

# PA1实验报告

姓名：何伟      学号：171240537

2018 年 10 月 31 日

## 摘要

在docker下完成了pa1.1的内容，并将实验环境移至debian虚拟机下完成了pa1.2和pa1.3。

## 1 实验进度

### 1.1 PA1.1

已全部完成并且完成得较为顺利。

但是用docker的container始终跑不出超级玛丽，在做PA1.2时换成了虚拟机，并成功运行超级玛丽。

寄存器结构体主要考察寄存器之间的相对关系，在阅读有关寄存器结构的内容后已实现。

单步执行通过调用cpu\_exec函数实现；打印寄存器很顺利，直接输出cpu.e\*\*即可；扫描内存卡了一点，找了好久才发现如何访问内存中的值，通过输出pmem数组中值实现。

### 1.2 PA1.2

已全部完成并且完成得较为困难。

首先是对正则表达式有点畏惧，不过好在pa中要求掌握的并不多。刚开始做会有点晕，不断地尝试，对代码框架有了较为清晰地印象之后实现起来也不是很麻烦。

表达式生成器也很顺利地完成。其中，将溢出情况赋值为“0 0”，除0的情况在expr.c中生成报错信息。

最后用表达式生成器生成的数据验证了表达式求值的大概率正确性。

### 1.3 PA1.3

已全部完成并且完成地较为顺利。

实现了表达式求值功能后，扩展顺利完成。

监视点根据讲义上的框架已实现。

## 2 必答题

### 2.1 selector

在i386手册5.1.3节的Selectors中。

### 2.2 理解基础设施

在调试上花费的时间 $t = 500 \times 0.9 \times 30 \times 20 = 270000s$

简易调试器节省的时间 $t' = t \times \frac{2}{3} = 180000s$

### 2.3 查阅i386手册

CF:需要阅读P34页大致介绍；P50页；P52页，P419页。

ModE/M:需要阅读P40大致介绍，P240内容，17.2.1的内容。

mov:P345页；P346页 P347页；P348页

### 2.4 shell

完成PA1内容后总共有代码4387行，用了shell命令

```
find . -type f -name '*.c' -o -name '*.h' -exec wc -l {} \;
```

切换至pa0 branch之后统计代码行数，有3798行，因此在PA1中编写了589行。

去除空行后有代码3655行，用了shell命令

```
find . -name '*.h' -or -name '*.c' -exec grep -v '^$' {} \; | wc -l
```

### 2.5 gcc编译

-Wall是编译之后显示所有警告。

-Werror视警告为错误，出现警告则编译错误。

很多时候我们写的bug并不是failure，faliure并不可怕，更可怕的是fault，这些bug隐藏在我们的程序中，不知道什么时候会引发难以挽回的错误。因此我们尽可能早的发现fault，而-Wall和-Werror选项会尽可能的将fault转化为failure。

## 3 实验遇到的问题

1、始终没有在docker中跑出超级玛丽，STFW未果。

2、在生成表达式测试的时候出现了括号匹配错误问题，后来发现是cese里面少了break。改好了之后发生段错误，改了'\0'的位置之后正常。

## 4 实验心得

做完PA1，深刻感觉到了自己对C语言的不了解，很多用到的函数都是以前没有听说过的。可惜做完了才想到记录一些有用的函数，有点迟。实验过程中，也有一些时候

做着做着不知道自己在干嘛了，源于对代码框架还不够理解，思路不清晰。希望以后在做什么东西的时候先构思好，理清思路在动手。

做完了实验，对计算机的结构和一些功能的实现有了大体的了解。KISS法则尤为重要，实现gdb的功能不可能一步完善，想要一开始就做的最好反而会因为bug太多且找不到而陷入僵局，先实现一个大体的简陋的版本在一步步添加新代码是一个不错的选择。

命令行是一个神奇的东西，一个个不起眼的小命令，通过管道链接会有一些很有用的奇特功效。