

# Sesta Esercitazione

File comandi  
Unix

---

# Passi di sostituzione: esempi

`echo `pwd` > "f1"`

# R: crea il file di nome **f1**, poi collega stdout di echo a **f1**

# 1: espande i backquote `` → `echo /usr/bin`

# 2: sostituisce variabili: nessuna operazione ulteriore di parsing

# 3: sostituisce metacaratteri: nessuna operazione di parsing

---

`test -f `pwd`/$2 -a -d "$HOME/docs"`

# R: redirectione IO: nessuna operazione di parsing

# 1: `test -f /temp/$2 -a -d "$HOME/docs"`

# 2: `test -f /temp/pluto -a -d "/home/utente/docs"`

# 3: sostituisce metacaratteri: nessuna operazione di parsing

# Esempio di file comandi

scrivere un file comandi che ogni 5 secondi controlli se sono stati *creati o eliminati file in una directory*.

- In caso di cambiamento, si deve visualizzare un messaggio su stdout (quanti file sono presenti nella directory)
- il file comandi deve poter essere invocato con *uno e un solo parametro*, la directory da porre sotto osservazione (fare opportuno controllo dei parametri)

**Suggerimento:** uso di un file temporaneo, in cui tenere traccia del numero di file presenti al controllo precedente

---

# Esempio di file comandi

```
#!/bin/bash
```

```
if [ $# -ne 1 ] ; then echo "Sintassi!" ; fi
```

```
echo 1 > loop.$$tmp
```

```
OK=0
```

```
while [ $OK = 0 ]
```

pid del processo in esecuzione

```
do
```

[ ] sono equivalenti a test

```
sleep 5s
```

```
new=`ls $1|wc -w`
```

```
old=`cat loop.$$tmp`
```

```
if [ $new -ne $old ]
```

```
then
```

```
echo $new > loop.$$tmp
```

```
echo in $1 ci sono `cat loop.$$tmp` file
```

```
fi
```

```
done
```

# Esercizio 1

- Realizzare un file comandi che preveda la seguente sintassi:

**maxcontainer D S**

dove **D** è una directory, e **S** una stringa.

- Il file comandi deve individuare nel direttorio **D** il file che contiene il maggior numero di occorrenze della stringa **S**.

# Suggerimenti

- Ciclo su un elenco di file:

```
for file in /path/to/file1.txt  
    /path/to/file2.txt /path/to/file3.txt; do  
    # do something on $file  
done
```

- Usando il percorso relativo alla directory corrente /path/to:

```
for file in file1.txt file2.txt file3.txt
```

- Usando anche il metacarattere ?

```
for file in to/file?.txt
```

E se voglio tutti i file nella directory D passata come parametro? quale **metacarattere** devo usare?