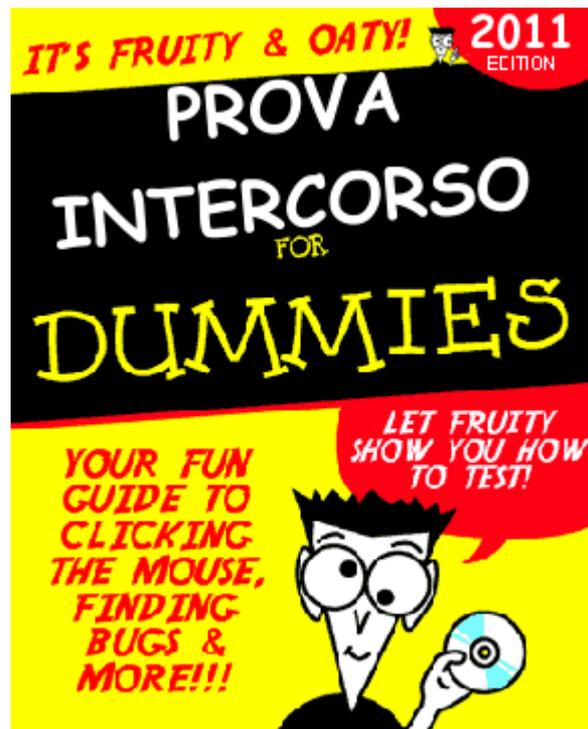


Laboratorio di sistemi operativi  
A.A. 2010/2011  
Gruppo 2  
Gennaro Oliva  
Soluzioni della prova intercorso



# Prova intercorso a1-2

- *a1)* mkdir uno && chmod a-r uno && mkdir uno/uno && \ touch uno/uno/uno && cat uno/uno/uno && echo uno  
uno
- *a2)* mkdir due && chmod a-x due && mkdir due/due || \ echo due.1 && touch due/due/due && echo due.2  
due.1

# Prova intercorso a3-5

- Supponendo che nella directory corrente ci siano i file:  
INSTALL Makefile Makefile.am Makefile.in README  
aux.c core input.c main.c main.o functions.h
- *a3)* `ls|grep '.c'`  
`aux.c functions.h input.c main.c`
- *a4)* `ls|grep '[ux]'`  
`aux.c input.c functions.h`
- *a5)* `ls|grep '[:upper:]*'|wc -l`

# Prova intercorso a6-8

- Supponendo che nella directory corrente ci siano i file:  
INSTALL Makefile Makefile.am Makefile.in README  
aux.c core input.c main.c main.o functions.h
- *a6)* `ls|grep '^[[[:upper:]]'|wc -l`  
5
- *a7)* `ls|sed 's/put/get/'|grep get`  
inget.c
- *a8)* `ls|sed 's/a\(.*\n/a\1\1n/'`
- aux.c core functions.h input.c INSTALL main.c main.o  
Makefile Makefile.am Makefile.in README

# Prova intercorso a9-12

- *a9)* rimuovere tutti i file che hanno estensione di 3 caratteri di cui l'ultimo carattere è una cifra (es. song.mp3, archivio.tar.bz2, ...)

```
rm *.*[0-9]
```

- *a10)* rimuovere tutti i file il cui nome non inizia con un cifra

```
rm [^0-9]*
```

- *a11)* contare le righe di un file che iniziano per una vocale

```
grep '^[aeiou]' file | wc -l
```

- *a12)* visualizzare le righe di un file che iniziano con due o più cifre consecutive o con una vocale ed una cifra utilizzando grep ed una singola espressione regolare

```
grep '^([0-9]{2}|[aeiou][0-9])' file
```

```
grep '^([0-9aeiou][0-9])' file
```

# Prova intercorso a13-14

- *a13)* scrivere nel file mytmpfiles nella home directory dell'utente l'elenco di file regolari di proprietà dell'utente contenuti nella directory /tmp, il comando deve funzionare a prescindere dal nome utente e dalla directory corrente

```
ls -l /tmp | awk -v user=$USER '$1 ~ /^-/\ \
    && $3 == user {print $9}' > ~/mytmpfiles
```

- *a14)* stampare le matricole degli studenti file matricole nei formati:

```
N86113 N86000100 N86\113 N8633 N86000034
N86\132 566127 566000033 566/44 566\255
```

- nel formato modificato:

```
N86/113 ... 566/255
```

```
sed "s/\<\(...\)^[1-9]*\([0-9]*\) \>/\1\/\2/g"
```

# Prova intercorso a15

- *a15*) Si scriva un comando che conti le occorrenze del basename di un file (la parte prima del punto o il nome del file se non c'è il punto). Esempio se nella directory ci sono i seguenti file:

hello main.c main.i main.o main.s Makefile  
Makefile.am Makefile.in

- il comando produrrà un output del tipo

1 hello

4 main

3 Makefile

- `ls | sed "s/\\.*/"/" | sort | uniq -c`

# Prova intercorso b1

- Si realizzi uno script che accetta come argomento sulla linea di comando due parametri: una parola chiave ed il nome di un file. A seconda della parola chiave passata lo script visualizza l'elenco di file memorizzati nella directory corrente con le caratteristiche specificate:
- newer file con data di ultima modifica successiva a quella del file passato come argomento
- older file con data di ultima modifica antecedente a quella del file passato come argomento
- larger file con dimensione maggiore di quella del file specificato come argomento
- smaller file con dimensione maggiore di quella del file specificato come argomento
- Lo script dovrà verificare gli argomenti passati a linea di comando (numero di argomenti, esistenza del file, validità parola chiave). Si supponga che nella directory corrente ci sono solo file regolari.

# Prova intercorso b1

```
#!/bin/bash
[ $# == 2 ] && [ -e $2 ] || exit 1
case $1 in
  newer)
    for file in * ; do
      if [ $file -nt $2 ] ; then echo $file ; fi
    done ;;
  older)
    for file in * ; do
      if [ $file -ot $2 ] ; then echo $file ; fi
    done ;;
  larger)
    LIMIT=$(ls -l $2 | awk '{print $5}')
    for file in * ; do
      FILESIZE=$(ls -l $file | awk '{print $5}')
      [ "$FILESIZE" -gt "$LIMIT" ] && echo $file
    done ;;
  smaller)
    LIMIT=$(ls -l $2 | awk '{print $5}')
    for file in * ; do
      FILESIZE=$(ls -l $file | awk '{print $5}')
      [ "$FILESIZE" -lt "$LIMIT" ] && echo $file
    done ;;
  *)
    exit 1 ;;
esac
```

# Prova intercorso b2

- Si realizzi uno script che stampi a video la somma degli interi positivi passati come argomento sulla linea di comando fino ad arrivare ad una massima di 100 ed in caso venga superato questo limite stampi la frase.

- La somma e' maggiore di 100

- `#!/bin/bash`

```
TOTALE=0
```

```
while [ "$TOTALE" -le 100 -a "$#" != 0 ] ; do
```

```
    TOTALE=$((TOTALE+$1))
```

```
    shift
```

```
done
```

```
if [ "$TOTALE" -le 100 ] ; then
```

```
    echo $TOTALE
```

```
else
```

```
    echo La somma e\' maggiore di 100
```

```
fi
```

# Prova intercorso c

```
#!/usr/bin/awk -f
/^#/ {next}
$1 !~ /^192.168.1/ {traffic=traffic+$4; nome[$1]=$1; host[$1]+=$4}
{count[$5]+=1}
END {
    print "traffico totale="traffic/2^30
    print "traffico verso ip esterni"
    for (ip in host)
        print ip,host[ip]/2^30
    for (file in count)
        if (count[file]>max) {
            max=count[file];
            mostdownloaded=file
        }
    print mostdownloaded, "(" count[mostdownloaded] " volte)"
}
```