The Unimaginative Software Engineer

Gonçalo S. Martins

2024-08-14T13:45:18

Chapter 1

Once upon a time, a software engineer had a problem. You see, the engineer fancied himself a writer – for some reason – but the engineer also fancied himself other things, such as a fan of open source, as well as of simplistic structures.

"A problem for the future," the engineer thought. "As long as I stick to plain text files, I should be absolutely fine in the future. What could go wrong?"

And so the engineer wrote. Short stories, novellas, novels, essays, random thoughts, you name it, he wrote them, until he ended up with gigantic directories full of markdown files which, whilst neatly filed by date, topic, manuscript, etc, were no closer to ever being something cohesive.

All this time, the engineer edited his work by reading markdown files. "It's as good as reading something else," the engineer thought, foolishly.

Until one day the engineer was working on something very long, and about thirty files in realised, with some dismay, how nice it would be to be able to read all of that in a single file.

And so a problem is found.

Chapter 2

"The key issue," the engineer thought, "is that of simple concatenation. All I need to do is cat a few files together and kablamo, I can read the entire manuscript!"

In a flurry of ugly code, the engineer wrote some python that would concatenate files according to some index, spitting out a massive markdown file. The engineer then started reading it and realised – with some dismay – that as useful as that file was, there was no way to annotate it!

"Damn and blast!" said the engineer, furious. "So much wasted effort in building a pointless script!"

But many years ago the engineer found someone, and that someone has – for some reason – decided to share her life with the engineer. As the engineer's fury raged across his eyes, she looked up from her iPad and asked

"What the hell are you even talking about?"

And that was it! Eureka! If the manuscript found itself on an iPad, surely the engineer could pencil in some notes!

Chapter 3

And thus an idea is born. If the engineer could code up a tool that imported the manuscript into an intermediate format, then he could write exporters into whatever technologies he needed!

He drew up his requirements. He had to keep the plain text files, of course, as plain text is the ultimate representation. The tool had to happily tie into his existing system for organising files, his so-called index files. The tool had to be extensible to support various exporters, with PDF being the highest priority.

And, of course, the whole thing had to be open source!

"What do I call it, though?" the engineer thought.

Well there was a time somewhere in the 90s where everything was either "Turbo" or "2000". 2000 is long gone, of course, but what's next?

The Manuscript Generator 3000!