

## Senza Soluzioni

- Scrivere i due seguenti semplici script in bash shell:
  - a. (1.5pt.)uno script che cancella un file dando il suo inode. Lo script e' attivato cosi':

```
irm <inode>
```
  - b. (1.5pt.)uno script che elimina un processo dando il suo nome. Lo script e' attivato cosi':

```
ikill <nome>
```

Si richiede che gli script, oltre alla prima riga `#!/bin/bash`, siano formati da due righe max ciascuno

- Scrivere uno script che cerca, in una stringa fornita come argomento, la posizione di una sottostringa di 3 caratteri data come argomento. Lo script e' attivato cosi':

```
search <stringa> <sottostringa>
```

Si assuma che non si sappia a priori se la sottostringa sia contenuta nella stringa data, nel qual caso lo script restituisce 0, e che c'e' eventualmente solo una sottostringa nella stringa data.

In un file system della famiglia UNIX si consideri i file `/home/mgiorgio/dati.txt`, `/home/mgiorgio/pippo.txt` creati dall'utente `mgiorgio`. Facendo `ls -l` emerge che le protezioni delle directory `home`, `mgiorgio` e del file `dati.txt` sono le seguenti

```
r-xr-xr-x home
rwxr-xr-x mgiorgio
rwxr-xr-- dati.txt
rwxrwx-w- pippo.txt
```

Se l'utente `giulio` appartiene allo stesso gruppo di `mgiorgio`, può eseguire i seguenti comandi?:

```
$ls /home/mgiorgio
```

```
$ln -s /home/mgiorgio/dati.txt /home/mgiorgio/dati1.txt
```

```
$rm /home/mgiorgio/dati.txt
```

```
$cat /home/mgiorgio/dati1.txt > pippo.txt
```

- Può un altro utente non appartenente al gruppo di `mgiorgio`, eseguire gli stessi comandi?

- Spiegare il risultato di questi comandi:

(2pt)

```
$cat>a
```

```
stringa
```

```
^EOF
```

```
$mkdir test
```

```
$cd test
```

```
$ln ../a a1
```

```
$ln -s ../a a2
```

```
$cd ..
```

```
$rm a
```

```
$cd test
```

```
$cat na1
```

```
$cat na2
```

Scrivere una linea di comando Unix per scrivere le linee 10...20 (comprese) di un file (2pt)

*(Questa linea può essere scritta usando head e tail. Ricorda:*

*tail -n 5 file stampa le ultime 5 righe del file, mentre tail -n +5 file stampa le ultime righe partendo dalla 5a riga fino in fondo.*

*head -n 5 file stampa le prime 5 linee del file)*

- Scrivere uno script per scrivere le linee da n1 a n2 di un file senza usare tail/head (2pt)

- Trovare gli errori del seguente script: (1pt)

```
if [ "nome" = "" ]
then
    print "dare il nome del file: "
    read
fi
sort | wc -l
```

- Scrivere uno script per trovare il numero complessivo di linee dei file di testo data0.txt, data1.txt, ..., data15.txt (1.5pt)

- Per brevità diciamo che RUID, EUID significhino “Real User ID”, “Effective User ID”.

Si supponga che una directory di Unix contenga i seguenti file:

Filename	Owner	Group	SetUID	Owner	Group	Other
a10.txt	15	15	-	rw-	r--	---
b10.txt	15	99	-	---	rw-	rw-
c12.txt	75	75	-	rw-	rw-	---
c14.txt	25	25	-	rw-	r--	r--
dati.txt	25	99	-	--x	r-x	---
all.exe	0	75	y	--x	--x	--x

Si supponga inoltre che l'utente 15 sia nel gruppo 99 e che l'utente 25 sia nel gruppo 25. Rispondere alle seguenti domande:

- a. Quali file può leggere un processo con RUID=15, EUID=15? (1pt)
- b. Quali file può scrivere un processo con RUID=15, EUID=15? (1pt)
- c. Quali file può leggere un processo con RUID=25 e EUID=25 che esegue il programma all.exe? (1pt)
- d. Quali file può scrivere un processo con RUID=25 e EUID=25 che esegue il programma all.exe? (1pt)

- Scrivere uno script in Korn Shell per stampare i file speciali di tipo directory e i file speciali di tipo link simbolici.

[ricorda: i tipi dei file sono visualizzati con `ls -l`]

- Scrivere uno script in Korn Shell per copiare tutti i file con estensione .c nella sottodirectory

copiaprogetto. Se la sottodirectory non esiste, la si crei.

- Si supponga che un utente del sistema Unix esegua il comando `id` e subito dopo il comandi `ls`:

```
$id
uid=21132(mgiorgio) gid=10000(studenti)
$ ls -l
total 17354
-rw-r---wx 1 mgiorgio studenti 12249 May 8 2007 10
-rw-r---wx 1 mgiorgio studenti 7781 May 8 2007 11
-r--r--r-- 1 mgiorgio studenti 775 Jan 23 2007 1q
--w-r--r-- 1 mgiorgio studenti 1 Nov 16 2007 2
drwxr-xr-x 2 mgiorgio studenti 1024 Nov 26 2007 source
lrwxrwxrwx 1 mgiorgio studenti 3 Jan 27 2000 ss -> pse
srwxrwxrwx 1 mgiorgio studenti 0 Jan 25 17:05 nomesocket
```

Quale delle seguenti operazioni è permessa? (scrivere SI/NO a fianco della operazione)

```
$cat 10
$cat 2
$ls -l >10
$cat /etc/passwd
$cat /etc/shadow
```

Un altro utente del gruppo 10000 può eseguire le seguenti operazioni?

```
$rm 10
$rm 1q
$rm 2
$cp 10 11
$cp 1q 10
$rm /etc/passwd
$rm /etc/shadow
```

## Con Soluzioni

Scrivere un comando per listare i file della Home directory che hanno come prima lettera una a, come terza lettera una b e come ultima lettera un numero tra 0 e 9. (1pt)

Scrivere la/le righe di comandi Linux per visualizzare i file della directory corrente nel formato:  
                  proprietario    nomefile    tipofile  
ordinando la lista per nome del proprietario.

### Soluzione

Un modo è quello di usare queste righe:

```
$ls -l | cut -c1 > tipo  
$ls -l | cut -c15-21 > owner  
$ls -l | cut -c55- > nome  
$paste owner nome tipo | sort
```

Un modo molto più compatto è il seguente:

```
$ls -l | cut -c1,22-27,55- --output-delimiter=' ' | sort -k 2,2
```

In questo modo, si estraggono i campi tipofile-proprietario-nomefile, si separano con lo spazio e si ordinano per il secondo campo, il tutto in una riga unica. Questo comando però dà un ordinamento sbagliato dei campi, rispetto a quello richiesto. L'unica possibilità è dunque di usare i file come uscita temporanea. Usando il separatore ; si ha la seguente riga di comando:

```
$ls -l | cut -c1 > tipo; ls -l | cut -c15-21 > owner; ls -l | cut -c55- > nome; paste owner nome  
tipo | sort
```

L'utente Giorgio, con UID=1000 e GID=1000, acquisisce l'accesso a Linux. Nella sua HomeDirectory, è memorizzato il file `lista`, che contiene il codice eseguibile di un programma che lista il contenuto di file testuali (l'uso del comando è quindi del tipo: `$lista <file>`). Il file `lista` è dell'utente Giulio, con UID=1500 e GID=2000. Per vedere le caratteristiche del file `lista`, Giorgio effettua il comando:

```
$ls -l > dati
```

Facendo `$ls -l dati` Giorgio vede che il file `dati` è ovviamente di Giorgio ed ha le seguenti protezioni: 304.

- (2pt) Può Giorgio vedere il contenuto del file `dati` con il comando `$lista dati` ?  
spiegare il risultato.
- (2pt) Cosa succede se Giorgio scrive il comando `$lista dati` dopo che Giulio setta il bit SetUID del file `lista`? Spiegare il risultato

### Soluzione

a) Assunto che Giorgio possa eseguire il programma `lista`, appena il file cerca di leggere il file `dati` vede che il file `dati` è di Giorgio e che il processo `lista` ha l'EUID uguale a quello di Giorgio, che ha attivato il programma. Allora i bit di protezione sono quelli dell'owner, cioè `-wx`. Quindi il processo `lista` non può leggere il file.

b) Se il bit SetUID del file lista è settato, l'EUID del processo lista è pari a quello del proprietario del file lista, cioè Giulio. Quindi, l'EUID del processo è diverso dal proprietario del file dati. Inoltre, il l'EGID del processo è diverso. I bit di protezione sono quelli di other, cioè r---. Quindi in questo caso Giorgio può vedere il contenuto del file dati perchè il processo lista può leggere il file.