A Programming Guide For BF++

Jared Derinsk

June 26, 2006

Abstract

BF++ is a superset language of BF, it provides file I/O and networking support to BF. This paper covers both syntax, and some tips to make programming in BF as easy as possible.

1 Syntax

BF++ is just an extension of BF, you can run plain BF programs through the BF++ compiler/interpreter and they will run as planned. The commands inheirited from BF are as follows:

- > - Move to the cell to the right.
- < - Move to the cell to the left.
- + - Increase the value in the current cell by one.
- - - Decrease the value in the current cell by one.
- , - Ask for input from STDIN and places the ASCII value of the input in the current cell.
- . - Print the ASCII character from the value at the current cell.
- [- Do the enclosed if the current value is no zero.
-] - Jump back the matching [is current value is not zero.

Using the above command, it is possible to perform many calculations, however, there is no way to perform any file I/O nor networking, making BF a mostly useless language. This is where BF++ comes into play. Below are the added commands that allow for this functionality:

- # - Goto to the cell number that is the value of the current cell away. Then reads in the following cells until a 0 is found, it uses this as the file name to open. If a file is already open, it closes the file.

- : - Reads in a character from the file and advances the file pointer. It places the ASCII value in the current cell. If it reaches the EOF of the file, it places a 0 in the current cell.
- ; - Writes the ASCII character at the current cell to the file and advances the file pointer.
- % - Goto to the cell number that is the value of the current cell away. Then reads in the following cells until a 0 is found, it uses this as the IP address and port number to open. If a socket is already open, it closes the socket.
- ^ - Sends the ASCII character in the current cell via the open socket.
- ! - Receives a character and places it's ASCII value in the current cell. If the end of transmission is reached, it returns a 0.

As you can see, just by adding these six commands, BF becomes a semi-viable language for other projects, not just a joke.