## 信息就是上下文与位

我很难理解上下文这个东西，原文单词是“context”，意思有“语境；上下文；背景；环境”。

呐，这样一来就舒服多了不是，”背景“。在计算机里，一切数据本质上都是01串。

10000001可以代表 -1（有符号数）也可以代表129（无符号数）。因此，那串01到底是什么意思，得看它的环境。

// 又或者看顺序，一堆01串，从什么地方开始读，得到的结果也不一样，如11111111读4位跟读8位的效果肯定是不一样的，当然这是我瞎猜的

## 从一个Hello World开始

Hello World是如何运行的呢？

我在编辑器里写下代码，编译器处理后得到机器语言，然后执行。这是目前我的理解极限。恩，幼稚。完整的过程是这样的。

// C语言版本

1. 代码不是程序，只是文本。“预处理器”接手代码，并按照这段代码中的头文件去加载内容最后得到一串“完整的”代码。
2. 编译器接手“完整的”代码，并把它翻译成汇编代码。
3. 汇编器接手汇编代码，并把它翻译成机器指令。此刻，代码已经变成了一堆二进制文件(0101010111.....)。但它还不能运行。
4. 最后一个阶段叫做链接阶段。源代码中可能调用了其他的函数像printf，scanf之类的。这些函数是单独编译的，要将这些单独编译过的函数的机器指令与hello world的机器指令合并到一起

呐，这里面那个单独编译的过程我不理解哈。怎么单独编译的？首先，计算机必须得知道hello world要调用这个函数才会去编译这个函数。因此，我设想的情况是，计算机一边编译hello world然后看见个没见过的函数，于是就编译它，等编译完了在回头继续编译hello world，也就是DFS。一点小猜想吧。

## shell

外壳，最接近操作系统的程序？这个解释很糟糕啊，因为还得弄懂操作系统是个什么玩意才行。到底什么是操作系统呢？

书上说是命令行解释器，就把它理解成指令好了。