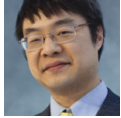


# Why is the line terminator CR+LF?

 [devblogs.microsoft.com/oldnewthing/20040318-00](http://devblogs.microsoft.com/oldnewthing/20040318-00)

March 18, 2004



Raymond Chen

This protocol dates back to the days of teletypewriters. CR stands for “carriage return” – the CR control character returned the print head (“carriage”) to column 0 without advancing the paper. LF stands for “linefeed” – the LF control character advanced the paper one line without moving the print head. So if you wanted to return the print head to column zero (ready to print the next line) and advance the paper (so it prints on fresh paper), you need both CR and LF.

If you go to the various internet protocol documents, such as [RFC 0821 \(SMTP\)](#), [RFC 1939 \(POP\)](#), [RFC 2060 \(IMAP\)](#), or [RFC 2616 \(HTTP\)](#), you’ll see that they all specify CR+LF as the line termination sequence. So the the real question is not “Why do CP/M, MS-DOS, and Win32 use CR+LF as the line terminator?” but rather “Why did other people choose to differ from these standards documents and use some other line terminator?”

Unix adopted plain LF as the line termination sequence. If you look at [the stty options](#), you’ll see that the onlcr option specifies whether a LF should be changed into CR+LF. If you get this setting wrong, you get staircase text, where

```
each
  line
    begins
```

where the previous line left off. So even unix, when left in raw mode, requires CR+LF to terminate lines. The implicit CR before LF is a unix invention, probably as an economy, since it saves one byte per line.

The unix ancestry of the C language carried this convention into the C language standard, which requires only “\n” (which encodes LF) to terminate lines, putting the burden on the runtime libraries to convert raw file data into logical lines.

The C language also introduced the term “newline” to express the concept of “generic line terminator”. I’m told that the ASCII committee changed the name of character 0x0A to “newline” around 1996, so the confusion level has been raised even higher.

[Here’s another discussion of the subject, from a unix perspective.](#)

Raymond Chen

**Follow**

