

So You Like Color (The mysterious ^[[characters)

Pradeep Padala

ppadala@cise.ufl.edu

v1.0, 2001-01-01

Revision History

Revision 1.0

2000-12-22

Revised by: ppadala

Submitted to LG

This article explains the mysterious escape ^[[characters sent to the terminal. It shows a simple way of producing colors on a terminal with a C program using the escape sequences. Usage of tput is also explained.

1. Introduction

Have you ever redirected the output of a curses program with colors and wondered what those mysterious ^[[are? Did you ever try to produce colors with out using curses just by a printf ? If answer to any one of these questions is yes, then read on...

This article attempts to explain what those mysterious characters you find in the output of a curses program which produces colors. Then We will try to apply that concept to produce colors with just a printf.

2. Terminal Codes

In the olden days of teletype terminals, terminals used to be away from computers and are connected to them through a serial cable. The terminals could be configured by sending a series of bytes to them. All the capabilities of terminals can be accessed by these byte series which are usually called escape sequences, because they start with an escape(0x1B) character. Even today with vt100 emulation, we can send a series of bytes to the computer which in turn sends to the emulation and converts it to the corresponding functionality. So to get color we can just echo a control code.

Type this on your console.

```
echo "^[[0;31;40mIn Color"
```

The first character is an escape character, which looks like two characters ^ and [. To get that you have to press CTRL+V and then ESC key. All others are normal printable characters. You see the string "In Color" in red. It stays that way and to revert back type this

```
echo "^[[0;37;40m"
```

As you can see it's pretty easy to set color and reset it back. There are hundreds of escape sequences with which you can do a lot of things like moving the cursor, resetting the terminal etc..

3. The Color Code: <ESC>[**{attr};{fg};{bg}**]m

I'll explain the escape sequence to produce color. The sequence to be printed or echoed to the terminal is

```
<ESC>[{attr};{fg};{bg}]m
```

The first character is ESC which has to be printed by pressing CTRL+V and then ESC in VIM or on a linux console. Then {attr}, {fg}, {bg} have to be replaced with one of the following to get the corresponding effect. attr is the attribute like blinking or underlined etc.. fg and bg are foreground and background colors respectively. You don't have to put braces around the number. Just write the number.

{attr} is one of

```
0  Reset All Attributes (Comes to Nomal mode)
1  Bright (Usually turns on the BOLD)
2  Dim
3  Underline
5  Blink
7  Reverse
```

So You Like Color (The mysterious ^[[characters)

8 Hidden

{fg} is one of

30 Black
31 Red
32 Green
33 Yellow
34 Blue
35 Magenta
36 Cyan
37 White

{bg} is one of

40 Black
41 Red
42 Green
43 Yellow
44 Blue
45 Magenta
46 Cyan
47 White

So to get a blinking line with Blue foreground and Green background, you should use

```
echo "^[[5;34;42mIn color"
```

which actually is very ugly. :-)) revert back with `echo "^[[0;37;40m"`

4. With printf()

You want to use this functionality in a c program. Simple! Before you printf some thing print this escape sequence to produce it in the desired color. I have written a small routine `textcolor()` which does this automatically. You can use it in your programs along with the define constants.

```
#include <stdio.h>

#define RESET      0
#define BRIGHT    1
#define DIM        2
#define UNDERLINE 3
#define BLINK      4
#define REVERSE    7
#define HIDDEN     8

#define BLACK      0
#define RED        1
#define GREEN      2
#define YELLOW     3
#define BLUE       4
#define MAGENTA    5
#define CYAN       6
#define WHITE      7

void textcolor(int attr, int fg, int bg);
int main()
```

4. With printf()

So You Like Color (The mysterious ^[[characters)

```
{   textcolor(BRIGHT, RED, BLACK);
    printf("In color\n");
    textcolor(RESET, WHITE, BLACK);
    return 0;
}

void textcolor(int attr, int fg, int bg)
{   char command[13];

    /* Command is the control command to the terminal */
    sprintf(command, "\033[;%d;%d;%dm", attr, fg + 30, bg + 40);
    printf("%s", command);
}
```

The `textcolor()` is modeled against the Turbo C API function. You call the function to set the color and then print with a `cprintf()` (a function used in Turbo C to produce console output with color).

5. The Catch

Then what's the catch? If producing color is so easy why people bother to write such big programs in curses, which in turn queries terminfo in a complex way. As we know, there are many terminals with less capabilities and terminals which don't recognize these escape codes or need different codes to achieve same effect. So if you wanted a portable program which will run on any terminal with same or reduced functionality, then you should use curses. Curses reads the terminfo database and finds the correct codes for the functionality and does the work. Terminfo is a big database which contains information about various functionalities of different terminals.

But if you wanted to write a simple program which produces color on a Linux console, then you can use this info to do it easily. Linux console mostly emulates vt100 and so recognizes these escape sequences.

6. With tput

Well. There's a way to query the terminfo database and do the work. `tput` is the command which can query the database and execute the functionality you specify. The two capabilities `setf` and `setb` are useful to set foreground and background colors. Use this to set foreground color to red and background color to green.

```
tput setf 4 # tput setf {fg color number}
tput setb 2 # tput setb {bg color number}
```

The man page for `tput` explains the usage of `tput`. The manpage for `terminfo` contains a lot of information regarding the terminal capabilities and how to set and get them. I will list the color numbers to be given as arguments to `tput setf` and `tput setb`.

```
0   Black
1   Red
2   Green
3   Yellow
4   Blue
5   Magenta
6   Cyan
7   White
```

Have fun !!!

7. References

- The Text–Terminal–HOWTO at <http://www.tldp.org/HOWTO/Text–Terminal–HOWTO.html>
- Man pages for tput and terminfo.