For a full list of *BASHing data* blog posts, see the index page. RSS

The trouble with Windows CRLF

In 2018 there was hope for Windows end-of-line (CRLF) sufferers. Microsoft announced that the latest version of its text editor "Notepad" would correctly open text files with plain linefeed (LF) line endings:

For many years, Windows Notepad only supported text documents containing Windows End of Line (EOL) characters - Carriage Return (CR) & Line Feed (LF). This means that Notepad was unable to correctly display the contents of text files created in Unix, Linux and macOS...

...This has been a major annoyance for developers, IT Pros, administrators, and end users throughout the community.

Today, we're excited to announce that we have fixed this issue!

Then Microsoft acquired GitHub, and people began to wonder if 2018 was the beginning of the end for CRLF in Windows programs, and an end to the Great Newline Schism. I doubt it, just as I doubt that Microsoft will join the rest of us in the 21st century and move to UTF-8 encoding anytime soon.

A Data Cleaner's Cookbook has a page devoted to finding and removing CR on the command line. Here I give examples of how a Windows carriage return at the end of a line can stuff up some basic command-line operations.

An operation specifying the last character on a line will give the wrong result. Because CR is non-printing, the line *looks* OK but shell tools will recognise CR (\r) as the character at the end of the line, just before the linefeed (LF, \n):

```
$ printf "aaa\nbbb\nccc\n"
aaa
bbb
ccc
$ printf "aaa\r\nbbb\r\nccc\r\n"
aaa
bbb
ccc
$
$ printf "aaa\nbbb\nccc\n" | sed 's/$/X/'
aaaX
bbbX
cccX
$ printf "aaa\r\nbbb\r\nccc\r\n" | sed 's/$/X/'
Xaa
Xbb
Xcc
$
$ printf "aaa\nbbb\nccc\n" \
> | awk '{print gensub(/$/,"X",1,$0)}'
aaaX
bbbX
cccX
$ printf "aaa\r\nbbb\r\nccc\r\n" \
> | awk '{print gensub(/$/,"X",1,$0)}'
Xaa
Xbb
Xcc
$
```

The **rev** command will put a CR at the beginning:

```
$ printf "aaa\nbbb\nccc\n" | rev | cat -v
aaa
bbb
ccc
$ printf "aaa\r\nbbb\r\nccc\r\n" | rev | cat -v
^Maaa
^Mbbb
^Mccc
$
```

---

echo with the *-n* option to omit a newline will fail and hang:

```
$ echo -n $(printf "aaa\n")
aaa$
$ echo -n $(printf "aaa\r\n")
$ a
```

---

tr deleting newlines will fail and hang:

```
$ tr -d '\n' <<< $(printf "aaa\n")
aaa$
$ tr -d '\n' <<< $(printf "aaa\r\n")
$ a
```

---

grepping to match a full line will fail:

```
$ printf "aaa\nbbb\nccc\n" | grep -x bbb
bbb
$ printf "aaa\r\nbbb\r\nccc\r\n" | grep -x bbb
$
$ printf "aaa\nbbb\nccc\n" | grep "^bbb$"
bbb
$ printf "aaa\r\nbbb\r\nccc\r\n" | grep "^bbb$"
$
```

---

reading lines in a **while** loop will fail:

```
$ printf "aaa\nbbb\nccc\n" \
> | while read line; do echo "$line"Y; done
aaaY
bbbY
cccY
$
$ printf "aaa\r\nbbb\r\nccc\r\n" \
> | while read line; do echo "$line"Y; done
Yaa
Ybb
Ycc
$
```

Concatenation of successive lines with **paste** or **sed** will fail:

```
$ printf "aaa\nbbb\nccc\n" | paste - - - -d"|"
aaa|bbb|ccc
$ printf "aaa\r\nbbb\r\nccc\r\n" | paste - - - -d"|"
|ccc
$
$ printf "aaa bbb\nccc ddd\n" | paste -s -d" "
aaa bbb ccc ddd
$ printf "aaa bbb\r\nccc ddd\r\n" | paste -s -d" "
 ccc ddd
$
$ printf "aaa bbb\nccc ddd\n" | sed 'N;s/\n/ /'
aaa bbb ccc ddd
$ printf "aaa bbb\r\nccc ddd\r\n" | sed 'N;s/\n/ /'
 ccc ddd
$
```

AWK will fail on records and fields:

```
$ printf "aaa\nbbb\nccc\n" | awk '{print $0"|"}'
aaa|
bbb|
ccc|
$ printf "aaa\r\nbbb\r\nccc\r\n" | awk '{print $0"|"}'
|aa
|bb
|cc
$
$ printf "aaa bbb\nccc ddd\n" | awk '{print $2 " " $1}'
bbb aaa
ddd ccc
$ printf "aaa bbb\r\nccc ddd\r\n" | awk '{print $2 " " $1}'
 aaa
 ccc
$
```

---

diff and **comm** will respect CR and give unexpected results:

```
$ printf "aaa\nbbb\nccc\n" > fileLF
$ printf "aaa\r\nbbb\r\nccc\r\n" > fileCRLF
$ diff fileLF fileCRLF
1,3c1,3
< aaa
< bbb
< ccc
---
> aaa
> bbb
> ccc
$ comm fileLF fileCRLF
aaa
        aaa
bbb
        bbb
ccc
        ccc
$
```

---

join will give unexpected results:

```
$ cat -v file1
a 1
b 2
e 5
$ cat -v file2
a X
e Y
f Z
$ join file1 file2
a 1 X
e 5 Y
$
$ cat -v file1a
a 1^M
b 2^M
e 5^M
$ cat -v file2a
a X^M
e Y^M
f Z^M
$ join file1a file2a
 X1
 Y5
$
```

Last update: 2019-03-17
