Searching byte[]

Asked 16 years, 2 months ago Modified 16 years, 2 months ago

Viewed 468 times

6

Searching for a string within a string is extremely well supported in .NET but what do you do when the data you need to search isn't a string?

I have binary data arriving in regular chunks via a NetworkStream. Packets are binary but they all start with a signature sequence of bytes. I accumulate the chunks into a larger buffer and look for the start-of-packet signature.

What I'm really looking for is the `byte[]` equivalent to the `String.IndexOf(ss)` method. I have a nasty feeling I'm going to have to implement this myself with a loop and a state machine.

Any suggestions? Over to you!

As suggested, Array.IndexOf(byte) will at least save me an explicit loop. Since posting, it occurred to me to find the first signature byte, then probe ahead for a match where the last signature byte should be, then if they both match try a brute-force comparison for the rest of the string. This approach has the advantage of rejecting false

matches cheaply and allowing me to cheaply reject when I have a partial signature pending another chunk.

Google reveals that the above brilliant plan is a degenerate case of "KMP" or Knuth-Morris-Pratt algorithm. On the bright side if Knuth put his name on it it's probably greased lightning, on the downside why is it that whenever I have a good idea Donald Knuth thought of it 25 years ago?

Since I can't award the points to Donald Knuth I guess they go to Nelson.

.net search binary

Share

Improve this question

Follow

edited Oct 10, 2008 at 7:21

asked Oct 10, 2008 at 5:46

Peter Wone

18.7k ● 14 ● 94 ● 147

3 Answers

Sorted by: Highest score (default)

3

You can use Array.IndexOf to find a single byte.

However, I would caution you that some valid data might accidentally be your signature and completely throw off your application. A better solution in my opinion would be

to always send a four byte integer that contains the size of the packet. Then read that many bytes in to clear the buffer of that packet.

If you are using TCP it is completely acceptable to kick a client off if they lie about packet size or request a stupid amount of memory :)

Share Improve this answer

Follow

answered Oct 10, 2008 at 5:50

user19302

> I don't get to write the protocol, I'm talking to legacy hardware. I do get to write the next version and I've already specified precisely your suggestion. – Peter Wone Oct 10, 2008 at 7:09

2

The fastest algorithms for finding patterns within byte arrays and strings that I have used are Boyer-Moore and simple Boyer-Moore (useful when the pattern is significantly different to the text being searched). I used this to implement a fast mime parser in Java. The code could be easily ported to .Net (the license is LGPL).

Share Improve this answer

Follow

answered Oct 10, 2008 at 9:09

Michael Barker

14.4k ● 4 ● 50 ● 55

Can you use unmanaged/unsafe code? If so I would probably suggest looking into using pointer arithmetic for

0

searching your byte array. That is how strings are effective. You can do similar.

another solution could be to use a dictionary to store your packet data. Have the key be your signature. Then its rather fast and easy to find it. Several ways to have the byte as a key, such as base64string, a simepl wrapper (use KeyedCollection if you do this) etc.

Share Improve this answer

Follow

answered Oct 10, 2008 at 6:35

mattlant

15.4k ● 4 ● 36 ● 44

Actually unmanaged code is a PITA as we have a mixed 32/64 environment. It's amazing how much less hassle this is for pure managed code. Catch-22: I need the signature to parse the stream into packets. – Peter Wone Oct 10, 2008 at 7:05