

Lab. di Sistemi Operativi

Esercitazioni proposte per le lezioni del 18 e 19/03/10

Utilizzando la shell di Linux e disponendosi in gruppi di due persone per ogni PC del laboratorio.

1. Creare una directory nella propria home di nome "miadir"
2. Invocare ls per visualizzare il contenuto di miadir
3. Invocare ls per visualizzare il nome e gli attributi della directory miadir
4. Rendiamo la directory miadir illeggibile da tutti
5. Invochiamo nuovamente ls per visualizzare la directory poi il suo contenuto
6. Rendiamo il file leggibile dal solo proprietario
7. Verificare cosa avviene al file prova settando a 1 e 0 i 12 bit dei permessi
8. Impostare il bit di eseguibilità di miadir a zero per tutti. Provare a spostare la directory corrente in miadir
9. Impostare il bit di eseguibilità di miadir a uno per tutti. Provare a spostare la directory corrente in miadir
10. Spostare la directory corrente al di fuori di miadir, utilizzando il link ".."
11. Eliminare la directory miadir
12. Creare nuovamente la directory miadir e popolarla con qualche file
13. Provare ad eliminare la directory miadir
14. Utilizzare il comando "rm" per eliminare miadir ed il suo contenuto.
15. Invocare il comando true e visualizzarne l'exit value
16. Invocare il comando false e visualizzarne l'exit value
17. Creare uno shell script che invochi il comando true e ne visualizzi l'exit value
18. Fare lo stesso con il comando false
19. Creare uno shell script che controlli l'exit value di true e stampi "vero" su stdout in caso affermativo
20. Creare la variabile d'ambiente "var1" ed assegnarle il valore "ciao"
21. Utilizzando la man page del comando test, invocarlo da linea di comando con gli argomenti necessari a verificare che la variabile d'ambiente "var1" sia uguale alla stringa ciao; visualizzare l'exit value
22. Assegnare alla variabile precedente il valore di stringa nulla o ""
23. Ripetere l'invocazione precedente del comando test ed analizzare il comportamento della shell. Con quali argomenti viene invocato il comando test?
24. Trovare un metodo per eseguire il test anche con stringhe vuote
25. Creare uno shell script che stampi su stdout il messaggio "vero" se il suo primo argomento è uguale alla stringa ciao.
26. Creare uno shell script che stampi su stdout il messaggio "vero" se il suo primo argomento è uguale alla stringa 123.

27. Si modifichi lo script precedente in modo che il test venga eseguito sulla rappresentazione numerica della variabile; cosa avviene se lo script viene invocato senza argomenti?
28. Aggiungere un controllo preventivo sulla presenza di almeno un argomento nella linea di comando
29. Visualizzare l'exit value di test quando incontra un errore, ad esempio in `test abc -eq 123`. Confrontarlo con l'exit value di condizione vera e di condizione falsa
30. Modificare lo script precedente in modo da visualizzare il messaggio "errore" in caso di argomento non numerico; si ridirezioni il messaggio di errore del comando `test` (standard error) su `/dev/null`
31. Possiamo migliorare il nostro `crea_script.sh` in modo che lo script generato esegua un controllo sul numero dei parametri dopo aver chiesto all'utente se esiste un numero minimo?

Soluzione

Esercitazioni proposte per le lezioni del 18 e 19/03/10

Utilizzando la shell di Linux e disponendosi in gruppi di due persone per ogni PC del laboratorio.

1. Creare una directory nella propria home di nome "miadir"

Soluzione:

```
mkdir miadir
```

2. Invocare ls per visualizzare il contenuto di miadir

Soluzione:

```
ls miadir
```

3. Invocare ls per visualizzare il nome e gli attributi della directory miadir

Soluzione:

```
ls -l -d miadir
ls -ld miadir
# Piu' opzioni, come -l e -d, possono essere normalmente composte
# in una opzione singola -ld
```

4. Rendiamo la directory miadir illeggibile da tutti

Soluzione:

```
chmod a-r miadir
```

5. Invochiamo nuovamente ls per visualizzare la directory poi il suo contenuto

Soluzione:

```
ls -ld miadir
ls miadir
```

6. Rendiamo il file leggibile dal solo proprietario

Soluzione:

```
chmod u+r miadir
ls -ld miadir
ls miadir
```

7. Verificare cosa avviene al file prova settando a 1 e 0 i 12 bit dei permessi

Soluzione:

```
chmod 00000 miadir
ls -ld miadir
ls miadir
chmod 00001 miadir
ls -ld miadir
ls miadir
chmod 00002 miadir
ls -ld miadir
ls miadir
```

```
...
chmod 0400 miadir
ls -ld miadir
ls miadir
```

8. Impostare il bit di eseguibilità di miadir a zero per tutti. Provare a spostare la directory corrente in miadir

Soluzione:

```
chmod a-x miadir
cd miadir
```

9. Impostare il bit di eseguibilità di miadir a uno per tutti. Provare a spostare la directory corrente in miadir

Soluzione:

```
chmod a+x miadir
cd miadir
```

10. Spostare la directory corrente al di fuori di miadir, utilizzando il link ".."

Soluzione:

```
cd ..
# In ogni directory, esiste un link di nome "." alla directory
# di livello superiore
```

11. Eliminare la directory miadir

Soluzione:

```
rmdir miadir
```

12. Creare nuovamente la directory miadir e popolarla con qualche file

Soluzione:

```
mkdir miadir
touch miadir/a
touch miadir/b
```

13. Provare ad eliminare la directory miadir

Soluzione:

```
rmdir miadir
```

14. Utilizzare il comando "rm" per eliminare miadir ed il suo contenuto.

Soluzione:

```
rm -Rf miadir
# Attenzione a questo comando!
```

15. Invocare il comando true e visualizzarne l'exit value

Soluzione:

```
true
echo $?
```

16. Invocare il comando false e visualizzarne l'exit value

Soluzione:

```
false
echo $?
```

17. Creare uno shell script che invochi il comando true e ne visualizzi l'exit value

Soluzione:

```
#!/bin/bash
true
echo $?
```

18. Fare lo stesso con il comando false

Soluzione:

```
#!/bin/bash
false
echo $?
```

19. Creare uno shell script che controlli l'exit value di true e stampi "vero" su stdout in caso affermativo

Soluzione:

```
#!/bin/bash
if true
then
    echo vero
fi
```

20. Creare la variabile d'ambiente "var1" ed assegnarle il valore "ciao"

Soluzione:

```
var1=ciao
```

21. Utilizzando la man page del comando test, invocarlo da linea di comando con gli argomenti necessari a verificare che la variabile d'ambiente "var1" sia uguale alla stringa ciao; visualizzare l'exit value

Soluzione:

```
test $var1 = ciao
echo $?
```

22. Assegnare alla variabile precedente il valore di stringa nulla o ""

Soluzione:

```
var1=""
```

23. Ripetere l'invocazione precedente del comando test ed analizzare il comportamento della shell. Con quali argomenti viene invocato il comando test?

Soluzione:

```
test $var1 = ciao
```

24. Trovare un metodo per eseguire il test anche con stringhe vuote

Soluzione:

```
test "$var1" = ciao
test B$var1 = Bciao
```

25. Creare uno shell script che stampi su stdout il messaggio "vero" se il suo primo argomento è uguale alla stringa ciao.

Soluzione:

```
#!/bin/bash
if test "$1" = ciao
then
    echo vero
fi
```

26. Creare uno shell script che stampi su stdout il messaggio "vero" se il suo primo argomento è uguale alla stringa 123.

Soluzione:

```
#!/bin/bash
if test "$1" = 123
then
    echo vero
fi
```

27. Si modifichi lo script precedente in modo che il test venga eseguito sulla rappresentazione numerica della variabile; cosa avviene se lo script viene invocato senza argomenti?

Soluzione:

```
#!/bin/bash
if test "$1" -eq 123
then
    echo vero
fi
```

28. Aggiungere un controllo preventivo sulla presenza di almeno un argomento nella linea di comando

Soluzione:

```
#!/bin/bash
if test $# -ge 1
then
    if test "$1" -eq 123
    then
        echo vero
    fi
fi
```

29. Visualizzare l'exit value di test quando incontra un errore, ad esempio in test abc -eq 123. Confrontarlo con l'exit value di condizione vera e di condizione falsa

Soluzione:

```

test abc -eq 123
echo $?
test 123 -eq 123
echo $?
test 124 -eq 124
echo $?

```

30. Modificare lo script precedente in modo da visualizzare il messaggio "errore" in caso di argomento non numerico; si ridirezioni il messaggio di errore del comando test (standard error) su /dev/null

Soluzione:

```

#!/bin/bash
if test $# -ge 1
then
    test "$1" -eq 123 2> /dev/null
    if test $? -eq 2
    then
        echo errore
    fi
fi

```

31. Possiamo migliorare il nostro crea_script.sh in modo che lo script generato esegua un controllo sul numero dei parametri dopo aver chiesto all'utente se esiste un numero minimo?

Soluzione:

```

#!/bin/bash
# crea_script: script per creare altri script
#
# Richiesta interattiva del nome del file da creare:
echo -n "Nome_dello_script_da_creato, _senza_.sh_"; read nomefile
# Aggiunta dell'estensione .sh:
nomefile=${nomefile}.sh
# Creazione del file (vuoto)
>$nomefile
# Modifica dei permessi: tutti lo possono eseguire, solo il proprietario puo'
# leggerlo o modificarlo
chmod 00711 $nomefile
# Aggiunta dello sharp-bang
echo "#!/bin/bash" >>$nomefile
# Richiesta del numero minimo di argomenti che lo script accetta
echo -n "Numero_minimo_di_argomenti, _0_per_nessun_controllo_"
read minargs
# Siamo sicuri che l'utente abbia immesso un numero?
test $minargs -eq 0 2>/dev/null
if test $? -eq 2
then
    echo "Valore_non_valido"
    exit
fi
if test $minargs -gt 0
then
    echo 'if test $# -lt ' $minargs >> $nomefile
    echo 'then' >> $nomefile
    echo '$\'t' 'echo "Numero_argomenti_errato"' >> $nomefile

```

```
echo $\t 'exit 1' >> $nomefile
echo 'fi' >> $nomefile
```

fi