

上課小考

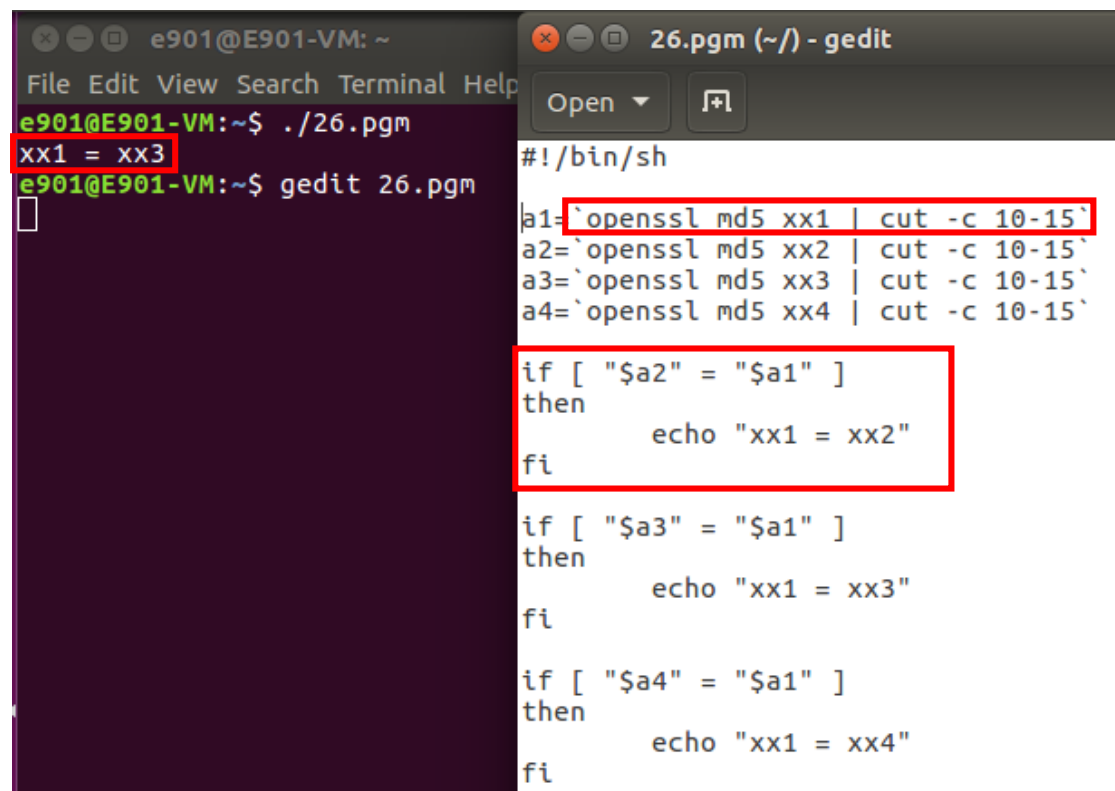
創四個檔案

1. `echo 11 > xx1`
2. `echo 11 > xx2`
3. `echo 11 > xx3`
4. `echo 11 > xx4`

並寫一個程式(.pgm)找出一樣的檔案

方法一：用 if 來判斷

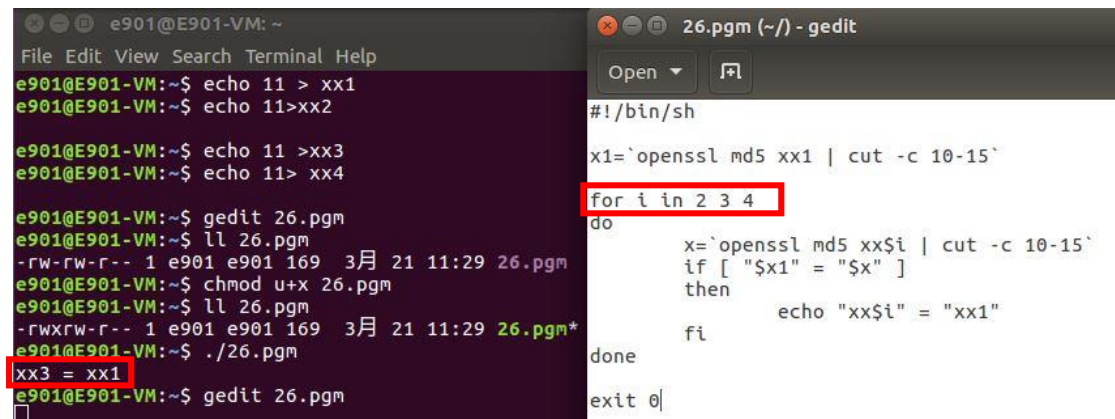
1. 先將 4 個檔案變成數位指紋，並用 `cut` 擷取部分，分別存入 `a1,a2,a3,a4` 中
2. 並用 if 判斷是否為一樣的數位指紋



```
e901@E901-VM: ~  
File Edit View Search Terminal Help  
e901@E901-VM:~$ ./26.pgm  
xx1 = xx3  
e901@E901-VM:~$ gedit 26.pgm  
[  
26.pgm (~/) - gedit  
Open [v] [F1]  
#!/bin/sh  
a1=`openssl md5 xx1 | cut -c 10-15`  
a2=`openssl md5 xx2 | cut -c 10-15`  
a3=`openssl md5 xx3 | cut -c 10-15`  
a4=`openssl md5 xx4 | cut -c 10-15`  
  
if [ "$a2" = "$a1" ]  
then  
    echo "xx1 = xx2"  
fi  
  
if [ "$a3" = "$a1" ]  
then  
    echo "xx1 = xx3"  
fi  
  
if [ "$a4" = "$a1" ]  
then  
    echo "xx1 = xx4"  
fi
```

方法二：使用 if 判斷加上 for 迴圈

1. i 的範圍設 2、3、4
2. 用 for 迴圈跑，比對一樣的印出



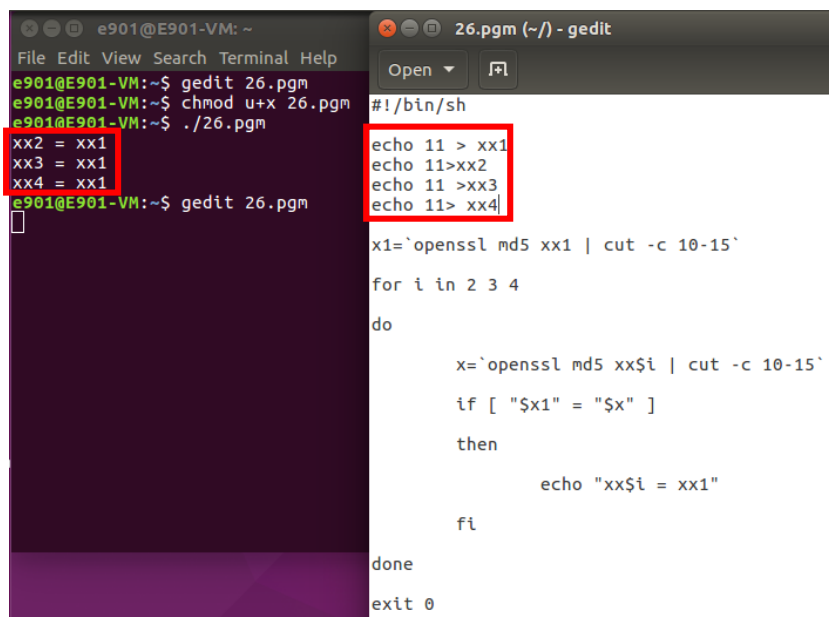
The screenshot shows a terminal window on the left and a gedit editor on the right. In the terminal, the user sets variables `xx1`, `xx2`, `xx3`, and `xx4` to the value `11`. They then create a file `26.pgm`, set permissions, and run it. The gedit editor shows the script `26.pgm` which uses a `for` loop to iterate over values 2, 3, and 4, calculating MD5 hashes and comparing them to `xx1`.

```
e901@E901-VM: ~  
File Edit View Search Terminal Help  
e901@E901-VM:~$ echo 11 > xx1  
e901@E901-VM:~$ echo 11>xx2  
  
e901@E901-VM:~$ echo 11 >xx3  
e901@E901-VM:~$ echo 11> xx4  
  
e901@E901-VM:~$ gedit 26.pgm  
e901@E901-VM:~$ ll 26.pgm  
-rw-rw-r-- 1 e901 e901 169 3月 21 11:29 26.pgm  
e901@E901-VM:~$ chmod u+x 26.pgm  
e901@E901-VM:~$ ll 26.pgm  
-rwxrwx-r-- 1 e901 e901 169 3月 21 11:29 26.pgm*  
e901@E901-VM:~$ ./26.pgm  
xx3 = xx1  
e901@E901-VM:~$ gedit 26.pgm
```

```
#!/bin/sh  
  
x1=`openssl md5 xx1 | cut -c 10-15`  
  
for i in 2 3 4  
do  
    x=`openssl md5 xx$i | cut -c 10-15`  
    if [ "$x1" = "$x" ]  
    then  
        echo "xx$i" = "xx1"  
    fi  
done  
  
exit 0
```

補充：

1. 在 if 判斷中[]要空格，並且變數最好用""，使用比較安全
2. ">"前有沒有空格的差異，是根據不同版本的系統與不同的 shell 有所差異，Ubuntu 的/bin/sh 對">"前後有沒有空格的解讀都一樣，但/bin/bash 對">"前若沒有空格，會有問題，不會將資料寫入，也就是只會產生一個空檔



This screenshot shows the same setup as the previous one, but with the script `26.pgm` executed. The terminal now shows the output of the script, which prints `xx2 = xx1`, `xx3 = xx1`, and `xx4 = xx1`. The gedit editor shows the script with the `echo` statements highlighted, showing the values being compared.

```
e901@E901-VM: ~  
File Edit View Search Terminal Help  
e901@E901-VM:~$ gedit 26.pgm  
e901@E901-VM:~$ chmod u+x 26.pgm  
e901@E901-VM:~$ ./26.pgm  
xx2 = xx1  
xx3 = xx1  
xx4 = xx1  
e901@E901-VM:~$ gedit 26.pgm
```

```
#!/bin/sh  
  
echo 11 > xx1  
echo 11>xx2  
echo 11 >xx3  
echo 11> xx4  
  
x1=`openssl md5 xx1 | cut -c 10-15`  
  
for i in 2 3 4  
do  
  
    x=`openssl md5 xx$i | cut -c 10-15`  
  
    if [ "$x1" = "$x" ]  
    then  
        echo "xx$i = xx1"  
    fi  
done  
  
exit 0
```

上課內容

`openssl md5 xx1 | cut -d " " -f 2`

欄位之間用空格當作區隔，`-f 2` 是指取第二個欄位的內容

```
e901@E901-VM: ~  
e901@E901-VM:~$ openssl md5 xx1  
MD5(xx1)= 166d77ac1b46a1ec38aa35ab7e628ab5  
e901@E901-VM:~$ openssl md5 xx1 | cut -d " " -f 2  
166d77ac1b46a1ec38aa35ab7e628ab5
```

用 **rc4** 加密解密

加密：`openssl rc4 -e -in xx1 -out xx1.rc4 -k 123`

解密：`openssl rc4 -d -in xx1.rc4 -out xx1.ans -k 123`

`-e` 加密(最好要加上) `-d` 解密

`-k` 後面加密碼，就不會出現輸入密碼

```
e901@E901-VM: ~  
File Edit View Search Terminal Help  
e901@E901-VM:~$ openssl rc4 -e -in xx1 -out xx1.rc4 -k 123  
e901@E901-VM:~$ openssl rc4 -d -in xx1.rc4 -out xx1.ans -k 123  
e901@E901-VM:~$ openssl md5 xx1  
MD5(xx1)= 166d77ac1b46a1ec38aa35ab7e628ab5  
e901@E901-VM:~$ openssl md5 xx1.ans  
MD5(xx1.ans)= 166d77ac1b46a1ec38aa35ab7e628ab5
```

產生私鑰及公鑰

私鑰：`openssl genrsa -out 26.pri`

公鑰：`openssl rsa -in 26.pri -pubout -out 26.pub`

```
e901@E901-VM: ~  
e901@E901-VM:~$ openssl genrsa -out 26.pri  
Generating RSA private key, 2048 bit long modulus  
.....+++  
.....+++  
e is 65537 (0x10001)  
e901@E901-VM:~$ openssl rsa -in 26.pri -pubout -out 26.pub  
writing RSA key
```

私鑰

```
e901@E901-VM:~$ cat 26.pri
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAuu9FBjIqmiwEfumEDLTfQJsprwv9PDbCFFCkjmWdYhJA10kv
c3AIhU4Uffoist5EEQgrJU4GvIJXNl4E+QvA6Bkuj6p8ocgytF4sqMKbiEw1FJGI
85mZNNgHNLoQPjwM2gFEKJrvWHC/j2y3DN4KrpagJ2FbVlgwkb0KyPDj6KzKsA3
bvbxXErMqlhF33rQucTWQNXZpRB/YjLAHRT1foCDSMf9lR0SgJd7R0/h+bjX8WZq
7ik//+a+i2ESJ0diuPTbap+nzGU7EUvsLpyx0hsxnbPkBx1svvL0TOIJgsQGG7B4
/b20H+C+fGqM50ExiwKw/B4Epc9u3H8s3hbMRQIDAQABAoIBAEqA4nm9tM0Njh0m
sJLTHh2nv5tQNm2mNOYmbc8zkLHVywhqJuPHckz3YGI542h9ph7xBHMNgg+rFo
l4Lq9E+M+cm4u05jrvkbNg199cyyFUSTAsKxM0S7IXpO/or501eY9PrLEmLZk27
pLaRdefo0URIF4hd/oaHM5JdAxdW1TdFntRKNokTrS0246S6L/ecT5lyIxxjfxe
89fF3XOHQo/BAjuaFzqoTzf2VSDw8F84d7TN5GGDT3LW4Pu4Zhx3B+X0+8w0lo7k
/rj2pIV0ycZLtkiipUULRSQ33XjVEYEn3a2CaLDqUoibxCwPKgEA5pmppBkXVoOf
5W7IOWEcGYEA70rcON+51Wh+ZiW7eHnGq/yowPjter4y0k2i46nqFtItBZ/JLLUb
FVHd0KTFKbeKMT+1TNTGG/uHA+hb2vasDIeb5Vt1YFjqMwWvE6qYuiT6cHJRXt
ldJia7h7S5Rkav+FYI13NjN4Eivk45ntm9C4vApg7+8TQI25szzwIMCgYEAyf3I
/UvK0+xxkqNcva6jNtLWigOLyC57iCxiUjM1o4pUXz9skRWiv2/00XGjwo19jsjE
fWpSfM60Va9nY7m6Clw8fk3p5eYViDxrqC1ifSxWD7BakxJn9EJzuqFnitZd5vh
/7L3Hpl8TT1GLL3IdcJMjvUyCLVPGRZTU0DLZcCgYAMDW6Ing8QIj670mnVGSzY
IwRBcnhyVBauDKEKv+weuRZ9QMG10olpeK62HivDMYk780Vi5lhjWF2ufldP6nRw
E9qfBuXJa+Dgfr97r1Xyph3du3uLkXABHHDRVhM/pw7suo8InHgek805+02t/niu
pJDtqvbyOfL0until4dDQKBgA3KD4CLPgt9pdmoUVP5o5biifDeERrXYyOIVCX8
pPr/JIdkEwtMiNG81keiFGXhtuKafhrC3+a2fhEQBN30taC/w4AhXVVJmhK3KQic
AfQ9pJpplaVYqLQYfUIOZr0ppqTttVNYQWtztGVpEiF5FjnvLieJwhrwB2uaBiaq
Jz99AoGAfjC3dBRxFJyrP7H2WnMniHa1HvHqH8vkr40LPuM7S3rzL0IqE2CKxzB
Ibl47sa/ZFBA0VwCROcspOegPR4d9dkjCnjdVpE36yxJkNDDkTwdQSh7SCfgq9/
JYGV5ta0tDsrWajEjdPf8JVBwl+Z1HbyufHGzwddeH0W3t8sJYk=
-----END RSA PRIVATE KEY-----
```

公鑰

```
e901@E901-VM: ~
File Edit View Search Terminal Help
e901@E901-VM:~$ cat 26.pub
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuu9FBjIqmiwEfumEDLTf
QJsprwv9PDbCFFCkjmWdYhJA10kvx3AIhU4Uffoist5EEQgrJU4GvIJXNl4E+QvA
6Bkuj6p8ocgytF4sqMKbiEw1FJGI85mZNNgHNLoQPjwM2gFEKJrvWHC/j2y3DN4K
rpagJ2FbVlgwkb0KyPDj6KzKsA3bvbxXErMqlhF33rQucTWQNXZpRB/YjLAHRT1
foCDSMf9lR0SgJd7R0/h+bjX8WZq7ik//+a+i2ESJ0diuPTbap+nzGU7EUvsLpyx
0hsxnbPkBx1svvL0TOIJgsQGG7B4Vb20H+C+fGqM50ExiwKw/B4Epc9u3H8s3hbM
RQIDAQAB
-----END PUBLIC KEY-----
```

補充：

因為沒有使用-out 參數，所以產生出來的私鑰會出現在螢幕，不會寫入檔案，本案例，因為沒有用-out，系統把 26.pri，當成是另一個參數 26，所以產生了 26 個 bit 的私鑰到螢幕上(預設是產生 512bit 私鑰)

```
e901@E901-VM: ~
File Edit View Search Terminal Help
e901@E901-VM:~$ openssl genrsa 26.pri
Generating RSA private key, 26 bit long modulus
+++++
+++++140207023728280:error:04081078:rsa routines:RSA_BUILTIN_KEYGEN:key size too
small:rsa gen.c:175:
e901@E901-VM:~$ cat 26.pri
cat: 26.pri: No such file or directory
```


公鑰加密，私鑰解密

加密：openssl rsautl -encrypt -pubin -inkey 26.pub -in xx1 -out xx1.rsa

解密：openssl rsautl -decrypt -inkey 26.pri -in xx1.rsa -out xx1.ans

```
e901@E901-VM: ~
File Edit View Search Terminal Help
e901@E901-VM:~$ openssl rsautl -encrypt -pubin -inkey 26.pub -in xx1 -out xx1.rsa
e901@E901-VM:~$ cat xx1
11
e901@E901-VM:~$ cat xx1.rsa
y+ 1d+r6og@0 0w  &Jf+ys| 5^KV<EY+4
5+6Zt>(\)%+E+s%  $
w/u5+Z8+W"o+ \;+Y+8&+l+m
d++++G+Q+++++8p|/^
4F+x+n">{1.Y+++ "Q+V+++8+++U+x+|+
[+_L)e901@E9
01-VM:~$ openssl rsautl -decrypt -inkey 26.pri -in xx1.rsa -out xx1.ans
e901@E901-VM:~$ openssl md5 xx1
MD5(xx1)= 166d77ac1b46a1ec38aa35ab7e628ab5
e901@E901-VM:~$ openssl md5 xx1.ans
MD5(xx1.ans)= 166d77ac1b46a1ec38aa35ab7e628ab5
```

簽章私鑰，公鑰解簽章

簽章：`openssl rsautl -sign -inkey 26.pri -in xx1 -out xx1.sign`

解簽章：`openssl rsautl -verify -pubin -inkey 26.pub -in xx1.sign -out xx1.a`

```
e901@E901-VM: ~
e901@E901-VM:~$ openssl rsautl -sign -inkey 26.pri -in xx1 -out xx1.sign
e901@E901-VM:~$ openssl rsautl -verify -pubin -inkey 26.pub -in xx1.sign -out xx1.a
e901@E901-VM:~$ cat xx1.sign
MI
-----BEGIN RSA SIGNATURE-----
MII
-----
e901@E901-VM:~$ cat xx1.a
-----BEGIN RSA SIGNATURE-----
MII
-----
e901@E901-VM:~$ openssl md5 xx1
MD5(xx1)= 166d77ac1b46a1ec38aa35ab7e628ab5
e901@E901-VM:~$ openssl md5 xx1.a
MD5(xx1.a)= 166d77ac1b46a1ec38aa35ab7e628ab5
```

補充指令：

CPU 內容：cat /proc/cpuinfo (cat 後要加空白鍵)

mem : cat /proc/meminfo (cat 後要加空白鍵)

WC：統計指定檔案中的位元組數、字數、行數

抓用幾顆 CPU : `cat /proc/cpuinfo | grep processor | wc -l`

```
e901@E901-VM: ~  
File Edit View Search Terminal Help  
e901@E901-VM:~$ cat /proc/cpuinfo | grep processor | wc -l  
2
```

抓 CPU 的規格 : `cat /proc/cpuinfo | grep "model name" | head -1 | cut -d " " -f 3-7`

```
e901@E901-VM: ~  
File Edit View Search Terminal Help  
e901@E901-VM:~$ cat /proc/cpuinfo | grep "model name" | head -1 | cut -d " " -f 3-7  
Intel(R) Core(TM) i7-3770 CPU @
```

增加使用者

新增 : `useradd -m xx1`(使用者名稱)

密碼 : `echo "xx1:xx1" | chpasswd`

```
e901@E901-VM:~$ sudo passwd root  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
e901@E901-VM:~$ su -  
Password:  
root@E901-VM:~# useradd -m xx1  
root@E901-VM:~# useradd -m xx2  
root@E901-VM:~# echo "xx1:xx1" | chpasswd  
root@E901-VM:~# exit  
logout  
e901@E901-VM:~$  
e901@E901-VM:~$ su - xx1  
Password:  
xx1@E901-VM:~$ id  
uid=1001(xx1) gid=1001(xx1) groups=1001(xx1)  
xx1@E901-VM:~$
```