Hell World: A Hello World Spoof

Sydney Gibbs
sigibbs@smu.edu
Sothern Methodist University

Abstract

During HackSMU IV which took place from September 16th, 2022, to September 18th, 2022, I wrote up a submission dubbed "Hell World". This project is meant to be purely for fun, and not intended for professional or academic use/purposes. The idea of Hell World was to write a Hello World program using one of the fastest programming languages, but to make it as slow and inefficient as possible. The constraints were as follows:

1. **Every piece of code has to contribute to somehow printing hello world (i.e. can't do random mathematical calculations to increase time). In this case, I am not counting pre-processing statements as executable code. Other than that, any other unoptimizations are fair game.**
2. **I am using one of the fastest languages just to add an extra challenge.**
3. **As long as the program would finish in theory given infinite time, I am counting it as a success**

As many of my programming attempts in the academic and professional world usually involve optimizing code, it was a refreshing change of pace to try and do the opposite.

I. Introduction

Hell World was coded in pure C. It was ran using the JetBrains Clion IDE. This code was compiled and executed on an HP x360 Spectre Laptop that has 16 GB of RAM, 1 TB of Storage, and is approximately 3.5 years old.

Hell World took a few main actions in order to make the printing of "Hello World" as painful as possible.

1. For every for loop, the loop goes through EVERY possible integer (all the way from the smallest integer possible, to the largest integer possible).
2. Without extraneous spacing, code was expanded to as many lines as possible.
3. Instead of just printing "Hello World!", there is a for loop that randomly generates a character for each index of a string that is equal to the length of "Hello World!", and the hope is that eventually "Hello World!" is randomly generated. As seen in later calculations, however, the probability of that occurring is very slim.

II. Hell World Pseudocode

```
Repeat until "Hello World!" is generated
    If number of runs > 0
        Calculate previous runtime
        Print previous runtime
    For i = INT-MIN to INT-MAX
        Calculate the length of "Hello World!"
    Allocate memory for expected value of "Hello World!"
    For i = INT-MIN to INT-MAX
        If i = 0
            Expected[i] = 'H'
        Else if i = 1
            Expected[i] = 'e'
        Else if i = 2
            Expected[i] = 'l'
        Else if i = 3
            Expected[i] = 'l'
        Else if i = 4
            Expected[i] = 'o'
        Else if i = 5
            Expected[i] = ' '
        Else if i = 6
            Expected[i] = 'W'
        Else if i = 7
            Expected[i] = 'o'
        Else if i = 8
            Expected[i] = 'r'
        Else if i = 9
            Expected[i] = 'l'
        Else if i = 10
            Expected[i] = 'd'
        Else if i = 11
            Expected[i] = '!'
```

Else if i = 12
Expected[i] = '\0'
Allocate memory for randomly generated value of "Hello World!"
For I = INT-MIN to INT-MAX
If 0 <= i <= length("Hello World!")
Generate Random Number between 32 and 126
Cast that number to a char
Set result[i] to that char
Print result
if result equals expected
break out of loop
else
try again

III. Probability Calculations

Let's start with the probability that one of the characters are right. Let $ASCII_{Max}$ equal the maximum ASCII value used, let $ASCII_{Min}$ equal the minimum ASCII value used, and let x equal the number of possible characters total:

$$ASCII_{Max} = 126$$
$$ASCII_{Min} = 32$$
$$x = ASCII_{Max} - ASCII_{Min}$$
$$x = 126 - 32$$
$$x = 94$$

Since we have 94 possible characters for a given position in a word, we can denote the probability (p) of a character being correctly generated as so:

$$p = \frac{1}{94}$$
$$p \approx 0.0106$$

Since "Hello World!" is 12 characters long, we would need the correct character generated 12 times in a row. Assuming random character generation is a truly random, independent event, the probability of getting all 12 characters in a row to be correctly generated (denoted q) can be calculated like so:

$$q = p^{12}$$
$$q = \left(\frac{1}{94}\right)^{12}$$
$$q \approx 2.101 * 10^{-24}$$

For reference, the probability of you winning the Powerball is approximately $3.422 * 10^{-9}$ [1]. Normally, people try to dissuade you from playing the lottery by talking about astronomical things that have a higher probability of happening. In this case, you are more likely to win the lottery than have this program finish.

IV. Time Calculations

The number of trials until success (denoted y) can be calculated by taking the reciprocal of p [2].

$$y = \frac{1}{p} = \frac{1}{\frac{1}{94}} = \left(\frac{94}{1}\right)^{12}$$
$$y = 475{,}920{,}314{,}814{,}253{,}376{,}475{,}136$$
$$y \approx 4.759 * 10^{23}$$

Assuming it will take that many tries until Hell World correctly prints out "Hello World!", we can determine how much time it would take for the program to do so.

To determine how long it takes for the program to print "Hello World!", I first found out how long it took for one loop of the program to run. I took the average of 5 runs, and used that as the average run time for one iteration:

i. Table I

Trial	1	2	3	4	5	Average
Time (s)	31.50	32.41	32.08	32.48	32.38	32.17

Calculates for how long it would take for the program to successfully finished (denoted t) are as follows:

$$t = y * 32.17$$
$$t \approx 1.5 * 10^{24}\ seconds$$
$$t \approx \frac{(1.5 * 10^{24})}{60}$$
$$t \approx 2.552 * 10^{23}\ minutes$$
$$t \approx \frac{2.552 * 10^{23}}{60}$$
$$t \approx 4.253 * 10^{21}\ hours$$

$$t \approx \frac{4.253 * 10^{21}\ hours}{24}$$
$$t \approx 1.772 * 10^{20}\ days$$
$$t \approx \frac{1.772 * 10^{20}}{365}$$
$$t \approx 4.858 * 10^{17}\ years$$

If for some reason my machine could last that long without dying, this program would far outlive everything we know and hold dear. And all for a single print statement: Hello World!

V. Future Works

If I were to try and make this program run even slower, I would next try to use Python instead of C. I would also add a GUI component to make "Hello World!" show up as an image, instead of on the console. I cannot even begin to fathom how long it would take that program to run, and it may be best to leave it alone so we don't have the despair of finding out.

VI. Conclusion

As I sit here finishing up this paper, I ponder what in my life has led to this moment. Every decision I have ever made, from where to go to college, from what to major in, and even who I chose to hang around: all of it led to this very moment. This program has no practical applications whatsoever. At the end of the day though, sometimes we have to do things for laughs and enjoyment rather than for serious, stuffy reasons.

VII. References

[1]. https://www.cnbc.com/2022/04/23/odds-youll-win-400-million-powerball-jackpot.html#:~:text=If%20you're%20hoping%20to,in%20292%2C201%2C338%20chance%20of%20winning.
[2]. https://www.cut-the-knot.org/Probability/LengthToFirstSuccess.shtml