## Script della shell

**Esecuzione di uno script**

sh scriptfile

oppure assegnando attributo di esecuzione

chmod u+x scriptfile

./scriptfile

**Commenti**

# commenti

## Parametri passati allo script

$n $1 1° parametro

$# numero parametri

$\* elenco tutti i parametri

$? Codice uscita dell’ultimo comando

$- ultimo parametro

## Variabili

**Assegnazione**

nome=valore

**Espansione delle variabili**

echo $nome $ consente di sostituire al nome il valore

**Espansione di espressioni aritmetiche**

$((espressione)) oppure $[espressione]

Esempio: $((a+b)), $[a+b], $((10+4))

**Input di variabili da uno script**

**read** VAR1 VAR2…

## Condizioni

La condizione si esprime tra parentesi quadre (attenzione spazio dopo apertura, tra gli operatori e prima chiusura)

Esempio : [ $# != 2 ]

Operatori con numeri interi

-eq,-ne,-gt,-ge,-lt,-le

Operatori con stringhe

$var vero se la stringa è stata definita

= esempio $var1 = $var2

!= esempio $var1 != $var2

-z vero se stringa ha lunghezza uguale a zero

-n vero se stringa h

a lunghezza diversa da zero

Operatori logici

! not

-a and

-o or

Se combinati devono essere distanziati da uno spazio

Operatori su file

-d vero se il file esiste ed è una directory

-e vero se il file esiste

-f vero se il file esiste ed un file normale

-s vero se file esiste e non è vuoto

-r vero se autorizzazione in lettura

-w vero se autorizzazione in scrittura

-x vero se autorizzazione in esecuzione

-O vero se proprietario

-G vero se appartiene al proprio gruppo

file1 –nt file2 vero se file1 è più recente di file2

file1 –ot file2 vero se file1 è meno recente di file2

## Istruzioni di controllo

**if** condizione

then

istruzioni

else

istruzioni

fi

**for** variabile **in** elenco

do

istruzioni

done

**while** condizione

do

istruzione

done

**case** selettore **in**

schema1) le

comandi;;

schema2)

comandi;;

….

\*)

comandi;;

esac

**select** per creare dei menu di scelta nao

Esercizi

1. Creare uno script che accetti una serie di numeri terminante con 0 e restituisca il numero di valori pari e il numero di valori dispari inseriti.
2. Creare uno script che dato un numero da 1 a 12, restituisca il nome del mese corrispondente.
3. Realizzare uno script che riceva come parametri due operandi e un operatore ed esegua l'operazione.
4. Scrivere uno script “dir.sh” che accetti come solo parametro il path e il nome di una directory e ne che elenchi il contenuto;
5. Scrivere uno script che rinomina i file Pippo1,Pippo2,..,Pippo10 in PlutoN.dat ma solo per quelli con N dispari;
6. Scrivere uno script che elenchi i file della directory corrente che contengono la parola “grep”.

Esempio

Soluzione 3)

#!/bin/bash

# Lo script riceve come parametri due operandi e un segno di operazione ( NUM1 OP NUM2)

# ed esegue l'operazione richiesta.

RIS=0

bool=TRUE

if [ $# != 3 ]

then

echo " Devono essere inseriti 3 parametri ( NUM1 OP NUM2)"

else

case $2 in

+) RIS=$(($1 + $3)) ;;

-) RIS=$(($1 - $3)) ;;

x) RIS=$(($1 \* $3)) ;;

/) RIS=$(($1 / $3)) ;;

^) RIS=$(($1 \*\* $3)) ;;

%) RIS=$(($1 % $3)) ;;

\*) echo "Operatore non valido!"

$bool=FALSE ;;

esac

if [ $bool=FALSE ]

then

echo "Risultato = $RIS"

fi

fi

exit 0