

#! /bin/bash

ESERCIZI

```
~root: env X="(cat /dev/shm/shellshock) && echo completed"
```

```
> shellshock
```

```
> completed
```

- Creare uno script che aggiorni data e ora ogni secondo:

```
#!/bin/bash
while [ 1 = 1 ]
do
    date;
    sleep 1
done
```

- Contare il numero di file di log di un sistema:

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

```
ls -l /var/log/*.log | wc -l
```

- Cercare le attività di un utente nei log:

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

```
# $1 contiene lo username.
```

```
ls -l /var/log/*.log | grep $1
```

- Creare uno script estragga nome utente e geccos dal file `/etc/passwd`:

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

```
getent passwd | cut -d: -f1,5
```

- Estendere lo script precedente per creare delle directory per tali utenti:

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

```
for i in $(getent passwd | cut -d: -f1); do
```

```
    mkdir /scratch/$i
```

```
    chown -R $i /scratch/$i
```

```
done;
```

- Realizzare uno script che stampa il contenuto di una directory con il peso di ogni file:

```
#!/bin/bash  
for i in $(ls -l); do  
    du -hs $i;  
done;
```

Esempio di until..do

```
x=100
```

```
until [ "$x" -ge 10 ]; do
```

```
    echo "Valore corrente di x: $x";
```

```
    x=$(( $x - 1 ))
```

```
done
```

- Realizzare uno script che stampa solo i file pdf presenti in una directory:

```
#!/bin/bash  
for i in $(ls -l *.pdf); do  
    lpr $i;  
done;
```

- Realizzare uno script che faccia un backup della vostra home:

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

```
tar cvf $USER.tar $HOME
```

- Realizzare uno script che faccia un backup della vostra home con data e ora:

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

```
tar cvf $USER-$(date '+%d-%m-%Y_%H:%m') .tar $HOME
```


- Supponendo di essere root di un sistema in cui tutti gli utenti hanno home in `/home`, creare uno script che effettua il backup della home di ogni singolo utente in `/backup/nomeutente`:

```
#!/bin/bash

mkdir -p /backup

for i in $(ls -1 /home); do
    mkdir -p /backup/$i;

    tar cvf /backup/$i/$i-$(date '+%d-%m-%Y').tar
    /home/$i;

done;
```

- Si supponga di avere $N < 100$ account didattici, nominati come `corso01...corsoN`, realizzare uno script (`myscript.sh`) che copi un file nella home di tali utenti:

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

```
#$1 contiene il file sorgente, $2 il  
#file destinazione.
```

```
for i in $(seq 1 9); do cp -av $1  
/home/corso0$i/$2; done
```

```
for i in $(seq 10 N); do echo cp -av  
$1 /home/corso$i/$2; done
```

- Aggiungere allo script precedente la possibilità di cancellare o copiare il file:

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

```
# $1 contiene il comando da eseguire, $2 il file da copiare/cancellare.  
Non è previsto un nome destinazione diverso da $2
```

```
if [ "$1" = "copia" ] ; then
```

```
    for i in $(seq 1 9); do cp -av $2 /home/corso0$i/$2; done
```

```
    for i in $(seq 10 N); do cp -av $2 /home/corso$i/$2; done
```

```
fi
```

```
if [ "$1" = "cancella" ] ; then
```

```
    for i in $(seq 1 9); do rm -ir /home/corso0$i/$2; done
```

```
    for i in $(seq 10 99); do rm -ir /home/corso$i/$2; done
```

```
fi
```

- Sintassi:

```
# myscript.sh copia pippo
```

```
# myscript cancella pippo
```

- Aggiungere allo script precedente dei controlli sui parametri passati:

```
#!/bin/bash

if [ $# -lt 2 ] ; then

    echo "Numero di parametri errato!" >&2

    echo "Sintassi: $0 <operazione> <file>"

    echo "<operazione> = copia|cancella"

    exit 1;

fi

if [ -z "$1" ] ; then

    echo "Parametri vuoti" >&2

    echo "Sintassi: $0 <operazione> <file>"

    echo "<operazione> = copia|cancella"

    exit 1;

fi

if [ -z "$2" ] ; then

    echo "Parametri vuoti" >&2

    echo "Sintassi: $0 <operazione> <file>"

    echo "<operazione> = copia|cancella"

    exit 1;

fi

...segue il resto dello script ...
```

- Realizzare uno script che, a partire da un file con un elenco di utenti elimini le home di tali utenti residenti in `/home` dopo un backup:

```
#!/bin/bash
mkdir -p /backup
for i in $(cat utentidaeliminare.txt); do
    echo Cancellato $i
    tar zcf /backup/$i.tgz /home/$i
    rm -rf /home/$i
    Echo Home $i eliminata
done;
```

Realizzare uno script che prende in input una lettera ed esegue il comando corrispondente `date`, `who`, `ls -l | more`, `pwd`:

```
#!/bin/bash

# scelte di comandi

case $1 in
    'D') date;;
    'W') who;;
    'L') ls -l | more;;
    'P') pwd;;
    *)

        echo 'D = data odierna'
        echo 'W = utenti collegati'
        echo 'L = lista dei file'
        echo 'P = directory corrente';;

esac
```

- Ricavare l'ipv4 di un host:

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

```
/sbin/ifconfig | grep 'inet' | grep -v '127.0.0.1' |  
grep -v 'inet6' | cut -d: -f2 | cut -d' ' -f1
```

```
/sbin/ifconfig | grep 'inet' | grep -v '127.0.0.1' |  
grep -v 'inet6' | cut -d' ' -f10
```

Ricavare FQDN di un host:

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

```
$IP=$(/sbin/ifconfig | grep 'inet' | grep -v  
'127.0.0.1' | grep -v 'inet6' | cut -d: -f2 | cut -d'  
' -f1)
```

```
$IP=$(/sbin/ifconfig | grep 'inet' | grep -v  
'127.0.0.1' | grep -v 'inet6' | cut -d' ' -f10)
```

```
FQDN=$(host $IP | cut -d' ' -f5 | awk -F"|" '{print  
$1}')
```

```
echo $FQDN
```