

تقرير مشروع النظم

جودي حسان طالب

يوسف بسام العيسى

زين ماهر محيسن

السؤال الأول :

1. إضافة حقل جديد في الداتابيز

```
1 AddRecord(){
2 read -r -p "please enter the key: " key
3 var=`grep ^$key os1db`
4 if [ -z "$var" ];
5 then
6 read -r -p "please enter the value: " value1
7 value=`echo -n $value1 | base64 `
8 record="$key : $value"
9
10 #TODO
11 echo $record >>os1db
12 echo "Adding new key to the database (first.os1db)..."
13 echo "Done"
14 else
15 echo "the key already exists please Try again";
16 AddRecord
17 fi
18 }
```

بداية نقوم بإدخال ال key من خلال تعليمة ال

Read -r -p

وهي تعليمة ل قراءة الادخال من المستخدم ال r- هي لعدم التعامل مع ال / على انها تعليمة خاصة واعتبرها سطر ادخال.
-p من اجل عم النزول سطر جديد

ثم قمنا بالبحث عن الكي داخل الداتابيز (يدل الرمز ^ على اننا نقوم بالبحث عن الكلمة في اول كلمة موجودة من كل سطر)
في حال عدم وجوده نقوم بطلب ادخال ال value ثم تشفيره وضعه في متغير جديد record من اجل مظهر الداتابيز ثم نقوم بإضافة ال record الى داخل الداتا

اما في حال وجود key سابق يحمل نفس القيمة نقوم بطلب استدعاء التابع Addrecord

```
zain@zain-virtual-machine: ~/os
Q1 (first.os1db)
1) Add new record
2) Delete a record
3) Search for a record
4) Update a record
5) Exit
6) view data
-----
Select an option please: 1
please enter the key: zain
please enter the value: mhesn
Adding new key to the database (first.os1db)...
Done
Q1 (first.os1db)
1) Add new record
2) Delete a record
3) Search for a record
4) Update a record
5) Exit
6) view data
-----
Select an option please: 5
```

2. حذف سجل موجود في الداتا.

```
19 DeleteRecord(){
20 read -r -p "please enter the key that you want to delete: " key
21 #TODO
22 grep -w ^$key os1db
23 if [ $? -ne 0 ];
24 then
25 echo "Record does not exist"
26 else
27 grep -v -w $key os1db >>tmp.txt && mv tmp.txt os1db
28 echo "Delete value from the database (first.os1db)..."
29 echo "Done"
30 fi
31 }
```

نطلب من المستخدم بإدخال ال key الذي يريد حذف الحقل الخاص به نقوم بالبحث عن السجل الذي يحوي ال key من خلال تعليمة ال grep مع -w التي تفيد في تحديد الكلمة المدخلة بدون زيادة او نقصان لعدم حدوث مشاكل في حال وجود key مثل 12 و 1 و 123 ثم نختبر اذا كانت ال key موجودة في حال عدم وجودها نطبع رسالة ان ال key غير موجودة اما في حال وجودها نقوم بعمل تعليمة grep -v -w والتي تفيد في احضار جميع اسطر الملف عدا السطر الذي يحوي على الكلمة المدخلة ثم نقوم بوضعها في ملف tmp.txt من خلال >> ومن ثم نقوم بنقل محتويات الملف tmp.txt الى ملف الداتا للتعديل على

ملف الداتا الأساسي عن طريق تعليمة ال mv التي تقوم بنقل محتويات ملف لملف اخر

```
Q1 (first.os1db)
1) Add new record
2) Delete a record
3) Search for a record
4) Update a record
5) Exit
6) view data
-----
Select an option please: 2
please enter the key that you want to delete: sa
sa :ZGFkYWZzZHNkZmhy
Delete value from the database (first.os1db)...
Done
Q1 (first.os1db)
```

3. البحث عن حقل في الداتا.

```
32 SearchRecord(){
33 read -r -p "please enter the value that you want to Search: " value1
34     #var=`grep -n $value os1db`
35     value=`echo -n $value1 | base64 `
36     grep $value os1db
37     if [ $? -ne 0 ];
38     then
39         echo "Record does not exist"
40     else
41         echo "The word in this records"
42     fi
43 }
```

نقوم بالبحث عن طريق ال value فقط يقوم المستخدم بإدخال القيمة التي يريد البحث عنها ثم نقوم بتشفيرها والبحث عنها ضمن الداتا في عامود ال value في حال وجدنا القيمة نطبع الحقول المشابهة وفي حال القيمة غير موجودة نقوم بطباعة ان القيمة غير موجودة

```
Q1 (first.os1db)
1) Add new record
2) Delete a record
3) Search for a record
4) Update a record
5) Exit
6) view data
-----
Select an option please: 3
please enter the value that you want to Search: zain
12 : emFpbG==
The word in this records
```

4. تحديث قيمة حقل موجود في الداتا.

```
53 # set ui
54 UpdateRecord(){
55 read -r -p "please enter the key that you want to Update: " key
56
57     #TODO
58     grep -w ^$key os1db
59     if [ $? -ne 0 ];
60     then
61         echo "Record does not exist"
62     else
63         #TODO `
64         #tmp=`grep -v $key os1db`
65         var=`grep -w ^$key os1db | sed 's/.*:/'`
66         read -r -p "please enter the value: " value1
67         value=`echo -n $value1 | base64 `
68         sed -i "s/$var/$value/" os1db
69         #TODO
70         echo "Update value from the database (first.os1db)..."
71         echo "Done"
72     fi
73 }
74
```

في البداية نبحث عن ال key التي يقوم المستخدم بإدخالها مع التأكد بنفس الطريقة في تابع الحذف ثم نختار اذا كان ال key موجود في حال وجوده نقوم بتخزين قيمة ال value المقابلة له عن طريقة تعليمة ال grep و sed حيث تقوم تعليمة grep بالبحث عن الحقل كامل اما بتعليمة ال sed نقوم باستخراج القيمة التي تلي (:) ثم نطلب ادخال القيمة الجديدة نقوم بتشغيلها ومن خلال تعليمة ال sed نقوم بالبحث عن القيمة القديمة من ومن ثم استبدالها بالجديدة

```
Q1 (first.os1db)
1) Add new record
2) Delete a record
3) Search for a record
4) Update a record
5) Exit
6) view data
-----
Select an option please: 4
please enter the key that you want to Update: 12
12 : emFpbG==
please enter the value: 255
Update value from the database (first.os1db)...
Done
Q1 (first.os1db)
```

السؤال الثاني:

1. النسخ احتياطي:

```
#!/
read -r -p "Enter the backup name (with file extentsion .zip or .gz): " input
if echo "$input" | grep -q ".zip"; then
zip -r $input os1db
elif echo "$input" | grep -q ".tar"; then
tar cvf $input os1db
elif echo "$input" | grep -q ".gz"; then

foo=${input%".gz"}
cp os1db $foo
gzip $foo
fi
echo "Backup Done (/home/zain/os1/backups/backup/$input)"
;;
```

نطلب من المستخدم ادخال الاسم الذي يريده ثم نقوم بالمرور على مجموعة شروط نقوم من خلالها بالتحقق من لاحقة الملف المدخل من خلال تعليمة `grep -q` التي تقوم باختبار اذا كان النص يحوي الكلمة المراد البحث عنها بالنسبة للضغط `.gz`. نقوم باستخراج الاسم المدخل دون اللاحقة بمساعدة `%` ومن ثم نقوم بعمل نسخة لملف الداتا بالاسم الجديد قم نقوم بضغطه

```
Q2 backup / restore (first.os1db)
1) Backup the database
2) restore a database
3) Enabel automatic backups
4) Exit
-----
Select an option please: 1
Enter the backup name (with file extentsion .zip or .gz): b1.zip
adding: os1db (deflated 20%)
Backup Done (/home/zain/os1/backups/backup/b1.zip)
Q2 backup / restore (first.os1db)
1) Backup the database
2) restore a database
3) Enabel automatic backups
4) Exit
-----
```

2. استعادة النسخ:

```
~/
read -r -p "Enter the backup name (with file extentsion .zip or .gz): " input
if echo "$input" | grep -q ".zip"; then
unzip $input
elif echo "$input" | grep -q ".tar"; then
tar xf $input
elif echo "$input" | grep -q ".gz"; then
gunzip $input
foo=${input%".gz"}
cp -f $foo os1db
rm $foo
fi
::
```

نطلب ادخال اسم الملف المراد استعادته ثم نقوم بفك ضغطه حسب اللاحقة الخاصة به ويتم معرفة اللاحقة كما في الطلب الأول وبالنسبة للضغط بصيغة .gz نقوم بالتعامل مع نفس الطلب الأول لكن مع إضافة تعليمة لنقل الملف المستخرج واستبدال محتوياته بملف الداتا .

```
Backup done (/home/zain/os1/backups/backup/os.zip)
Q2 backup / restore (first.os1db)
1) Backup the database
2) restore a database
3) Enabel automatic backups
4) Exit
-----
Select an option please: 2
Enter the backup name (with file extentsion .zip or .gz): ds.zip
Archive: ds.zip
replace os1db? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflating: os1db
Backup done (/home/zain/os1/backups/backup/os.zip)
```

3. نسخ احتياطي اوتوماتيكي:

```
read -r -p "please enter your backup schadlule (daily ,weekly or monthly): " x
read -r -p "Number of max backup files to keep: " y
#TODO
if [ $x = "daily" ]; then
crontab -l >cn
echo "0 0 * * * /home/zain/os1/backup_auto.sh" >> cn
crontab cn
rm cn
echo "find /opt/backup -type f -mtime $y -delete" >> backup_auto.sh
#TODO
elif [ $x = "weekly" ]; then
crontab -l >cn
echo "0 0 * * 0 /home/zain/os1/backup_auto.sh" >> cn
crontab cn
rm cn
echo "find /opt/backup -type f -mtime $(( $y * 7 )) -delete" >> backup_auto.sh
#TODO
elif [ $x = "monthly" ]; then
crontab -l >cn
echo " 0 0 1 * * /home/zain/os1/backup_auto.sh" >> cn
crontab cn
rm cn
echo "find /opt/backup -type f -mtime $(( $y * 365 )) -delete" >> backup_auto.sh
fi
::
```

```

backup.sh
1 filename="backup_`date +%d`_`date +%m`_`date +%Y`";
2
3 cp -pvi /home/zain/os1/"os1db" /home/zain/os1/backups/"$filename.bak"
4
5 find /opt/backup -type f -mtime 2 -delete

```

نقوم بإدخال التصنيف الذي نريده واختباره ضمن شروط ثم نقوم بعمل فتح لملف الكرون ووضع محتواه في ملف نسمة cn ثم نقوم بإضافة تعليمة الكرون التي تقوم بتشغيل ملف ال backup_outo.sh الذي وظيفته هي عمل نسخة احتياطية من خلال تعليمة ال cp -pvi التي تقوم باستخراج ملف صيغته bak. و من ثم نقوم بإسناد محتوى ملف ال cn الى الكرون وحذف ال cn ومن ثم كتابة تعليمة التي تقوم بإيجاد الملفات حسب التاريخ المحدد ومن ثم حذفها .

```

Q2 backup / restore (first.os1db)
1) Backup the database
2) restore a database
3) Enabel automatic backups
4) Exit
-----
Select an option please: 3
please enter your backup schadlule (daily ,weekly or monthly): daily
Number of max backup files to keep: 2
# For more information see the manual pages of crontab(5)
#
# m h dom mon dow   command
* 12 * * * /home/zain/os1/system_stats.sh
0 0 * * * /home/zain/os1/backup_auto.sh

```


السؤال الثالث:

```
1 path=/opt/monitoring1.txt
2 cunt=0
3 date=$(date)
4 echo "#####" >> $path
5 echo $date >> $path
6 echo "#####" >> $path
7
8 while [ $cunt -lt 720 ]
9 do
10     cunt=`expr $cunt + 1`
11
12     # Get the current usage of CPU and memory
13     cpuUsage=$(top -bn1 | awk '/Cpu/ { print $2}')
14     memUsage=$(free -m | awk '/Mem/{print $3*100/$3}')
15     diskUsage=$(df -h | awk 'NF=="/"{print $3/$2,$5}')
16     # Print the usage
17     echo "CPU Usage: $cpuUsage%" >>$path
18     echo "Memory Usage: $memUsage %" >>$path
19     echo "Disk Usage: $diskUsage" >>$path
20     echo " " >>$path
21     echo " " >>$path
22     echo " " >>$path
23
24     # Sleep for 5 second
25     sleep 5
26 done
27 #DiskUsage= df -h | awk 'NF=="/"{printf "Disk Usage: %d/%dGB (%s)\n", $3,$2,$5}'
28 #CpuUsage= top -bn1 | grep load | awk '{printf "CPU Load: %.2f\n", $(NF-2)}'
29 echo "#####" >> $path
```

```
#
# For more information see the manual pages
#
# m h dom mon dow   command
* 12 * * * /home/zain/os1/system_stats.sh
```

تم جدولة تشغيل الملف كل يوم على الساعة 12:00 لمدة ساعة كاملة كل 5 ثواني وذلك عن طريق خدمة ال cron التي تؤمن لنا جدولة الاوامر حسب توقيت زمني محدد.

حيث تم بالصورة اعلاه تحديد الوقت المناسب يوميا على الساعة 12 وحلقة ال while. التي تحوي اوامر حساب ال CPU وال MEMORY وال Disk Usage بالإضافة لتابع ال sleep الذي يحقق لنا الانتظار 5 ثواني .

وبالتالي تم تحقيق الامر الذي ينص على كتابة استخدام كل من الcpu، وال memory وال disk useage من الجهاز كل 5 ثواني لمدة ساعة كاملة. بدءاً من ال 12 ليلاً.

حيث تم تخزين الخرج كاملا في ملف monitoring.txt "حيث ان مساره موجود في ملف ال opt " مع الوقت الذي يدل على هذا الخرج وذلك لان الملف سيتم تشغيله يوميا وبالتالي نحن بحاجة لاستخدام تعليمة الdate لتظهر لنا تاريخ اليوم .

السؤال الرابع:

1. تنزيل وتحميل سيرفر ال Ftp:

```
sudo apt update  
sudo apt install vsftpd
```

نقوم بتحديث الحزم وتنزيل السيرفر

```
sudo ufw status
```

نقوم بالتحقق من جدار الحماية لكلي والا فقد يحظر جدار الحماية حركة البيانات

```
1. $ sudo ufw allow 20/tcp  
2. $ sudo ufw allow 21/tcp  
3. $ sudo ufw allow 990/tcp  
4. $ sudo ufw allow 40000:50000/tcp  
5. $ sudo ufw status
```

نقوم بفتح المنافذ باستخدام هذه التعليمات ونقوم بإنشاء ملف ftp داخل اليوزر الحالي

```
sudo chown nobody:nogroup /home/ftpuser/ftp  
sudo chmod a-w /home/ftpuser/ftp
```

ومن ثم إعطاء الصلاحية للملف من خلال التعليمتين

```
sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bak
```

نقوم بعمل نسخ احتياطي لملف التكوين قبل التعديل عليه لإعطاء صلاحية ليوزر واحد

نقوم بفتح الملف وكتابة إضافة التعليمات التالية اليه

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
chroot_local_user=YES
user_sub_token=$USER
local_root=/home/$USER/ftp
pasv_min_port=40000
pasv_max_port=50000
userlist_enable=YES
userlist_file=/etc/vsftpd.userlist
userlist_deny=NO
$echo"zain" | sudo tee -a /etc/vsftpd.userlist
```

ثم نقوم بإضافة اليوزر المراد .

2. رفع الملفات على السيرفر واحضارها:

```
1 putMlteFile(){
2 echo "please Enter the files you want to upload from our pc to the ftp server:"
3 read -a ALL_FILES
4 ftp -inv $HOST <<EOF
5 user $USER $PASSWORD
6 cd files
7 mput ${ALL_FILES[@]}
8 bye
9 EOF
10 }
11 getMlteFile(){
12 echo "please Enter the files you want to download from the FTP server to our pc:"
13 read -a ALL_FILES
14 ftp -inv $HOST <<EOF
15 user $USER $PASSWORD
16 cd files
17 mget ${ALL_FILES[@]}
18 bye
19 EOF
20 }
```

من اجل رفع ملف نقوم بإدخال الملفات ومن ثم نقوم بالدخول الى سيرفر ال ftp من خلال ادخال الهوست ومن ثم اسم اليوزر وكلمة السر وننتقل الى الفولدر الذي يحوي على الملفات ومن ثم رفع الملفات من خلال تعليمة

mput وبنفس الطريقة احضار الملفات من خلال تعليمة mget

```
31
32 putFolder(){
33 echo "please Enter the folder you want to upload from our pc to the ftp server:"
34 read -a Folde
35 zip -r $Folde.zip $Folde
36 ftp -inv $HOST <<EOF
37 user $USER $PASSWORD
38 cd files
39 mput $Folde.zip
40 bye
41 EOF
42 sudo unzip /home/zain/ftp/files/$Folde.zip -d /home/zain/ftp/files/
43 password $PASSWORD
44 sudo rm /home/zain/ftp/files/$Folde.zip <<EOF1
45 password $PASSWORD
46 EOF1
47 rm $Folde.zip
48 }
49 getFolder(){
50 echo "please Enter the files or folder you want to download from the FTP server to our pc:"
51 read -a Folde
52
53 cd /home/zain/ftp/files/
54 password $PASSWORD
55 sudo zip -r $Folde.zip $Folde
56 password $PASSWORD
57 cd /home/zain/os1/
58 ftp -inv $HOST <<EOF
59 user $USER $PASSWORD
60 cd files
61 get $Folde.zip
62 bye
63 EOF
64 unzip $Folde.zip
65 sudo rm /home/zain/ftp/files/$Folde.zip
66 password $PASSWORD
67 rm $Folde.zip
68 }
```

بالنسبة للمجلدات نقوم بالبداية بضغط المجلد مع الحفاظ على نفس الاسم
ومن ثم نفس الخطوات السابقة ونقوم برفع او احضار المجلد ومن ثم نقوم
بحذف النسخ المضغوطة وفك الضغط للاحتفاظ بالمجلدات