

实验报告

211240045 杨镇源

实验进度：

我完成了所有必做内容。

必做题：

必做题 1：站在计算机的角度来理解一下，计算机究竟是怎么通过指令方式来计算 $1+2+\dots+100$ 的。

计算 $1+2+3+\dots+100$ 这个“程序”，实际上是产生了一段指令去指挥 CPU 执行这个“程序”。这段指令产生后以一串 01 序列的形式保存在存储器中，激励 CPU 运转。其是一个“取指令→CPU 指令译码、PC 改变→CPU 进行运算→存储返回值→再取指令”周而复始的过程。直到达到循环条件，退出或者进入死循环。通过重复读取一条又一条 01 序列组成的机器指令，CPU 里的组件各司其职，最终计算出 $1+2+\dots+100$ 的结果。

必做题 2：程序是个状态机，画出计算 $1+2+\dots+100$ 的程序的状态机。

$\dots \rightarrow (4,1,1) \rightarrow (2,1,2) \rightarrow (3,3,2) \rightarrow (4,3,2) \rightarrow (2,3,3) \rightarrow (3,6,3) \rightarrow (4,6,3) \rightarrow \dots \rightarrow (2,4851,99) \rightarrow (3,4940,99) \rightarrow (4,4940,99) \rightarrow (2,4940,100) \rightarrow (3,5040,100) \rightarrow (4,5040,100) \rightarrow (5,5040,100) \rightarrow \dots$

必做题 3：理解基础设施。

调试所需时间： $500 \times 90\% \times 30s \times 20 = 270000s = 4500min$ 。

使用 SDB 节省的时间： $500 \times 90\% \times 20s \times 20 = 3000min$ 。

必做题 4：假设你现在需要了解一个叫 selector 的概念，请通过 i386 手册的目录确定你需要阅读手册中的哪些地方。

Intel 80386 Programmer's Reference Manual 中 5.1.3 节，也就是 96 面。

必做题 5：RTFM 理解了科学查阅手册的方法之后，请你尝试在你选择的 ISA 手册中查阅以下问题所在的位置，写出需要阅读的范围。

一、riscv32 有哪几种指令格式？

Chapter 6: RISC-V Privileged Instruction Set Listings。Page 115 – 116。

二、LUI 指令的行为是什么？

2.4 2.4 Integer Computational Instructions。Page 19。

三、mstatus 寄存器的结构是怎么样的？

3.1.6 Machine Status Registers (mstatus and mstatush)。Page 20-28。

必做题 6：Shell 命令，统计代码行数。

nemu 文件夹中.c/.h 文件代码行数总和（包括空行）=24029；

shell 命令：wc -l `find . -name '*.c' -o -name '*.h'`；

在 PA1 中编写的代码=24029-23467=562；

nemu 文件夹中.c/.h 文件代码行数总和（除去空行）=20826；

shell 命令：find . -name "*.c" -o -name "*.h"|xargs cat|grep -v ^\$|wc -l。

必做题 7：编译选项-Wall 和-Werror。

-Wall:打开 gcc 的所有警告；

-Werror：要求 gcc 将所有的警告当成错误进行处理。

为什么要使用-Wall 和-Werror 编译选项：在编译时刻把潜在的 fault 直接转变成 failure，同时侦测代码中的未定义行为，尽早地展示编程过程中可能产生的 bug，降低代码的调试难度。

实验心得：

PA1 代码量不是很大，我认为本次实验更加困难的点在于对于源代码的理解，以及熟悉 NEMU 到底是怎样跑起来的。在 RTFSC 的过程中，我学习到了 NEMU 工程代码中很多对我来说十分新颖，且增强代码可读性的写法。例如，ENUM，宏定义等等。我认为这些写法有助于降低编程的错误率，并且提高 debug 的效率，是非常值得学习的。