

# PA4 实验报告

姓名：何伟      学号：171240537

2019 年 1 月 6 日

## 摘要

完成所有必做内容。

## 1 实验进度

### 1.1 PA4.1

主要是理解讲义的内容，理解后按照讲义的按要求实现起来很快。不过在 `irq.c` 中的 `yield` 之后调用 `schedule` 没有设返回值，导致 `asm_trap` 中的 `eax` 返回地址不对，找了会 bug。

### 1.2 PA4.2

做起来很麻烦，根据 `ics` 书和手册上的内容理解分页机制，地址转换实现了 PA4.2 中的几个函数，测试的时候遇到了 `present` 位 `assert` 的问题，找了好久的 bug。

### 1.3 PA4.3

实现起来较为顺利。

## 2 必答题

先从 `main` 函数开始，由于定义了宏 `HAS_VME`，会执行 `init_mm`，接下来也是一些初始化的工作。在 `init_proc` 中，调用了 `context_uoload` 函数，在这里面调用 `loader` 进行程序的加载，并完成虚拟地址与物理地址的映射，返回程序的入口地址，之后调用了 `__ucontext` 函数创建上下文。之后触发了 `__yield` 函数，开始中断处理。会调用 `asm_trap`，进行上下文的恢复，跳转到加载的程序的入口。

由于实现了 `__EVENT_IRQ_TIMER`，每个一段时间就会调用 `__yield`，每隔一段时间就会触发自陷，捕捉到事件会调用 `__schedule` 对进程进行调度，`__schedule` 会执行与当前进程不同的另一个进程，实现了两个的分时运行。同时，通过分页机制，使得两个程序可以在虚拟地址上执行，但是由于映射不同，两个程序的物理地址上的内容不会相互覆盖。

### 3 实验心得

PA4 总体难度较大，也可能是上课没怎么听 (小声)，对堆区管理，虚拟地址这一部分不太熟悉，做实验是个很好的复习方式，可能写的时候还不太理解，但是 debug 会让人真正的理解。讲义的内容非常简略，但是一些十分重要的信息会在讲义中，要一遍一遍地仔细读。在实现函数时做好出错地报错显示确实很有帮助，比如 present 位，做后面的内容大部分时间都在调试，通过这个 assert 判断，确实也发现了不少 bug。不过时间太紧了，讲义上推荐的很多有意思的内容都没有看。