第一单元学习笔记

yinxuhao [xuhao_yin@163.com]

December 12, 2022

Contents

1 信息就是位 + 上下文

1. 源程序实际上就是一个由 0 和 1 组成的位 (又称为比特) 序列，8 个位被组织成一组，称为字节。1 byte = 8 bits

2. 系统中所有的信息——包括磁盘文件、内存中的程序、内存网络中存储的的用户的数据，**都是由一串比特表示**。区别不同数据对象的唯一方法是**我们读到的数据对象时的上下文**。

2 程序被其他程序翻译成不同的格式

`gcc -o hello hello.c` 可以读取源文件`hello.c`，并将其翻译成一个可执行目标文件`hello`。翻译过程分为 4 个阶段，执行这四个阶段的程序分别是预处理器、编译器、汇编器和链接器，它们一起构成了编译系统。

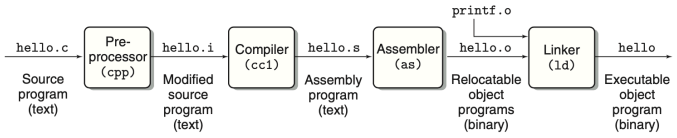

Figure 1: 编译系统

2.1 预处理阶段

`#include <stdio.h>`命令告诉预处理器 (`cpp`) 读取系统头文件内容，并把它直接插入程序文本中。结果得到另一个 C 程序，通常以`.i`作为文件扩展名。

2.2 编译阶段

编译器 (`ccl`) 将文本文件`hello.i`翻译成文本文件`hello.s`，它包含一个汇编语言程序。

2.3 汇编阶段

汇编器 (`as`) 将`hello.s`翻译成机器指令，把这些指令打包成一种叫做可重定位目标程序 (relocatable object program) 的格式，并将结果保存在目标文件`hello.o`中。该文件为二进制文件。

2.4 链接阶段

链接器 (`ld`) 负责将预编译的目标文件 (例如`printf.o`) 合并到`hello.o`中。结果是可执行目标文件 (简称可执行文件)`hello`，可以被加载到内存中，由系统执行。

3 了解编译系统的益处

1. 优化程序性能
2. 理解链接时出现的错误
3. 避免安全漏洞

4 运行可执行程序

在shell中执行可执行程序：./hello

shell是一个命令行解释器，它输出一个提示符，等待输入一个命令行，然后执行该命令。

4.1 系统的硬件组成

4.1.1 总线

一组电子管道，它贯穿整个系统。它携带信息字节并负责在各个部件间**传递**。通常被设计成**传送定长的字节块**，也就是**字 (word)**。

字中的字节数 (即字长) 是一个基本的系统参数，各个系统都不尽相同。64 位系统是 8 个字节。我们需要明确在上下文中一个字的大小。

4.1.2 I/O 设备

系统与外部世界的联系通道。一般由：

作为用户输入的键盘鼠标，作为用户输出的显示器及长期存储数据和程序的磁盘驱动器。

每个I/O设备都通过一个控制器 或适配器 与I/O总线相连。

控制器和适配器之间的区别在于它们的封装方式。控制器是I/O设备本身或者系统的主印制电路板 (主板) 上的芯片组。而适配器是插在主板卡槽上的卡。它们的功能都是在I/O总线和I/O设备之间传递信息。

4.1.3 主存

主存是**临时存储**设备。在处理器执行程序时，用来存放程序和程序处理的数据。

从物理上讲，主存是由一组动态随机存取存储器 (DRAM) 芯片组成的。从逻辑上讲，存储器是一个线性的字节数组，每个字节都有其唯一的地址 (数组索引)，这些地址从零开始。

4.1.4 处理器

中央处理单元，是解释存储在主存中指令的引擎。

处理器的核心是以恶搞大小为一个字的存储设备 (或寄存器)，称为程序计数器 (PC)。在任何时刻，PC都指向主存中某条机器语言指令 (含有该指令的地址)

处理器从系统通电开始，直到系统断电，一直在不断执行程序计数器指向的指令，再更新程序计数器，使其指向下一条指令。处理器看上去是按照一个非常简单的指令执行模型来操作，这个模型是由**指令集架构**决定的。

简单操作包括：

加载：从主存复制一个字节或一个字到寄存器以覆盖寄存器原来的内容。

存储：从寄存器复制一个字节或者一个字到主存的某个位置，以覆盖这个位置上原来的内容。

操作：把两个寄存器的内容复制到ALU，ALU对这两个字做算术运算，并将结果存放到一个寄存器中，以覆盖其原来的内容。

跳转：从指令本身中抽取一个字，并将这个字复制到程序计数器 (PC) 中，以覆盖PC原来的值

需要区分处理器的**指令集架构**和处理器的**微体系结构**：

指令集架构描述的是每条机器代码指令的效果；微体系结构描述的是处理器实际上是如何实现的。

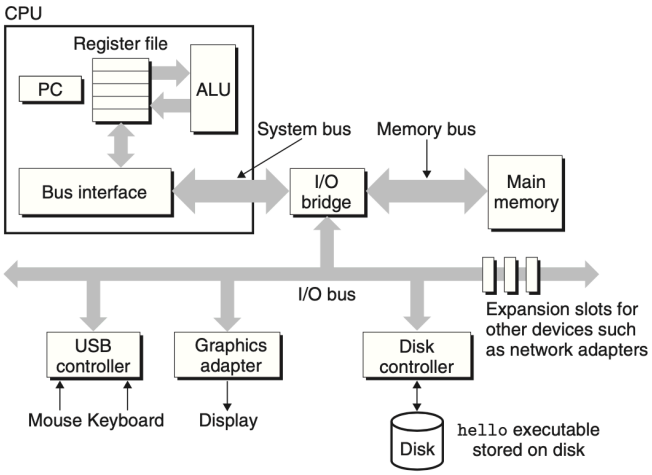

Figure 2: 一个典型系统的硬件组成

4.2 运行 hello 程序

当在shell中键入./hello后，shell将字符逐一读入寄存器，再把它存放到内存中。见图片Figure 3敲击回车后，shell就知道了命令已结束。然后会执行一系列操作来家在可执行的hello文件，这些指令将hello目标文件中的代码和数据从磁盘复制到主存。

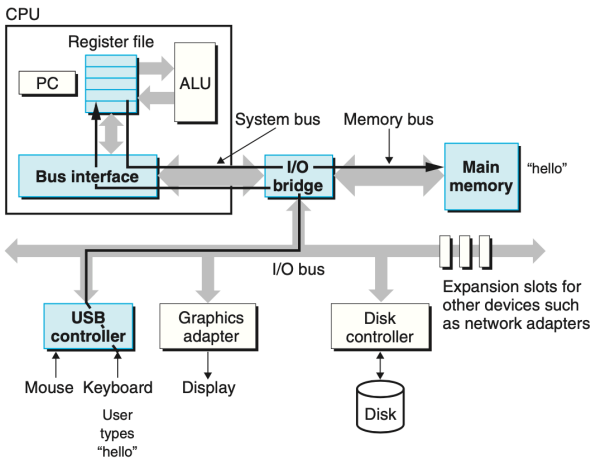

Figure 3: 从键盘上读取 hello 命令

利用直接存储器存取 (DMA)，数据可以不通过处理器直接从磁盘到达主存。该步骤可见图Figure 4。

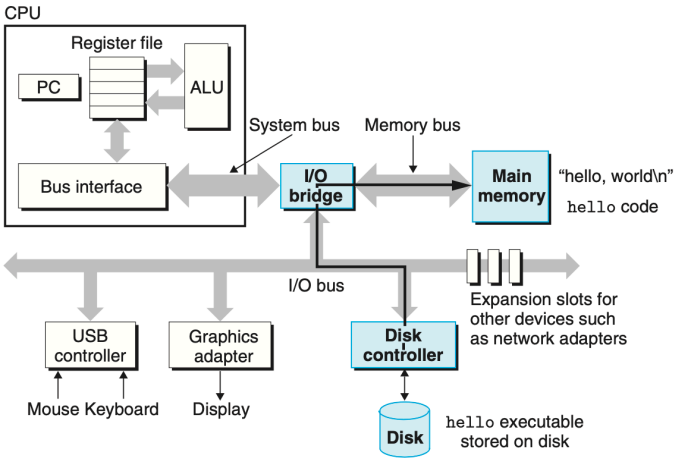

Figure 4: 从磁盘加载可执行文件到主存

一旦数据被加载到主存，处理器就开始执行机器语言指令。这些指令将"hello, world\n" 字符串中的字节从主存中复制到寄存器文件，再从寄存器文件中复制到显示设备，最终显示在屏幕上，见图Figure 5.

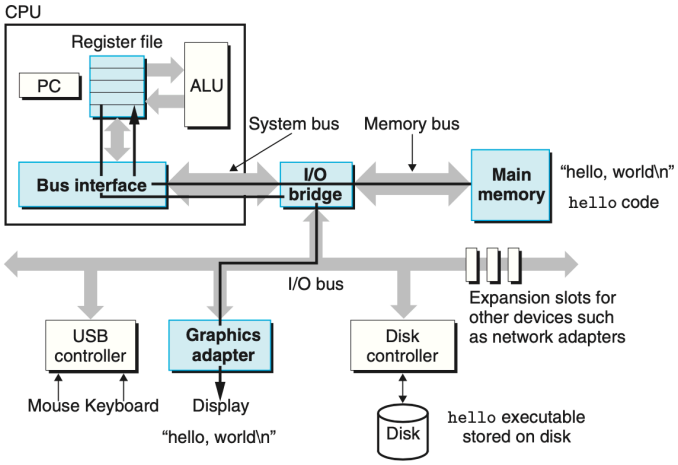

Figure 5: 将输出字符串从存储器写到显示器