EPAM University Programs

DevOps external course

Module 4 Linux Essentials with Bash

TASK 4.11

**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 созданнным 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

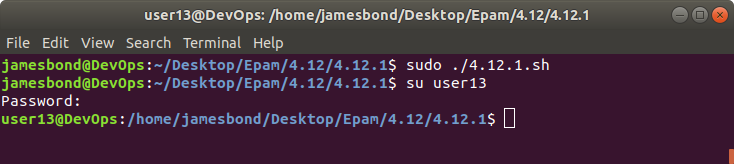
GEN=$(openssl rand -base64 8)

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



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

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

md5sum -c checksum.md5

var1="$?"

echo "var1= "$var1

sleep 5

else

#архивация проекта

tar -zcf /home/jamesbond/Desktop/Epam/\_\_4.12/4.12.2/arc\_$(date +%d%m%y,%T).tgz /home/jamesbond/Desktop/Epam/4.12/4.12.2

#запустить перекомпиляцию проекта

make -f Makefile-2 clean

#запись контрольной суммы в файл

md5sum factorial.cpp functions.h hello.cpp main.cpp > checksum.md5

sleep 5

#проверить не изменилась ли контрольная сумма

md5sum -c checksum.md5

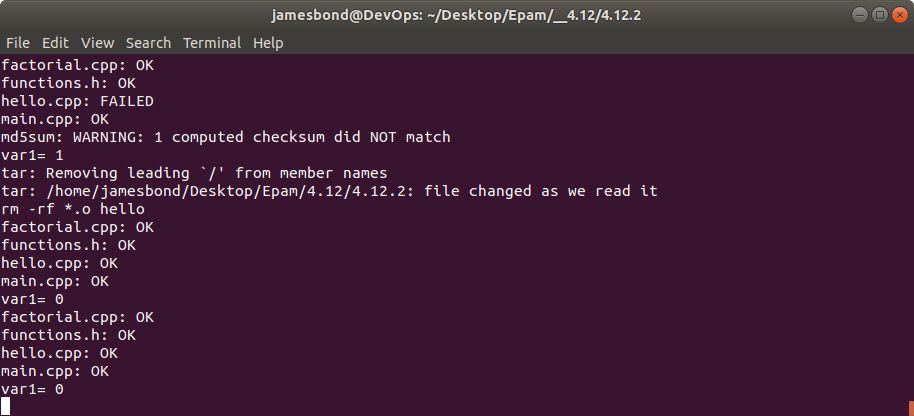
#0-если не изменилась, 1-если иеменилась, записали значение в var1

var1="$?"

echo "var1= "$var1

fi

done



**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/jamesbond/Desktop/Epam/\_4.12/4.12.3/arc.tgz /home/jamesbond/Desktop/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/jamesbond/Desktop/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(cs) ядра в секунду на заданном интервале времени. Информацию – в лог.

Получите информацию о средней загруженности 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.

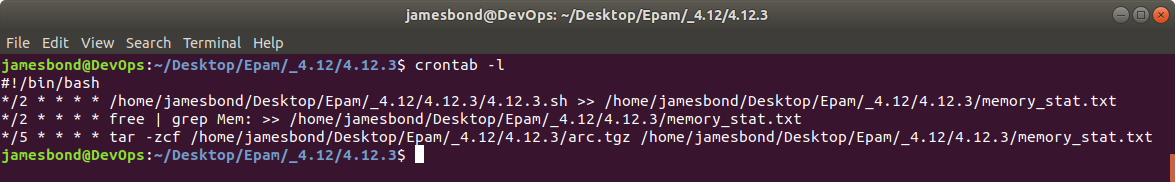
#!/bin/bash

vmstat 3 10

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

free -h

df -h

\*/2 \* \* \* \* /home/jamesbond/Desktop/Epam/\_4.12/4.12.3/4.12.3.sh >> /home/jamesbond/Desktop/Epam/\_4.12/4.12.3/memory\_stat.txt