Alcuni errori tipici

Frammento 2

Sulla gestione errori...

- Bisogna controllare l'esito dei comandi che possono fallire !
- Bisogna controllare che un parametro esista prima di accedere
  - testare sempre $# prima di accedere a $1 $2 etc
- Ritornare sempre un valore diverso da 0 se si e' verificato un errore
  - È la convenzione e va seguita altrimenti il vostro script non è usabile in modo naturale come gli altri comandi

Sulla gestione errori… (2)

- Bisogna verificare sempre i diritti e la natura dei file prima di manipolarli
  - Verificare di avere i diritti di lettura/scrittura/esecuzione prima di leggere/scrivere/eseguire
  - Verificare che un path passato come parametro corrisponda a un file se ci aspettiamo un file o una directory se ci aspettiamo una directory

- Le stampe di messaggi di errore devono essere effettuate su stderr
  - Per farlo si può usare la riderezione dei decrittori

```
echo ciccio 1>&2
```

Commenti

- Stesse considerazioni del primo frammento
- Non devono ripetere passo passo quello che è evidente dal codice!
- è preferibile:
  - scrivere un commento esauriente all'inizio della funzione, con eventuali note algoritmiche
  - commentare sinteticamente all'inizio di una blocco di comandi per spiegare cosa sta per succedere
  - commentare le variabili usate

Costanti nel codice

- *è PESSIMA norma* usare costanti numeriche o caratteri cablate nel codice

  Es:

```
exec 4<“ciccio”
echo ciccio aperto
```

  Meglio utilizzare invece variabili con nomi significativi ed usarle in modo consistente es

```
FILE=“ciccio”
exec 4<$FILE
echo $FILE aperto
```

Funzioni

- Bisogna sempre usare *local* dentro le funzioni per limitare lo scope delle variabili locali
  - Diminuisce la possibilita di errore per interferenze e rende il codice piu leggibile
- È bene anche utilizzare i parametri nelle funzioni
  - sempre per aumentare leggibilità ed evitare interferenze
- Dove possibile fattorizzare codice in funzioni per aumentare la leggibilità e diminuire la replicazione

Strutturazione

- Evitare il codice lungo, contorto e replicato
  - È possibile realizzare tutto lo script con 200—300 linee di codice max (stima larga)
  - Alcuni arrivano a 1500 linee!
  - in questi casi c’è qualcosa che non va

- Usare le funzioni per evitare la replicazione
  - Es. Evitare di replicare tutto il codice per ogni possibile combinazione dei parametri in ingresso allo script ma definire una funzione con opportuni parametri da chiamare nei vari casi

Accesso a variabili

- Se X è un nome che può contenere una stringa vuota, deve essere sempre racchiusto fra virgolette in una operazione di test

```
if [ -z “$X” ]; then ..
```

meglio di

```
if [ -z $X ]; then ..
```

inoltre

```
if [ $X = “” ]; then ..
```

da errore (ed esegue l’else!) se $X è vuota, invece

```
if [ “$X” = “” ]; then ..
```

Funziona, anche se per il test di stringa vuota è preferibile (e più chiaro) usare il **-z**

Accesso a variabili

- Se X è un nome che può contenere una stringa vuota, deve essere sempre racchiusto fra “...” in una operazione di test (contd.)

```
if [ $X = “abc” ]; then ..
```

da errore (ed esegue l’else in questo caso correttamente) se $X è vuota, invece

```
if [ “$X” = “abc” ]; then ..
```

È corretto e funziona anche con $X vuota ed esegue l’else correttamente senza segnalare l’errore

Accesso a variabili

- Se X è un nome che può contenere una stringa vuota, deve essere sempre racchiusto fra “...” in una operazione di test (contd.)

  È bene sottolineare che le virgolette eliminano la suddivisione in parole e preservano i blank se X contiene una spazio

  X=“ “

  I risultati di

  if [-z $X]; then ..

  e

  if [-z “$X”]; then ..

  sono ovviamente diversi

Miscellanea

- Bisogna sfruttare le potenzialità di comandi e builtin
  - In particolare i builtin sono molto più efficienti dei comandi esterni e sono da preferire se possibile
- Leggere sempre bene il man
  - e non assumere niente che non sia scritto li
- Evitare di leggere e scrivere file temporanei nella directory corrente
  - Usare **`mktemp`** per creare nomi univoci nella directory **`/tmp`**