

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

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
cat users.txt | while read LINE
do
    GEN=$(openssl rand -base64 8)
    PASS=$(mkpasswd -m md5 $GEN)
    echo "$LINE $GEN" > $LINE"-login-password".txt
    useradd -g user -s /bin/bash -d /home/$LINE -m $LINE -p $PASS
done
```

```
root@aku-ПК:/home/Epam/4.12# ./4.12.1.sh
root@aku-ПК:/home/Epam/4.12# su user12
user12@aku-ПК:/home/Epam/4.12$
```

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

while :
do
if [ "$var1" == "0" ]
then
sleep 5

md5sum -c /home/Epam/4.12/4.12.2/checksum.md5

var1="$?"

md5sum * */* 2>/dev/null > /home/Epam/4.12/4.12.2/checksum.md5

echo "var1= "$var1

else

#arch

tar -zcf /home/Epam/4.12/4.12.2/arc_$(date +%d%m%y,%T).tgz
/home/Epam/4.12/4.12.2

#recompile

make -f Makefile-2 clean

#check md5

md5sum * */* 2>/dev/null > /home/Epam/4.12/4.12.2/checksum.md5
```

sleep 5

#check md5

md5sum -c /home/Epam/4.12/4.12.2/checksum.md5

#0-zero, 1-new var

var1="\$?"

echo "var1= "\$var1

fi

done

```
root@aku-ПК:/home/Epam/4.12/4.12.2# ./4.12.2.sh
tar: Removing leading `/' from member names
rm -rf *.o hello
4.12.2.sh: OK
Makefile-1: OK
Makefile-2: OK
Makefile-3: OK
Makefile-4: OK
arc_140520,01:29:35.tgz: OK
arc_140520,01:29:40.tgz: OK
arc_140520,01:29:45.tgz: OK
arc_140520,01:29:51.tgz: OK
arc_140520,01:30:26.tgz: OK
checksum.md5: FAILED
factorial.cpp: OK
functions.h: OK
hello.cpp: OK
main.cpp: OK
md5sum: WARNING: 1 computed checksum did NOT match
var1= 1
tar: Removing leading `/' from member names
rm -rf *.o hello
```

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.

```
* /5 * * * * tar -zcf
```

```
/home/Epam/4.12/4.12.3/arc.tgz  
/home/Epam/4.12/4.12.3/memory_stat.txt
```

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.

```
* /2 * * * * free | grep Mem: >> /home/Epam/4.12/4.12.3/memory_stat.txt
```

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 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.

```
#!/bin/bash
```

```
vmstat 3 10
```

```
uptime | awk '{ print "Load average: "$9 }'
```

```
free -h
```

```
df -h
```

```
*/2 * * * * /home /Epam/4.12/4.12.3/4.12.3.sh >> /home/Epam/4.12/4.12.3/memory_stat.txt
```

```
root@aku-ПК:/home/Epam/4.12/4.12.3# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
*/5 * * * * tar -zcf /home /Epam/4.12/4.12.3/arch.tgz /home/Epam/4.12/4.12.3/memory_stat.txt
*/2 * * * * free | grep Mem: >> /home/Epam/4.12/4.12.3/memory_stat.txt
*/2 * * * * /home//Epam/4.12/4.12.3/4.12.3.sh >> /home/Epam/4.12/4.12.3/memory_stat.txt
root@aku-ПК:/home/Epam/4.12/4.12.3#
```