thomd / **LC_COLORS.md**

Last active 29 days ago

LSCOLORS & LS_COLORS

LC_COLORS.md

alternatively use: http://geoff.greer.fm/lscolors/

LSCOLORS

The value of this variable describes what color to use for which attribute when colors are enabled with CLICOLOR. This string is a concatenation of pairs of the format fb, where f is the foreground color and b is the background color.

The color designators are as follows:

```
a     black
b     red
c     green
d     brown
e     blue
f     magenta
g     cyan
h     light grey
A     bold black, usually shows up as dark grey
B     bold red
C     bold green
D     bold brown, usually shows up as yellow
E     bold blue
F     bold magenta
G     bold cyan
H     bold light grey; looks like bright white
```

```
x     default foreground or background
```

Note that the above are standard ANSI colors. The actual display may differ depending on the color capabilities of the terminal in use.

The order of the attributes are as follows:

1. directory
2. symbolic link
3. socket
4. pipe
5. executable
6. block special
7. character special
8. executable with setuid bit set
9. executable with setgid bit set
10. directory writable to others, with sticky bit
11. directory writable to others, without sticky bit

The default is `exfxcxdxbxegedabagacad`, i.e. blue foreground and default background for regular directories, black foreground and red background for setuid executables, etc.

LS_COLORS

LS_COLORS='di=1:fi=0:ln=31:pi=5:so=5:bd=5:cd=5:or=31'

The parameters for LS_COLORS (di, fi, ln, pi, etc) refer to different file types:

```
di      Directory
fi      File
ln      Symbolic Link
pi      Fifo file
so      Socket file
bd      Block (buffered) special file
cd      Character (unbuffered) special file
or      Symbolic Link pointing to a non-existent file (orphan)
mi      Non-existent file pointed to by a symbolic link (visible when you type
ls -l)
ex      File which is executable (ie. has 'x' set in permissions).
```

Color Codes

Through trial and error I worked out the color codes for `LS_COLORS` to be:

```
0 =     Default Colour
1 =     Bold
4 =     Underlined
5 =     Flashing Text
7 =     Reverse Field
31 =    Red
32 =    Green
33 =    Orange
34 =    Blue
35 =    Purple
36 =    Cyan
37 =    Grey
40 =    Black Background
41 =    Red Background
42 =    Green Background
43 =    Orange Background
44 =    Blue Background
45 =    Purple Background
46 =    Cyan Background
47 =    Grey Background
90 =    Dark Grey
91 =    Light Red
92 =    Light Green
93 =    Yellow
94 =    Light Blue
95 =    Light Purple
96 =    Turquoise
100 =   Dark Grey Background
101 =   Light Red Background
102 =   Light Green Background
103 =   Yellow Background
104 =   Light Blue Background
105 =   Light Purple Background
106 =   Turquoise Background
```

These codes can also be combined with one another:

```
di=5;34;43
```

abcarroll commented on Oct 9, 2019

You don't need to specify them all

It seems, at least using bash 5.x, you don't need to specify the entire list. For example, just doing

```
LS_COLORS="ow=1;105;30:di=1;34"
ls -lah
```

Is enough to change the color for other-writable (ow) and directories (di) while keeping the rest at their default.

All the codes

The list of codes is slightly incomplete. I still am not 100% sure what is even responsible for interpreting these, but for example the one I always have issues with is, seemingly files that were 777 -- it would be a bright-on-bright color.

This is, in fact "ow" -- which I assume means "owner writable". Although, following *this* syntax, you'd assume there would also be "or" for other-readable, "oex" or "ox" for other-executable... But nope. That would be too simple for bash! =)

So, the full list is probably quite long, but I can't find it, for sure.

abcarroll commented on Oct 9, 2019

Color Codes

Sorry, as far as the color goes, these are standard VT100 terminal codes. They are the numeric representation of the code used to "begin the color" (or effect), ... The code to begin color can be summed up as `\e[<CODE>m` where \e is escape (ESC or 0x1B (27) in ASCII). To end color you send code "0". So, to start and end writing in a bright red + bold, it would look like:

```
<0x1B>[1;41mHey there, how you doing?<0x1B>[0m
```

Of course, replacing <0x1B> with a byte equaling that.

```
#include <stdio.h>

void main(void) {
    printf("%s", "\e[1;31mHey there, how you doing?\e[0m\n");
}
```

craigbarnes commented on Jan 19, 2020 • edited

> ... as far as the color goes, these are standard VT100 terminal codes

Are they though? It seems a bit strange that a monochrome terminal would have color codes, doesn't it...?

```
$ TERM=vt100 tput colors
-1
```

2bndy5 commented on Sep 14

> as far as the color goes, these are standard VT100 terminal codes

To clarify, the color codes are actually a subset of the standardized ASCII escape sequences. These sequences are typically respected by the shell.