

# Unix - Shell dei comandi

## Sostituzioni

Prima dell'esecuzione di un comando, esso viene scandito (assieme ad i suoi parametri) e vengono operate delle sostituzioni

1. Sostituzioni delle variabili/parametri: ad ogni variabile viene sostituito il suo valore

Ad es.:

x=12

echo \$x

Produce 12

# Unix - Shell dei comandi

## Sostituzioni

Prima dell'esecuzione di un comando, esso viene scandito (assieme ad i suoi parametri) e vengono operate delle sostituzioni

2. Sostituzioni dei comandi: oogni comando compreso tra `` viene eseguito e sostituito col risultato

Ad es.:

```
echo `pwd`
```

Stampa il direttorio corrente

# Unix - Shell dei comandi

## Sostituzioni

Prima dell'esecuzione di un comando, esso viene scandito (assieme ad i suoi parametri) e vengono operate delle sostituzioni

3. Sostituzioni dei nomi dei file: i caratteri \*, ?, [ ] vengono sostituiti coi nomi dei file secondo il pattern matching sul file system

Ad es.:

ls \*

Elenca i file del direttorio corrente

# Unix - Shell dei comandi

## Sostituzioni

E' possibile decidere quali sostituzioni applicare e quando...

- ' <comando> ' : sospende tutte le sostituzioni
- " <comando> " : solo sostituzioni di tipo 1 e 2

# Unix - Shell dei comandi – Es. 1

- Creare un file di script (shell bash) che stampi a video la lista degli argomenti con cui è stato invocato, ed il numero di argomenti (senza considerare il nome stesso dello script).

# Unix - Shell dei comandi – Es. 1

## Soluzione

```
#!/bin/bash
for i in $*
do
    echo $i
done
echo numero argomenti: $#
```

## Unix - Shell dei comandi – Es. 2

- Creare un file di script (shell bash) che riceva come parametri una lista di nomi di file e, qualora se ne possieda il diritto di lettura, ne stampi a video il contenuto; si stampi a video un messaggio di errore altrimenti

# Unix - Shell dei comandi – Es. 2

## Soluzione

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

```
for i in $*
```

```
do
```

```
    if test -r $i
```

```
    then
```

```
        cat $i
```

```
    else
```

```
        echo Non ho i permessi di lettura per $i
```

```
    fi
```

```
done
```

## Unix - Shell dei comandi – Es. 3

- Creare un file di script (shell bash) che chieda all'utente di inserire una stringa, e produca in output tutti i file presenti nel direttorio corrente il cui nome contenga quella stringa.
- Al termine, il programma deve chiedere all'utente se vuole inserire una nuova stringa o terminare

# Unix - Shell dei comandi – Es. 3

## Soluzione

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

```
answer=yes
```

```
while test $answer = yes
```

```
do
```

```
    echo Insert the substring:
```

```
    read substring
```

```
    echo `ls *$substring*`
```

```
    echo Do you want to continue[yes]?
```

```
    read answer
```

```
done
```

## Unix - Shell dei comandi – Es. 4

- Creare un file di script (shell bash) che, ricevuto come argomento il nome di un file, provi ad accederne il contenuto.
- Supponendo che tale file contenga, per ogni riga, il nome di un ulteriore file, si scriva in unico file di nome temp.txt il contenuto di tutti i file elencati (in append).
- Si abbia cura di controllare che lo script sia invocato con almeno un parametro, altrimenti si stampi a video un messaggio di errore.

# Unix - Shell dei comandi – Es. 4

## Soluzione

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

```
case $# in
```

```
0 ) echo usage $0 nomefile;;
```

```
* ) for i in `cat $1`
```

```
do
```

```
    if test -f $i
```

```
    then
```

```
        cat $i >> temp.txt
```

```
    fi
```

```
done;;
```

```
esac
```