Ambienti di Programmazione per il Software di Base

Shell Script

Shell script

```
if [ $a –lt $b ]; then
        echo ”a < b"
else
        echo ”a >=b"
fi
```

-lt = less then

-gt = greater then

-le = less or equal then

-ge = greater or equal then

=, != uguale, diverso

```
if [ -d $file1]; then
    echo ”$file1 e’ una cartella"
fi
```

-f = è un file

-h = è un link simbolico

-d = è una directory

http://www.pluto.it/files/ildp/guide/abs/index.html

Shell script

read linea memorizza nella variabile linea l'input dell'utente

Es.

```
echo –n "inserisci un numero: "
read numero
echo "hai inserito $numero"
```

for *arg* in [*lista*]
do
 comando(i)...
done

```
for f in $(ls); do
    echo "$f"
done
```

a=$(comando) restituisce l'output di comando nella variabile a

Shell script

Argomenti di uno script bash

$# numero di argomenti

$0 nome dello script

$1, $2, …… n-esimo argomento

$* Tutti i parametri posizionali visti come un'unica parola

$@ Simile a $*, ma ogni parametro è una stringa tra apici (quoting),Questo significa, tra l'altro, che ogni parametro dell'elenco viene considerato come una singola parola.

Es. ‘arg1’ ‘arg 2’ ‘arg3’

Shell script

clean_my_files.sh

```
# elimino tutte le occorenze di MYPATTERN dai file di testo
LOGFILE=/tmp/myprogram.log
OUTDIR="nuovi_files”
MYPATTERN='errore’

if [ $# -eq 0 ]; then
        echo "Usage: $0 file1 ..."
        exit 1
Fi

mkdir -p $OUTDIR
                        …..
```

……

```
for i in $@; do
#check
if [ ! -f "$i" ]; then
        echo "Only files are accepted ($i)"
         continue
fi
# convert
echo -n "convert ${i}: "
cat "$i" | grep -v $MYPATTERN > "$OUTDIR/$i"  2>> $LOGFILE
 # result
if [ $? -eq 0 ]; then
msg="File converted successfully"
else
msg="An error occurred (view $LOGFILE )"
fi
echo "$msg"done
```

Shell script

Esercizio 1

Scrivere un programma shell che scriva in un file gli utenti attivi sul pc (e i loro programmi).
Il programma è invocato come segue:

MemUtenti <nome_file>

e memorizza nel file nome_file il nome di tutti gli utenti e per ogni utente i programmi che stanno usando.

SUGGERIMENTO: utilizzare who -q per conoscere gli utenti (e poi head.. per prendere solo gli utenti) e ps -uNome_Utente per vedere che programmi l'utente sta utilizzando.
Per eliminare i duplicati con un for si mettono gli utenti in un file utenti.txt e si può usare cat utenti.txt |sort | uniq (che ordina prima un file e poi elimina i duplicati)

Shell script

Esercizio 2

Scrivere un programma shell che cancelli i file inutili da una serie di cartelle. Il programma è invocato come segue:

Pulisci <cartella1> <cartella2>.... <cartellan>

Per ogni cartella, il programma chiederà, leggendo da tastiera, quali tipi di file eliminare (esempio: .txt , .java, ecc.). L'utente scriverà con la tastiera le estensioni da eliminare e indicherà la terminazione (nessun altra estensione) scrivendo END.

Il programma per ogni serie di estensione eliminata dovrà scrivere: Ho eliminato 10 file .txt oppure Non esiste nessun file con estensione .txt .

Sarà gradito il controllo degli errori (cartella non esistente, ecc..)

Shell script

Esercizio 3

Scrivere un programma shell che riceva da linea di comando 5 argomenti di tipo intero positivo.
Il programma leggerà il nome di una cartella da tastiera (con l'istruzione read), e quindi inserirà la media dei 5 numeri in tutti i file con estensione .txt contenuti nella cartella. Il programma deve gestire i casi d eccezione (numero di argomenti diverso da 5, interi non positivi, cartella non esistente) interrompendo l'esecuzione con un messaggio all'utente.

Ad esempio, se il programma si chiama esercizio e l'utente batte da tastiera cartmedia, l invocazione di:

esercizio 5 10 10 20 5

scriverà 10 in tutti i file con estensione .txt contenuti nella cartella cartmedia.

Shell script

Esercizio 4

Itunes salva la musica in cartelle cantanti e in cartelle album per ogni cantante in formato m4a che pochi lettori riconoscono.

Realizzare uno script (trasformMusic) che trasforma tutti i file in formato m4a in mp3 e uno script removeMusic che cancella tutti i file m4a.

Se non è installato, con ubuntu basta eseguire: sudo apt-get install ffmpeg
Quindi per convertire: ffmpeg -i "canzone.m4a" -b 320k "canzone.mp3"

Lo script funziona con quasi tutti i formati, basta sostituire nello script la dicitura *.m4a con il vostro formato (per esempio *.ogg)