

数码时钟汇编程序 - 实验报告

数据科学与计算机学院 2017 级 软件工程 4 班 郑佳豪 16305204

实验目的

- 了解 x86 平台下的汇编程序开发过程，以及掌握 emu8086 和 DOSBox 工具的使用

实验内容

使用 x86 汇编语言编写数码时钟程序，并能在 DOSBox 成功运行

实验设计

通过自顶向下的设计方法，可将程序划分为三大模块：获取时间、打印时间、中断处理。在划分模块后，可进而落实各模块的实现与测试。

实验过程

在获取时间模块中，我们通过 int 21h 的 2Ch 指令实现系统时间的读取，读取的时、分、秒分别存储于 CH、CL、DH 寄存器。

在打印时间模块中，我们需要对数值的个位、十位进行拆分，以实现数字字符的打印，故我们使用了 div 指令求取数值与 10 的商和余数。在得到个位、十位数值后，我们通过逐行定点打印数据段中的数字字符数据，实现了时、分、秒的显示。

在中断处理模块中，我们通过 int 16h 的 01h 指令获取键盘缓冲区的状态，进而根据缓冲区是否存在数据来判断是否中断程序。

实验心得

总体来说，本次实验是成功的。经过这次实验，我能基本掌握 x86 汇编程序设计的基本方法，能熟悉 emu8086 和 DOSBox 工具的使用。

在实现键盘中断程序的模块时，我走了不少的弯路，一开始想通过 int 21h 的 25h 指令实现预期功能，但始终无法成功。后来，又转而使用了 int 21h 的 0Ah 指令，想通过判断读取的缓冲数据来实现预期功能，但也无法成功。最后，StackOverflow 的一些回答给了我一些启发，我通过在程序的主循环中判断键盘缓冲区的有无数据，来判断用户是否触发键盘事件，进而判断是否中断程序。

在编写代码的过程中，为了使开发流程变得清晰，我先后使用了多种方式组织代码，如注释、空行等，后来发现通过标签和缩进的方式划分代码，是一种很好的方式。

在寄存器管理和子程序跳转的方面，x86 较 MIPS 提供了一种对更友好的开发方式。但开发者还是得注意寄存器的管理，要避免寄存器的相互污染。在开始开发 print_digit 子模块的过程中，由于自己开发的不细致，没有注意到存储数值的个位、十位的寄存器与子模块的临时寄存器的冲突，出现了数值打印异常的问题。

这次的课程设计，我学到了不少的底层知识，对计算机的底层原理有了更进一步的了解，希望自己能在今后的计算机组成原理课程中，能够学习与掌握更多相关知识。