

Stage 4: creating an expect script to telnet

- run the `telnet` command to login to a remote machine
- note the output carefully:

```
Trying 192.168.0.1...
Connected to merlin.
Escape character is '^]'.
Debian GNU/Linux 2.2 merlin
merlin login: foobar
Password:
Last login: Mon Mar  4 11:50:37 2002
```

Expect script considerations

- consider what happens if the wrong username or password is issued:

```
telnet merlin
Trying 192.168.0.1...
Connected to merlin.
Escape character is '^]'.
Debian GNU/Linux 2.2 merlin
merlin login: foobar
Password:
Login incorrect
merlin login:
Login timed out after 60 seconds.
Connection closed by foreign host.
```

- how should our expect script catch this case?

Tutorial/Lab/Assignment help

- write an expect script which takes 3 or 4 arguments
 - first argument is the machinename
 - second argument is the username
 - third argument is the password
 - optional fourth argument is `-a` if the telnet session is to be closed
 - in this case consider making your expect script exit with 0 if login was successful and exit with a 1 if it failed

Tutorial/Lab/Assignment help

- modify your `autotel` script to obey the `-h` option (as specified in the assignment)
- now modify your `autotel` script to call your expect script with the appropriate parameters
- now modify the `autotel` script to obey the `-a` option
 - in this case it should check the exit code from the expect script

autoftp revisited (part 1)

```
#!/bin/sh

scriptname=$0 # $0 is the name of the program
verbose=no
probeonly=no
TMP=/tmp/auto.$$

function usage () {
  cat <<EOF
Usage: $scriptname [-p] [-v] [-h] filename
  -p  probes the ftp sites and check urls in
      filename
  -v  executes and prints out verbose messages
  -h  displays basic help
EOF
  exit 0
}
```

autoftp revisited (part 2)

```
#
# findSiteDirFile - given a line $1 in the form
#                  site/dir/file extract the
#                  site dir and file components

function findSiteDirFile () {
  j=1
  SITE=`echo $1 | cut -d '/' -f 1`
  rest=`echo $1 | cut -d '/' -f 2-`
  if [ "`echo $rest | \
    cut -f 1 -d '/'`" = "$rest" ]
  then
    FILE=$rest
    DIR="/"
  else
    while [ "`echo $rest | \
      cut -f $j -d '/'`" != "" ]
    do
      j=`expr $j + 1`
    done
    j=`expr $j - 1`
    FILE=`echo $rest | cut -f $j -d '/'`
    j=`expr $j - 1`
    DIR=`echo $rest | cut -f -$j -d '/'`
  fi
}
```

autoftp revisited (part 3)

```
function scanURLs () {
  cut -d '#' -f1 $1 > $TMP
  n=`wc -l $TMP | cut -b 0-7`
  i=1
  while [ $i -le $n ]
  do
    line=`head -$i $TMP | tail -1`
    if [ "$line" != "" ]
    then
      findSiteDirFile $line
      echo "site is" $SITE "file is" $FILE \
        "dir is" $DIR
      exp7.exp $SITE $DIR $FILE
    fi
    i=`expr $i + 1`
  done
}
```

autoftp revisited (part 4)

```
while getopts ":pvh" opt; do
  case $opt in
    v ) verbose=yes ;;
    p ) probeonly=yes ;;
    h ) usage ;;
    \?) usage ;;
    esac
  done
  shift $((OPTIND - 1))

  PATH=$PATH:.
  scanURLs $1

  /bin/rm -f $TMP
```

autoftp in action!**autoftp**

```
fred@merlin:~$ ./autoftp4.sh autoftprc
site is guenevere file is m2f_4.2-1_i386.deb \
dir is pub/deb/dists/unstable/contrib/binary-i386
spawn ftp guenevere
Connected to guenevere.
220 guenevere FTP server (Version wu-2.6.0(1)).
Name (guenevere:fred): anonymous
331 Guest login ok, send your complete e-mail \
address as password.
Password:
230-Welcome, archive user anonymous@merlin !
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> binary
200 Type set to I.
ftp> cd pub/deb/dists/unstable/contrib/binary-i386
250 CWD command successful.
ftp> get m2f_4.2-1_i386.deb
local: m2f_4.2-1_i386.deb
200 PORT command successful.
150 Opening BINARY mode data connection for
m2f_4.2-1_i386.deb (1663122 bytes).
226 Transfer complete.
1663122 bytes received in 0.62 secs (2622.1 kB/s)
ftp> quit
221-You have transferred 1663122 bytes in 1 files.
221-Total traffic was 1664135 bytes
in 1 transfers.
221-Thank you for using the FTP service.
221 Goodbye.
```

- will now download files from a remote site
 - currently it is always verbose
 - alter the script to silence the output *unless* the `-v` option is present
- think how you would implement the probe only `-p` switch
 - we leave this as an exercise for the reader