Datawhale

Datawhale 开源社区

Datawhale open source community

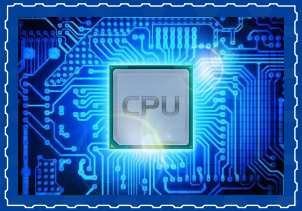
深入理解计算机系统（1）

Computer Systems A Programmer's Perspective CSAPP

李岳昆、易远哲

realyurk@gmail.com、yuanzhe.yi@outlook.com

2021 年 9 月 25 日



第 I 部分

计算机系统漫游-I

编译系统 应用与意义 系统的硬件组成 解释内存中的指令 存储设备

编译系统

Datawhale

首先看一个 hello 程序：

hello

```c
#include<stdio.h>

int main(){
    printf("hello , world");
    return 0;
}
```

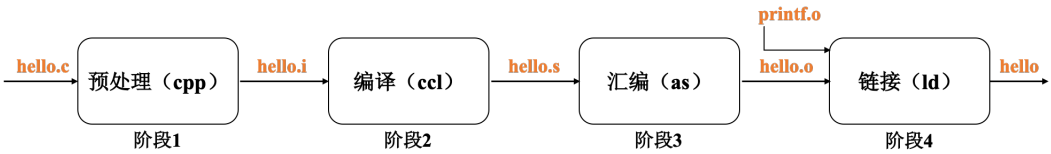
通过下述命令生成可执行程序 hello

- gcc -o hello hello.c

编译系统的工作流程

Datawhale

这个过程虽然是通过一条命令完成的，然而实际上编译系统的处理过程却是非常复杂的，大致可以分为四个阶段，分别为**预处理**、**编译**、**汇编**以及**链接**。



预处理与编译

Datawhale

预处理

预处理器会根据以 # 开头的代码，来修改原始程序。例如 hello 程序中引入了头文件 stdio.h，预处理器会读取该头文件中的内容，将其中的内容直接插入到源程序中，结果就得到了另外一个 C 程序。即：hello.c 经过预处理器后得到为文本文件 hello.i。

编译

编译器将 hello.i 文件翻译成 hello.s 文件，这一阶段包括词法分析、语法分析、语义分析、中间代码生成以及优化等等一系列的中间操作。

汇编与链接

Datawhale

汇编

汇编器根据指令集将汇编程序 hello.s 翻译成机器指令，并且把这一系列的机器指令按照固定的规则进行打包，得到可重定位目标文件 hello.o 。此时 hello.o 虽然是一个二进制的文件，但是还不能执行，还要经历最后一个阶段：链接。

链接

在 hello 这个程序中，我们调用了 printf 函数，这个函数是标准 C 库中的一个函数，存储在名为 printf.o 的文件中。链接器 (ld) 负责把 hello.o 和 printf.o 按照一定规则进行合并。正是因为链接器要对 hello.o 和 printf.o 的进行调整，所以 hello.o 才会被称之为可重定位目标文件。最终经过链接阶段可以得到可执行目标文件 hello。

了解编译系统如何工作是大有益处的

Datawhale

1. 理解编译系统可以**优化程序的性能**.

- 现代编译器是非常成熟的工具，通常可以生成很好的代码，作为一个程序员，我们没有必要为了写出高效的代码，而去研究编译器的内部是如何工作的，但是，我们还是需要对机器执行的代码有一个基本的了解，这样我们就知道编译器把不同的 C 代码转换成的机器代码是什么。
- 我们在写代码的时候可能会有这样的困惑，或者面试中会被问到以下类似的问题：
- 例如：一个 switch 语句是不是要比一连串的 if-else 要高效的多？个函数调用的开销有多大？while 循环比 for 循环更高效么？

了解编译系统如何工作是大有益处的

Datawhale

2. 理解编译系统可以帮助我们**理解链接过程中出现的错误**.

- 如果所有的程序都像 helloworld 一样简单，那的确没有必要去理解编译系统，但是当你试图去构建大型程序的时候，往往涉及到各种函数库的调用，根据以往的经验，一些奇奇怪怪的错误往往都是与链接器有关的。
- 例如：静态变量和全局变量的区别是什么？静态库和动态库的区别是什么？
- 更严重的是，还有一些链接错误直到程序运行的时候才会出现。

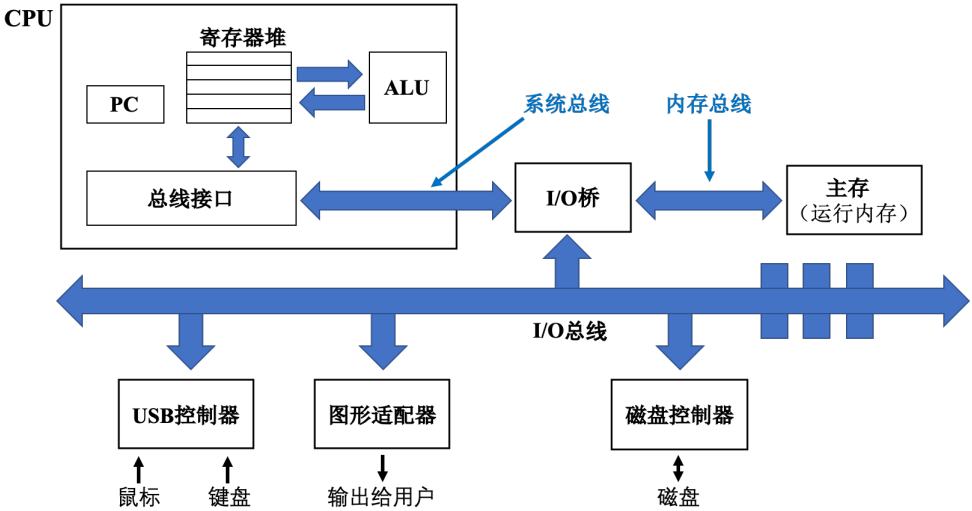
了解编译系统如何工作是大有益处的

Datawhale

3. 避免安全漏洞。

- 多年以来，缓冲区溢出（buffer overflow）是导致互联网安全漏洞的主要原因，如何避免写出的代码存在安全漏洞，第一步就是要理解数据和控制信息在程序栈上是如何存储的，了解不严谨不规范的书写方式会引起什么样的后果。

硬件架构图

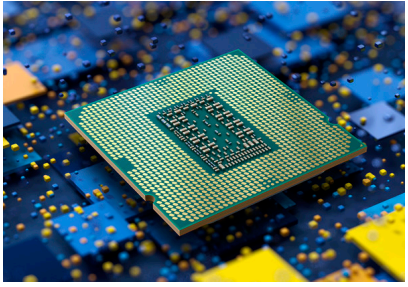

CPU
寄存器堆
PC
ALU
总线接口
系统总线
内存总线
I/O桥
主存
（运行内存）
I/O总线
USB控制器
图形适配器
磁盘控制器
鼠标
键盘
输出给用户
磁盘

CPU 架构

Datawhale

CPU

中央处理单元（Central Processing Unit，CPU），也称处理器，包含 PC（程序计数器：Program Count）、寄存器堆（Register file）、ALU（算数/逻辑计算单元：Arithmatic/logic Unit）三个部分.



CPU 内部结构含义

Datawhale

- **程序计数器 PC**：是一个 4 字节或是 8 字节的存储空间，里面存放的是某一条指令的地址。从系统上电的那一瞬间，直到系统断电，处理器就不断地在执行 PC 指向的指令，然后更新 PC，使其指向下一条要执行的指令（注意：这个下一条指令与刚刚执行的指令不一定是相邻的）
- **寄存器**：可以理解为一个临时存放数据的空间。例如计算两个变量 a+b 的和，处理器从内存中读取 a 的值暂存在寄存器 X 中，读取 B 的值暂存在寄存器 Y 中，这个操作会覆盖寄存器中原来的数值，处理器完成加载的操作后，ALU（Arithmatic/logic Unit）会从复制寄存器 X 和 Y 中保存的数值，然后进行算术运算，得到的结果会保存到寄存器 X 或者寄存器 Y 中，此时寄存器中原来的数值会被新的数值覆盖。
- **算数/逻辑计算单元 ALU**：计算速度极快，且专攻算数与逻辑的计算，计算机核心部分.

Datawhale

内存

主存（Main Memory），也称为内存、运行内存，处理器在执行程序时，内存主要存放程序指令以及数据。从物理上讲，内存是由随机动态存储器芯片组成；从逻辑上讲，内存可以看成一个从零开始的大数组，每个字节都有相应地址.

总线

内存和处理器之间通过**总线**来进行数据传递。实际上，总线贯穿了整个计算机系统，它负责将信息从一个部件传递到另外一个部件。通常总线被设计成传送固定长度的字节块，也就是字（word），至于这个字到底是多少个字节，各个系统中是不一样的，32 位的机器，一个字长是 4 个字节；而 64 位的机器，一个字长是 8 个字节.

输入输出设备

Datawhale

输入输出设备

除了处理器，内存以及总线，计算机系统还包含了各种输入输出设备，例如键盘、鼠标、显示器以及磁盘等等。每一个输入输出设备都通过一个**控制器**或者**适配器**与 IO 总线相连.

区别

控制器与适配器主要区别是在于它们的封装方式，无论是控制器还是适配器，它们的功能都是在 IO 设备与 IO 总线之间传递数据.

hello 程序执行过程中发生了什么?

Datawhale

操作过程

- hello.c 经过编译系统得到可执行目标文件 hello，此时可执行目标文件 hello 已经存放在系统的磁盘上，那么，如何运行这个可执行文件呢?
- 在 linux 系统上运行可执行程序：打开一个 shell 程序，然后在 shell 中输入相应可执行程序的文件名：
- linux>./hello

shell 是什么?

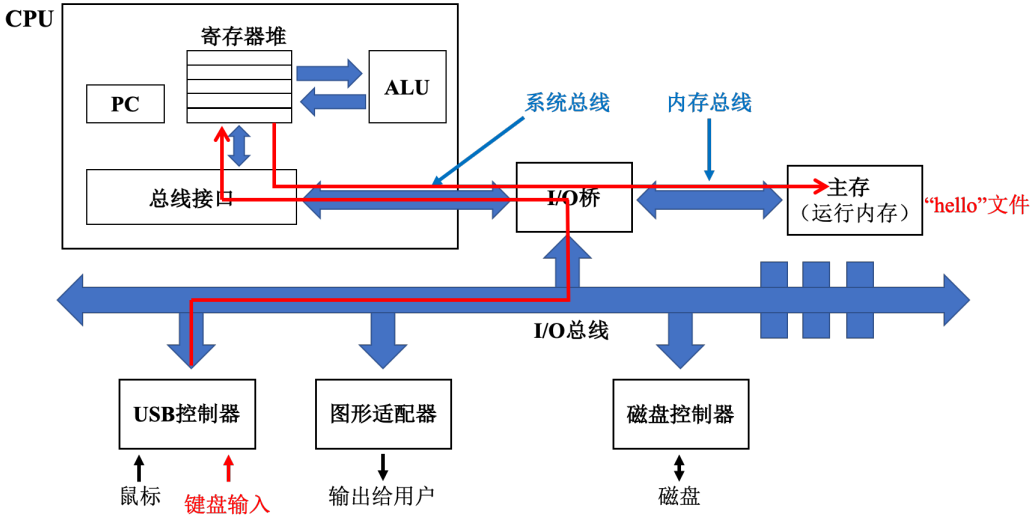
shell 是一个命令解释程序，如果命令行的第一个单词不是内置的 shell 命令，shell 就会对这个文件进行加载并运行. 此处，shell 加载并且运行 hello 程序，屏幕上显示 hello,world 内容，hello 程序运行结束并退出，shell 继续等待下一个命令的输入.

程序执行流程

1. 首先我们通过键盘输入"./hello" 的字符串，shell 程序会将输入的字符逐一读入寄存器，处理器会把 hello 这个字符串放入内存中。
2. 当我们完成输入，按下回车键时，shell 程序就知道我们已经完成了命令的输入，然后执行一系列的指令来来加载可执行文件 hello。
3. 这些指令将 hello 中的数据和代码从磁盘复制到内存。数据就是我们要显示输出的"hello , world\n" ，这个复制的过程将利用 DMA（Direct Memory Access）技术，数据可以不经过处理器，从磁盘直接到达内存。
4. 当可执行文件 hello 中的代码和数据被加载到内存中，处理器就开始执行 main 函数中的代码，main 函数非常简单，只有一个打印功能。

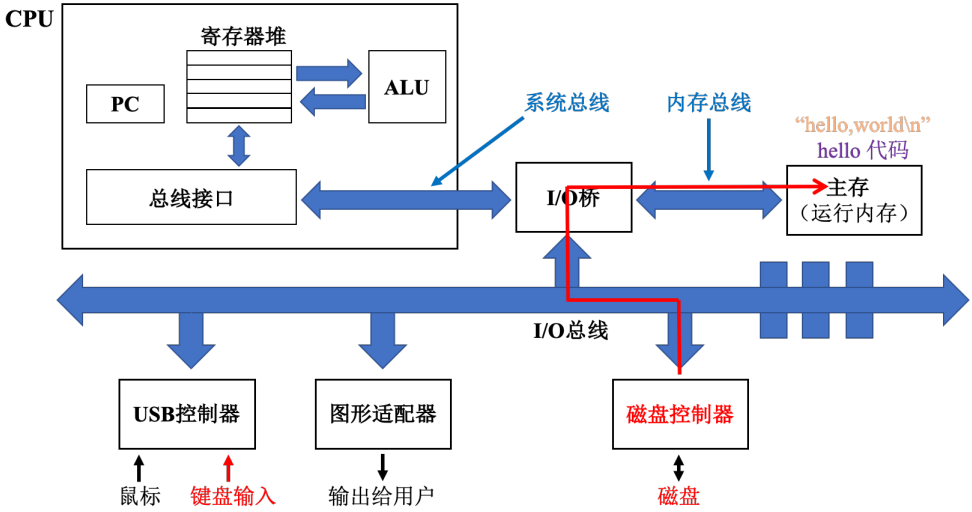
hello 程序执行过程中发生了什么?

Datawhale



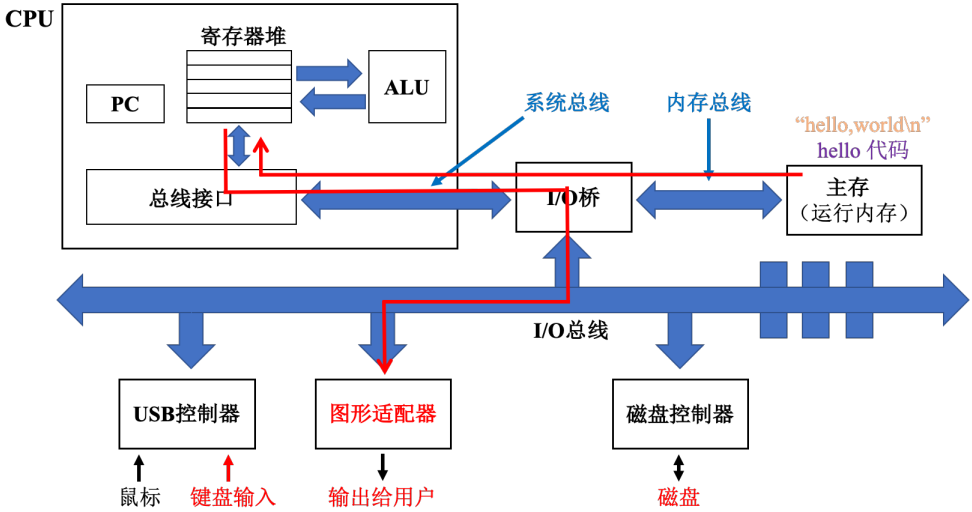
hello 程序执行过程中发生了什么?

Datawhale



hello 程序执行过程中发生了什么?

Datawhale



思考与拓展

从 hello 程序执行的过程来看，系统即使执行如此简单的程序，数据信息仍旧需要在磁盘、内存、处理器以及 IO 设备之间进行搬运。

数据从一个地方搬运到另外一个地方需要花费时间，系统设计人员的一个主要任务就是缩短信息搬运所花费的时间。

设备容量

通常情况下，大容量的存储设备的存取速度要比小容量的慢，运行速度更快的设备的价格相对于低速设备要更贵。例如：在一个系统上，磁盘的容量一般为 TB 级，内存的容量一般为 GB 级，磁盘的容量大概是内存的 1000 倍。

表: 存储容量对比

设备种类	存储容量
寄存器文件	100∼1000 B
内存	1∼100 GB
磁盘	1∼1000 TB

Cache 至关重要

Datawhale

对于处理器而言，从磁盘上读取一个字所花费的时间开销比从内存中读取的开销大 1000 万倍。寄存器文件的只能存储几百个字节的信息，而内存的可以存放几十亿的字节信息（GB 级），从寄存器文件读取数据比从内存读取差不多要快 100 倍。

随着半导体技术的发展，处理器与内存之间的差距还在持续增大，针对处理器和内存之间的差异，系统设计人员在寄存器文件和内存之间引入了高速缓存（cache），比较新的，处理能力比较强的处理器，一般有三级高速缓存，分别为 L1 cache，L2 cache 以及 L3 cache。

L1 cache 的访问速度与访问寄存器文件几乎一样快，容量大小为数万字节（KB 级别）；L2 cache 的访问速度是 L1 cache 的五分之一，容量大小为数十万到数百万字节之间；L3 cache 的容量更大，同样访问速度与 L2 cache 相比也更慢。

存储设备的层次结构

Datawhale

整个计算机系统的信息存储可以用一个层次结构来表示，通常而言，存储容量越小，速度越快，价格越高，上一层存储设备是下一层存储设备的高速缓存.