

Re: How to take in a string of any size?

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2004-08/2382.html

From: Gordon Burditt (gordonb.4s7ny_at_burditt.org)

Date: 08/23/04

Date: 23 Aug 2004 03:40:46 GMT

>IIRC, the OP didn't specify that the input string was coming from
>a file.

>

>If it IS in a file, then the solution is trivial – scan the file
>and count the number of characters you are going to read in then,
>allocate the correct amount of memory, rewind the file, and read
>the contents into the allocated memory.

(1) nobody said the string started at the beginning of the file.
(Consider a compiler parser where the parser limits significant characters in an identifier to only 4YB [1] (and available RAM is limited to 64YB for the entire program). The identifier could start nearly anywhere in the file). Yes, ANSI C lets you have limits a lot smaller, but this compiler vendor decided to document that 4YB identifiers are allowed.

(2) files are not necessarily rewindable, and even if they are, they may not be rewindable efficiently (consider a deck of punch cards or paper tape. Even magnetic tape doesn't rewind very fast.) Then there's these things that sorta look like files that aren't, like pipes and sockets.

>This will probably be
>quicker than reallocating memory so often (though, admittedly, by
>doubling the amount of allocated memory each time your total
>number of calls to realloc() isn't going to be excessive). But if
>you only get one shot at reading each character, then you have to
>store them someplace as they are read.

>Notice that each time you allocated memory you ask for twice the
>amount of memory that you have already used. The only real
>benefit that I can see to repeated realloc calls is if the goal
>is to keep the maximum allocated memory to a minimum. If you can
>afford to allocate so much memory, why not allocate it in blocks
>that get written to once and then stick around to get
>concatenated at the end.

comp.lang.c: Re: How to take in a string of any size?

It may be important that you not refuse requests when you can really handle it. For example, refusing a string only 4.2YB long because you can't get 8YB may not be acceptable. If a `realloc()` request FAILS, you may have to try for a smaller amount.

Gordon L. Burditt

[1] Note: 1 YB = 10^{24} bytes.