

Re: Algorithm for reading from continuous stream

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2004-03/0140.html>

From: James Harris (*no.email.please*)

Date: 03/01/04

Date: Mon, 1 Mar 2004 22:42:48 -0000

"Chee Liang" <cheeliang@DIESPAM.iname.com> wrote in message
news:e9q6409eoi1m1llse235vj9nkibsgoptqn@4ax.com...

> *Hi all,*

>

> *I don't have much experience with reading from continuous stream (I
> hope I'm using the right term here) as opposed to nicely packet-ed
> methods like readline, gets, fgets, etc. Thus hope to get some ideas
> on how to go about doing it.*

>

> *Say I have a winsock stream, and I wish to read in line by line
> (separated by newline). How can I go about doing that neatly? How
> about variable sized packets from the socket?*

>

> *After reading in the 1st time, and processing, I'm sure to have left
> over bytes and the end of the buffer. How can I combine this neatly
> with the next read from the socket? Circular buffer? I thought of
> using a 2*n buffer, but it still might overflow unless every line is
> <=n size.*

TCP will give you a stream input. UDP won't. Both are available under sockets and I presume winsock will do the same. With TCP one peculiarity is it will deliver to the receiving app up-to as many bytes as are requested. If you must have, say, 80 bytes then it is usual to ask for 80, subtract the number received and then if the result is not zero ask for as many as remain. Repeat until request is satisfied.

This isn't exactly what you want but illustrates the byte-by-byte, ie the stream, nature of TCP. To split the incoming stream in to lines you'll probably have to scan for the eol string. You could shift the remainder of the previous read down to the start of the buffer and rerequest. TCP will add the next bytes to the buffer at whatever point you specify. Alternatively if you are worried about the performance of that simple copy-down just split the following read into (from here to end of buffer) and then (from start of buffer to last byte free). In each case use the repeated requests steps, above.

You don't need to worry even if the buffer has one byte left at the end. TCP will happily return that one byte if requested to do so and then the next read will get any bytes which follow. I hope this answers your question. Bear in mind that TCP will worry about the packetising etc. It won't allow you to even see the data stream as packets. Your app will

comp.programming: Re: Algorithm for reading from continuous stream

only be aware of bytes and TCP will deliver up to as many as you request and no more.

--

HTH,

James

For what you are looking