ndexr

Freddy R. Drennan

2020-12-26

Contents

Chapter 1

About Me

You probably know me from somewhere. My name is Freddy Drennan and I have been in and out of data related projects for the past five years. I initially fell in love with the `R` programming language and best practices in programming development - unit testing, well documented code, quality over quantity type stuff.

I have wanted to break out of my programming rut for a while. Having devoured R, I want a bigger challenge. I'll be blogging about C, C++, and Rust in the coming weeks - taking notes about what I have learned as well and track my own growth.

Thanks for checking this out!

```
print('Hey, yalls, thanks again!')
```

```
## [1] "Hey, yalls, thanks again!"
```

Chapter 2

C

2.1 Basic Calculator

```c
#include <stdio.h>

int main() {
    /*print Fahrenheit-Celsius table for f=0, 20, ..., 300*/
    int lower, upper, step;
    float fahr, celsius;

    lower = 0;
    upper = 300;
    step = 20;

    fahr = lower;
    while (fahr <= upper) {
        celsius = (5.0/9.0) * (fahr-32.0);
        printf("%4.0f %6.1f\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

```
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -fdebug-prefix-map=/build/
## gcc -std=gnu99 -shared -L/usr/lib/R/lib -Wl,-Bsymbolic-functions -Wl,-z,relro -o c6f3f81d84552
```

2.2 Receiving Input

- `printf`: Print to console
- `getchar`: Get a string input from the console

- `putchar`: Print a single character to the screen.

The program below will stop and print 'Enter something...' and then received a string from the user. The while loop iterates over each letter in the input.

```c
#include <stdio.h>

int main()
{
    int c;

    printf("Enter something, or ctrl-D to quit.\n");
    c = getchar();
    while (c != EOF) {
        printf("\n");
        putchar(c);
        c = getchar();
    }
}
```

```
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -fdebug-prefix-m
## gcc -std=gnu99 -shared -L/usr/lib/R/lib -Wl,-Bsymbolic-functions -Wl,-z,relro -o c6
```

Simpler version of the above, showing that we can use assignment within a loop.

```c
#include <stdio.h>

int main()
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(c);
}
```

```
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -fdebug-prefix-m
## gcc -std=gnu99 -shared -L/usr/lib/R/lib -Wl,-Bsymbolic-functions -Wl,-z,relro -o c6
```

```
## For Loop

```c
#include <stdio.h>

int main()
{
    int fahr;

    for (fahr = 0; fahr <= 300; fahr = fahr + 20)
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32));
```
```

```
}
```

```
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -fdebug-prefix-map=/build/
## gcc -std=gnu99 -shared -L/usr/lib/R/lib -Wl,-Bsymbolic-functions -Wl,-z,relro -o c6f3f84844925
```

2.3 DEFINE

```
#include <stdio.h>

#define LOWER 0
#define UPPER 300
#define STEP 20

int main()
{
    int fahr;

    for (fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP)
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32));
}
```

```
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -fdebug-prefix-map=/build/
## gcc -std=gnu99 -shared -L/usr/lib/R/lib -Wl,-Bsymbolic-functions -Wl,-z,relro -o c6f3f8287ee5c
```

2.4 Precedence of Operators

`c = getchar() != EOF` is equal to `c = (getchar() != EOF)`

2.5 Charcter Count

```
#include <stdio.h>

int main()
{
    long nc = 0;
    while (getchar() != EOF)
    {
        nc = nc + 1;
        printf("There are %ld characters\n", nc);
    }
}
```

```
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -fdebug-prefix-map=/build/
## gcc -std=gnu99 -shared -L/usr/lib/R/lib -Wl,-Bsymbolic-functions -Wl,-z,relro -o c6f3f8752e159
```

2.6 Line Counting

```c
#include <stdio.h>

int main()
{
    int c, n1;

    n1 = 0;
    while((c = getchar()) != EOF)
    {
        if (c == '\n')
        {
            printf("%d\n", n1);
            ++n1;
        }
    }
}
```

```
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -fdebug-prefix-m
## gcc -std=gnu99 -shared -L/usr/lib/R/lib -Wl,-Bsymbolic-functions -Wl,-z,relro -o c6
```

2.7 Modern C Example

```c
#include <stdlib.h>
#include <stdio.h>

int main(void) {
    double A[5] = {
            [0] = 9.0,
            [1] = 2.9,
            [4] = 3.E+25,
            [3] = 0.00007
    };

    for (size_t i = 0; i < 5; ++i) {
        printf("element %zu is %g, \tits square is %g\n",
               i,
               A[i],
               A[i]*A[i]);
    }

    return EXIT_SUCCESS;
}
```

```
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -fdebug-prefix-map=/build/
## gcc -std=gnu99 -shared -L/usr/lib/R/lib -Wl,-Bsymbolic-functions -Wl,-z,relro -o c6f3f8600c78b
```

2.8 Parsing a CSV File

```c
#include <stdio.h>

#define ANSI_COLOR_PRIMARY   "\x1b[32m"

void print_color(char* message, char* color) {
    printf("%s", color);
    printf("%s", message);
    printf("%s", "\x1b[0m");
}

int main(void) {
    int c;

    FILE *file;

    file = fopen("/home/fdrennan/Programming/C/modernc/mtcars.csv", "r");

    int n_lines = 0;
    int n_columns = 0;

    if (file) {
        while ((c = getc(file)) != EOF)
        {
            if (c == '\n')
                ++n_lines;
            if ((n_lines < 1) & (c == ','))
                ++n_columns;
            if (n_lines < 5) {
                putc(c, stdout);
                if(c == ',')
                    printf(" ");
            }
        }
        fclose(file);
    }

    print_color("\n\nNumber of lines: ", ANSI_COLOR_PRIMARY);
    printf("%d\n", n_lines - 1);
```

```
    print_color("Number of columns: ", ANSI_COLOR_PRIMARY);
    printf("%d\n", n_columns);

}
```

```
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -fdebug-prefix-m
## gcc -std=gnu99 -shared -L/usr/lib/R/lib -Wl,-Bsymbolic-functions -Wl,-z,relro -o c6
```

Chapter 3

C++

Here is a review of existing methods.

Chapter 4

Rust

```
fn main ()
{
  println!("Yo, this is cool.");
}
```