

Sesta esercitazione

File comandi

Stefano Monti
stefano.monti6@unibo.it

Un esempio di ricorsione

- Scrivere un programma che, dato un direttorio in ingresso, stampi un elenco di file contenuti nel direttorio e in tutti i suoi sottodirettori (analogamente al comando ***ls -R***)

Schema di soluzione

recurseDirectory <arg1>

- Caso base: <arg1> **è un file** --> stampo il nome
- Caso generale espresso in termini ricorsivi: se <arg1> **è una directory** -->
 - mi muovo nella directory <arg1>
 - per ogni file (normale o directory) **invoco nuovamente recurseDirectory**

3

Bozza di soluzione

```
#!/bin/sh
```

```
if ! test -d $1  
then  
  echo `pwd`/$1
```

Caso base

```
else  
  cd $1  
  for f in *  
  do  
    $0 "$f"
```

Caso generale

```
  done  
fi  
exit 0
```

Ricorsione!

4

Struttura di un file comandi ricorsivo

invoker.sh



recursive.sh



5

Esercizio

Si scriva un file comandi con interfaccia

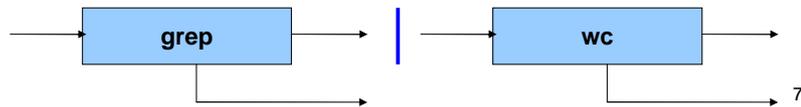
contaOccorrenze parola dir

- **dir** è un nome **assoluto** di direttorio
- **parola** è una qualunque sequenza di caratteri
- compito del file comandi è quello di stampare a video il nome del file con il maggior numero di linee contenenti **parola**, per ogni **sottodirettorio** di **dir**.

6

Suggerimenti...

- Per cercare occorrenze di stringhe all'interno di file → comando `grep`
 - a default, stampa in output le sole righe contenenti la stringa cercata
- Per contare righe, caratteri o parole di un file → comando `wc`
- Nel nostro caso, è molto comodo pensare che l'output di `grep` costituisca l'input di `wc` → `piping`



Soluzione – invoker.sh

```
#!/bin/sh
if test $# -ne 2
then
echo "usage:$0 <parola> <dir>"
exit 1
fi
case $2 in
/*) ;;
*) echo "$2 is not an absolute directory"
exit 3;;
esac
if ! test -d "$2"
then
echo "$2 is not a valid directory"
exit 4
fi
oldpath=$PATH
PATH=$PATH:`pwd`
recursive.sh "$1" "$2"
PATH=$oldpath
```

8

Esercizio - recursive.sh

```
#!/bin/sh
cd "$2"
counter=0 #valore massimo corrente
for f in *
do
  if test -h "$f" #se il file corrente e' un link
  then
    continue
  elif test -d "$f"
  then
    $0 "$1" "$f"
  elif test -f "$f"
  then
    cur=`grep "$1" "$f" | wc -l`
    if test $cur -gt $counter
    then
      counter=$cur
    fi
  fi
done
echo massimo numero di righe in $2: $counter
exit 0
```

Se volessi il file con massimo num di righe in tutti i sottodirettori, potrei continuare ad utilizzare la variabile \$counter?

9

Espansione

- Sequenza dei passi di sostituzione/ espansione
 - Comandi (` ` tra backquote)
 - Variabili e parametri (\$nome)
 - Metacaratteri nei nomi di file (* ? [])
- Inibizione dell'espansione
 - \ carattere successivo considerato come normale
 - ' ' proteggono da qualsiasi espansione
 - " " proteggono da espansioni eccetto \$\` `

10

Esercizio

Realizzare un file comandi (ricorsivo) che abbia la sintassi

```
search.sh <minSize> <maxSize> <dir1> <dir2>...<dirN>
```

Elenco (di lunghezza non nota a priori)
di direttori

dove

- <minSize> <maxSize > sono due interi
- <dir1> ... <dirN> sono un numero N qualsiasi, non noto a priori, di nomi di **direttori assoluti** che devono esistere nel file system.

11

Esercizio

Il compito del file comandi è quello di

- visitare (ricorsivamente) tutti i **sottoalberi** individuati da <dir1> ... <dirN>
- per ogni file trovato verificare che la **dimensione in KB** sia compresa tra <minSize> e <maxSize>
 - suggerimento: vedere il comando **stat** per ottenere la dimensione del file
- scrivere il nome assoluto di ciascun file che soddisfi il precedente requisito in un file di report da inserire nella home dell'utente che ha invocato il comando

12

Estensione

- Al termine dell'esecuzione, il file comandi deve stampare una stringa del tipo "nessun file soddisfa il requisito!" se non è stato trovato alcun file che soddisfi il requisito, oppure le righe del file di report ordinate alfabeticamente.