

Chapter 8 Notes

March 27, 2018

About this Chapter

This chapter was the last chapter of the book, and its subject was mainly the system calls of unix.

A lot of things remain the same they were when the book was written. For example, now there are two sets of functions to work with files and standard I/O: the `open` family and the `fopen` family. Now at least it's possible to understand the purpose. It's because the I/O between OS and the devices is not per byte. So in order to not reread the same thing every time we need a byte, the library saves an amount of bytes and doesn't read it again if the user asks to read from the same space.

There was also a discussion about the unix file system and about the implementation of directories. It is that this info is now outdated, because even though directories still have *inodes*, they are not files anymore (but it seems symbolic links still are). A lot of the file system related interface remains the same. It might be due to the fact that the designers of the `c` library tried to make the interface implementation independent, so most of functions still work.

Then there was a memory allocation example. It's surprising how simple it is. It's more complicated than the *FIFO* allocator of a previous chapter, but much more powerful. Of course it would be too slow to use it in a real application because everything is linear, which is bad for the performance.

Conclusion

So after reading this book it seems that it contains most of information to understand how `C` works, how to use it and how to implement functions of the `C` standard library.

What to do after reading the book

There are still a lot of things to do after reading this book. But the main subject of this book was the C programming language, that means that after this book, C will not be the main subject. So, in order to understand the OS, one can learn about the implementation of the different components of the operating system, in order to learn about programming languages one can learn about parsers and code generation, and so on... So that means that this book did its job and it's possible to do things using C and to not worry about using it the wrong way, because C is not a very big language and it's actually possible to learn it.