factorial.o hello.o main.cpp.md5\
    hello.cpp.md5 functions.h.md5 factorial.cpp.md5
```

```
bobrov@bobrov-VirtualBox:~/Documents/homework412/4122$ ./scr4122
md5sum factorial.cpp > factorial.cpp.md5
md5sum functions.h > functions.h.md5
md5sum hello.cpp > hello.cpp.md5
md5sum main.cpp > main.cpp.md5
tar -zvcf project.tar.gz factorial.cpp functions.h hello.cpp main.cpp
factorial.cpp
functions.h
hello.cpp
main.cpp
tar -zvcf project.tar.gz factorial.cpp functions.h hello.cpp main.cpp
factorial.cpp
functions.h
hello.cpp
main.cpp
g++ -c main.cpp
g++ -c factorial.cpp
g++ -c hello.cpp
g++ main.o factorial.o hello.o -o hello
tar -zvcf project.tar.gz factorial.cpp functions.h hello.cpp main.cpp
factorial.cpp
functions.h
hello.cpp
main.cpp
md5sum hello.cpp > hello.cpp.md5
tar -zvcf project.tar.gz factorial.cpp functions.h hello.cpp main.cpp
factorial.cpp
functions.h
hello.cpp
main.cpp
g++ -c hello.cpp
g++ main.o factorial.o hello.o -o hello
tar -zvcf project.tar.gz factorial.cpp functions.h hello.cpp main.cpp
factorial.cpp
functions.h
functions.h
hello.cpp
main.cpp
md5sum hello.cpp > hello.cpp.md5
^C
bobrov@bobrov-VirtualBox:~/Documents/homework412/4122$
```

#### 4.12.3 Создать скрипт сбора статистики работы системы

На этапе инициализации:

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

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.

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 загрузке процессора в течении последних 15с. 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.