MasterIT - Scripting Es1a

Sh, Csh, Grep, Sed, Awk

Sandro.Angius@lnf.infn.it

30/10/2002

ese4.sh

- banner “Bye Bye” (su axcalc o dxcalc)

```
######                          ######
#     #   #   #  ######        #     #   #   #  ######
#     #    # #   #             #     #    # #   #
######      #    #####         ######      #    #####
#     #     #    #             #     #     #    #
#     #     #    #             #     #     #    #
######      #    ######        ######      #    ######
```

- ese4.sh “Bye Bye”

```
BBBBBB                          BBBBBB
B     B   Y   Y  EEEEEE        B     B   Y   Y  EEEEEE
B     B    Y Y   E             B     B    Y Y   E
BBBBBB      Y    EEEEE         BBBBBB      Y    EEEEE
B     B     Y    E             B     B     Y    E
B     B     Y    E             B     B     Y    E
BBBBBB      Y    EEEEEE        BBBBBB      Y    EEEEEE
```

- Su linux banner ~ /usr/libexec/filters/lpbanner -L"Bye Bye" | grep -E 'X|^ *$'
- Utilizzare awk; Soluzioni alternative?

ese4.sh

```
#!/bin/sh
banner "$*" | awk -v cfont="$*" '{
        linea=$0;
        slen=length(linea);
        nline="";
        for (cnt=1; cnt <= slen; cnt++) {
                cptr=1 + cnt / 8;
                if (substr(linea, cnt, 1) != " "){
                        nline=nline toupper(substr(cfont, cptr, 1));
                }
                else {
                        nline=nline " ";
                }
        }
        print nline;
}'
```

- Soluzioni alternative?

ese5.sh, ese5.csh, ese5.awk

- Dato un numero IP valido e il numero di bit della netmask, riportare l'identificativo di network, esempio:
  - **./ese5.sh 193.206.84.219 21**
    193.206.84.219/21 ===> 193.206.80.0
  - **./ese5.csh 192.168.160.14 24**
    192.168.160.14/24 ===> 192.168.160.0
  - **./ese5.awk 10.199.213.87 12**
    10.199.213.87/12 ===> 10.192.0.0
- Codifica con sh, csh, awk

ese5.sh

```
#!/bin/sh

digip=0

if test $# -ne 2
then
    echo "Usage: $0 <Numero IP> <Numero bit maschera IP>"
    exit -1
fi

chk=`expr $1 : ^\[0-9\]\*\.\[0-9\]\*\.\[0-9\]\*\.\[0-9\]\*\$`

if test $chk -lt 7
then
    echo "Numero IP non valido, deve essere A.B.C.D"
    exit -2
fi

if [ $2 -lt 1 ] || [ $2 -gt 32 ]
then
    echo "Il numero di bit maschera deve essere tra 1 e 32"
    exit -3
fi

for byte in `echo $1 | tr '.' ' '`
do
        if test $byte -gt 255
        then
            echo "Numero ip non valido ($byte > 255)"
            exit -4
        fi
        digip=`expr 256 \* $digip + $byte`
done
```

```
cnt=$2
pow=1

while test  $cnt -lt 32
do
        pow=`expr 2 \* $pow`
        cnt=`expr $cnt + 1`
done

digim=`expr $pow \* \( $digip / $pow \)`

cnt=0
ipmsk=""
spc=""

while [ $cnt -lt 4 ]
do
    byte=`expr $digim % 256`
    digim=`expr $digim / 256`
    cnt=`expr $cnt + 1`
    ipmsk="${byte}${spc}${ipmsk}"
    spc='.'
done

echo "$1/$2 ===> $ipmsk"
```

ese5.csh

```
#!/bin/csh

@ digip=0

if ( $#argv != 2 ) then
    echo "Usage: $0 <Numero IP> <Numero bit maschera IP>"
    exit -1
endif


if ( $1 !~ [0-9]*\.[0-9]*\.[0-9]*\.[0-9]* ) then
    echo "Numero IP non valido, deve essere A.B.C.D"
    exit -2
endif

if ( $2 < 1 ||  $2 > 32 ) then
    echo "Il numero di bit maschera deve essere tra 1 e 32"
    exit -3
endif

foreach byte ( `echo $1 | tr '.' ' '` )
        if ( $byte > 255 ) then
            echo "Numero ip non valido ($byte > 255)"
            exit -4
        endif
        @ digip = ( $digip << 8 ) + $byte
end
```

```
@ pow = (1 << (32 - $2))
@ digim = $pow * ( $digip / $pow )

@ cnt = 0
set ipmsk=""
set spc=""

while ( $cnt < 4 )
    @ byte  = ( $digim & 255 )
    @ digim = ( $digim >> 8 )
    @ cnt++
    set ipmsk="${byte}${spc}${ipmsk}"
    set spc='.'
end

echo "$1/$2 ===> $ipmsk"
```

ese5.awk

```
#!/usr/bin/awk -f
BEGIN{
    if (ARGC != 3) {
        print "Usage: ese5.awk <Numero IP> <Numero bit maschera IP>";
        exit -1;
    }
    ip = ARGV[1];
    if ( ip !~ "^([0-9]{1,3}\.){1,3}[0-9]{1,3}$" ) {
        print "Numero IP non valido, deve essere A.B.C.D";
        exit -2;
    }
    bmsk = ARGV[2];
    if ( (bmsk < 1) || (bmsk > 32)) {
        print "Il numero di bit maschera deve essere tra 1 e 32";
        exit -3;
    }
    split( ip, ipbytes, ".");
    digip = 0;
    for (cnt = 1; cnt < 5;) {
        if ( ipbytes[cnt] > 255 ) {
            printf "Numero IP non valido (%s > 255)\n", ipbytes[cnt];
            exit -4;
        }
        digip = (256 * digip) + ipbytes[cnt++];
    }
    pow   = 2 ^ (32 - bmsk);
    digim = pow * int( digip / pow);
    ipmsk = "";
    spc   = "";

    for ( cnt = 4; cnt--;) {
        byte   = digim % 256;
        digim /= 256;
        ipmsk  = byte spc ipmsk;
        spc    = ".";
    }

    printf "%s/%s ===> %s\n",
    ip, bmsk, ipmsk;
}
```

Ulteriori esempi, uso di “eval”

- Costruzione di record per file userdb usato da sendmail per tradurre le username in mailaddress e viceversa, esempio:

- genuserdbrec angius Sandro.Angius nickname

angius:maildrop	angius@uxcalc.lnf.infn.it
angius:mailname	Sandro.Angius
Sandro.Angius:maildrop	angius@uxcalc.lnf.infn.it
nickname:maildrop	angius@uxcalc.lnf.infn.it

```
#!/bin/sh

# Just a little script to generate line-records for userdb
# Sandro Angius (LNF/CS) 21 Feb 1997

if [ $# -lt 2 ] ; then
        echo ‘Usage: genuserdb <user> <mailbox> [mailname] [maildrop ...]’
        exit
fi

_format=“awk ‘{printf \"%-40s %s\\n\", \$1, \$2}’”
_user=$1
_mbox=$2
shift
shift

echo "${_user}:maildrop ${_user}@${_mbox}.lnf.infn.it" | eval ${_format}
if [ $# -gt 0 ] ; then
        _mname=$1
        echo "${_user}:mailname ${_mname}" | eval ${_format}
        echo "${_mname}:maildrop ${_user}@${_mbox}.lnf.infn.it" | eval ${_format}
        shift
fi
while [ $# -gt 0 ] ; do
        echo "${1}:maildrop ${_user}@${_mbox}.lnf.infn.it" | eval ${_format}
        shift
done
```

Ulteriori Esempi: controllo quota

– La procedura che segue e’ un prototipo che consente di inviare un mail di avviso agli utenti che superano il 90% di quota occupata.

  La procedura verra’ eseguita piu’ volte al giorno ma dovra’ inviare al massimo un avviso ogni 10 giorni.

```
#!/bin/sh

# Check User Inbox Quotas          # Author: Sandro Angius - LNF/INFN
# Last update: 15/Jul/2001         # (Test Version)

BDIR=$HOME/tmp/log/checkquota
TOUCH=/usr/bin/touch
DOMAIN='dummy.it'
INBOXES=/var/spool/mail
MAILX="/bin/echo MAIL "
FIND=/usr/bin/find
AWK=/usr/bin/awk
TEXT="…. Testo del mail …"

$FIND $BDIR -type f -ctime +10 -exec rm {} \;

##For test we cant use repquota...
##repquota $INBOXES |\
##Following two lines are an accettable test
awk 'BEGIN{srand(); cnt=1; while (cnt++<1000) print "uid" cnt, " -- ", int(rand()*40000), "40000 90000 1 1 1"}' |\
$AWK '{if (1*$4) {if ( ($3/$4) > 0.9) {print $1}}}' | grep -v -E `(echo root; ls $BDIR) | xargs echo | tr ' ' '|'` |\
while read uname
do
   echo $TEXT |\
   $MAILX -s "ATTENZIONE/WARNING: Spazio x '$uname' in esaurimento!" "${uname}@$DOMAIN"
   $TOUCH $BDIR/$uname
done
exit
```

Ulteriori Esempi

– La procedura che segue e' un prototipo per il controllo della temperatura all'interno di un sistema.

  La temperatura misurata viene conservata in un file e se vengono superate determinate soglie viene inviato un (e uno solo) mail di allarme.

```
#!/bin/sh

BDIR=/tmp/log;                    LOGFILE=${BDIR}/cstemplog;    TOUCH=/usr/bin/touch;
          EMAIL='support@dummy.it';    MAILX="/bin/echo";              DATE=`date
+%Y%m%d%H%M`
T1FILE=${BDIR}/ABOVET1;      TRIGT1=25
T2FILE=${BDIR}/ABOVET2;      TRIGT2=30

# TEMP=`/usr/sbin/ssaencl -l enclosure0 -a | grep ambient_temperature | cut -f2 -d' '`
# Per test inseriamo la temperatura da terminale
echo "Inserire temperatura\r"; read TEMP

TEXT=" \n\nLa temperaura misurata e' di $TEMP gradi centigradi.\n\n"

if [ $TEMP -gt $TRIGT1 ]
then
  if [ ! -f $T1FILE ]
  then
    $TOUCH $T1FILE
    echo $TEXT | sed "s/SOGLIA/$TRIGT1/" | $MAILX -s "ALERT: Soglia 1 superata" $EMAIL
  fi
  else
    rm -f $T1FILE
fi

# […omissis …]

echo "$DATE $TEMP" >> $LOGFILE
```

Bibliografia

- man sh
- man csh
- man regexp
- man grep
- man sed
- man awk
- Copia degli scripts si trova in: /afs/lnf.infn.it/project/master.it/doc/Scripting/EsScripts/