Subject: Socket server in > IDL 6.3
Posted by Robbie on Fri, 27 Jul 2007 00:23:22 GMT
View Forum Message <> Reply to Message

I've been mucking about with the undocumented feature of SOCKET, /
LISTEN in IDL.

Copying the synopsis from a previous post last year

> ;; First create a socket, but market it as a listener.
> ;; socket, <lun>, port, /listen
>
> Socket, 1, 8081, /listen
>
> ;; Then accept a connection on this socket.
> ;; This will listen on the above port (8081).
> ;; This blocks until a connection is made
> ;; Socket, <lun 2>, accept=lun
>
> Socket, 2, accept=1
>
> ;; Once the above routine returns, you can read and write
> ;; the socket that accepted the connection. In this example,
> ;; using lun 2.
> printf, 2, 'cow'
> data= ''
> readf, 2, data
>
> ;; when complete, just close the units.

I've found that FILE_POLL_INPUT is absolutely essential in setting up
a socket server. FILE_POLL_INPUT
has difficulties when you mix listening and connected LUNs, although
you can check each separately by using the TIMEOUT=0 keyword.

I've come across a difficult bug. I want to read at least 100K chunks
of data over the incoming stream. READU produces a runtime error if
the operation blocks. The server seems to work for a while, and then
it suddenly stops and produces the "OPERATION BLOCKS" error. If I exit
from IDL and come back in, the server seems to work ok again. When the
error occurs the (FSTAT(lun)).TRANSFER_COUNT is set to something like
630k and this is the exact same number every time the error occurs
regardless of the size of the chunks of data.

Has anyone else had this experience?

Robbie