

计算机系统基础
Programming Assignment

PA 0 实验环境配置与实验总览

2017年9月8日

课程信息

- PA授课时间
 - 单周周五1-2节（三班）或5-6节（二班）
- 授课团队
 - 讲师：汪亮，助教：何知涵、沈明杰、姚荣春
- 课程信息页（随时更新）
 - <https://cs.nju.edu.cn/wangliang/pa2017.html>
 - 确保能登录cms系统
 - 如有问题邮件联系吴海军老师：hjwu@nju.edu.cn
 - 下载教程（并关注更新）
 - 下载i386手册
 - 加入课程QQ群（按要求输入验证消息并修改群名片）

配置实验环境 – 操作系统

- 安装虚拟机软件
 - Vmware
 - VirtualBox
- 下载debian操作系统iso镜像
 - 32位 (i386)
 - 最小化安装 (Network install)
- 新建虚拟机并安装32位debian操作系统
 - 语言选择US_en
 - 地区选择中国并选择国内的源 (source)
 - Locale选择united states
 - 选择一个桌面环境, 推荐MATE

配置实验环境 – 实验环境

- 配置客户操作系统和开发环境
 - 将用户添加到sudoer list, 不允许使用root用户做实验
 - 安装vmware-tools
 - 安装build-essential libreadline-dev libsdl1.2-dev vim git
 - 请注意版本号：gcc 版本是否是6.x很关键，SDL库必须是1.x版本而非2.x版本

类型	名称	版本
虚拟机	VMware® Workstation 12 Player	12.5.6 build-5528349
客户操作系统	Debian GNU/Linux 9 (i386)	9.0
桌面环境	MATE	1.16.2
编译器	gcc	6.3.0
readline开发包	libreadline-dev	7.0-3
SDL开发包	libsdl1.2-dev	1.2.15+dfsg1-4

配置实验环境 – 获取框架代码

- 获取框架代码

- 执行git clone <https://github.com/ics-pa/pa2017.git>
- cd ./pa2017/
- 修改Makefile.git中的STU_ID为自己的学号
- make clean

- 框架代码的结构

- 安装tree
- tree pa2017/

```
pa2017/
├── game                // 包含游戏相关代码
├── include             // PA整体依赖的一些文件
│   ├── config.h       // 一些配置用的宏
│   └── newlib          // 公共用的库
├── kernel             // 一个微型操作系统内核
├── Makefile            // 帮助编译和执行工程的Makefile
├── Makefile.git       // 和git有关的部分
├── nemu               // NEMU
│   └── src
│       └── main.c      // NEMU入口
└── testcase           // 测试用例
```

提交规则

- 在CMS系统中提交打包的代码和报告（以及调查问卷）
- 使用make submit命令生成打包
 - 先把实验报告和调查问卷放到pa2017/目录下
 - make submit会在pa2017平行的目录中产生STU_ID.tar.bz2压缩包
 - 将压缩包上传cms即可
- 会在课程主页上公布每一个阶段的截止时间
 - 一定要在每个阶段截止时间前提交
 - PA 1、PA 2等大阶段截止前的小阶段，只要求提交能编译的版本，我们不做正确性检查和打分
 - 每个大阶段截止后，我们会检查实现的正确性并进行打分

调查问卷

- 我们准备了三个量表
 - 用于调研在PA实验过程中各阶段的心理状态和工作负荷
 - 量表公布在课程主页，请根据网页上的要求来填写和提交，随实验报告一起打包提交
 - 量表一和二在实验开始前填写一次，以后每一个大阶段结束时填写一次
 - 量表三在每一个大阶段结束时填写一次
 - 后续可能邀请部分同学来参加开放式访谈（请不要理解成“喝茶”）
- 鼓励参加，自愿参加
 - 量表所收集的数据将用于教学科研，为未来PA的调整和设计提供支撑
 - 需要和代码完成情况做对比，因此原始数据收集时需要提供学号
 - 我们不会公布原始数据
 - 在对数据进行任何形式的发表前我们会做匿名处理

谢谢支持！

下面我们开始讲PA

PA要干什么？

- 要创建NEMU（一个简化的x86模拟器）
 - 由C语言编写
 - 以用户软件的形态运行
 - 能够执行通过交叉编译得到的i386指令集程序

PA的基本原理是什么？

- 以一个软件来模拟硬件可行吗？
 - 答案是肯定的，有好多现成的例子



虚拟机



安卓模拟器



NES模拟器

PA的基本原理是什么？

- 以一个软件来模拟硬件可行吗？
 - 答案是肯定的，安装模拟器后的系统栈

模拟器上执行的程序 (OS, Hello World, Super Mario, ...)
模拟器/虚拟机 (NEMU , Vmware, x Emulator, ...)
操作系统 (Windows, GNU/Linux, Mac, ...)
机器硬件 (Lenovo, Dell, Mac, ...)

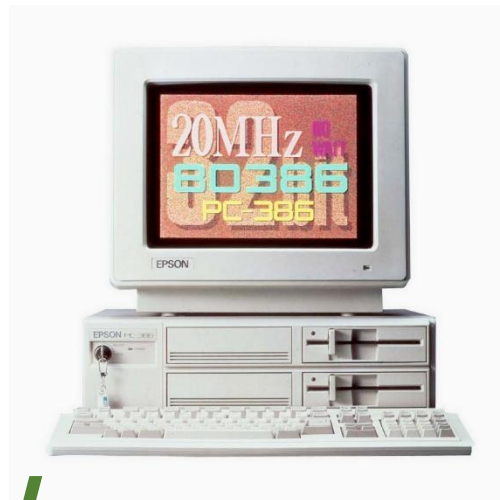
PA的基本原理是什么？

- 从用户和程序员的角度来看，**计算机**就是用来执行程序的
- 我们要模拟的对象就是一个**能够执行程序**的计算机

模拟器就是要把假的搞得跟真的一样



用来骗人的，太复杂☹



用来骗古董程序的，好办☺

PA的基本原理是什么？

- 程序（或者说程序的用户）为什么好骗呢？
 - 打个比方

	计算机	餐厅	
解决计算问题的步骤	程序	菜谱	做菜的步骤
程序处理的对象	数据	食材	做菜加工的对象
执行程序的器件	CPU	大厨	执行菜谱的人

只要能看懂菜谱并按照步骤做出菜肴的人就是好大厨！

~我们会反复使用这个例子~



=

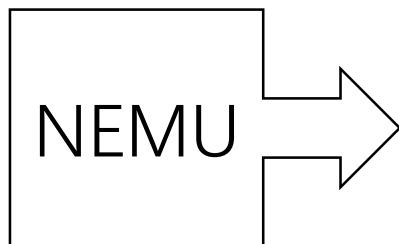


PA的基本原理是什么？

- 程序（或者说程序的用户）为什么好骗呢？
 - 所谓程序，在形式上就是一串指令的序列
 - 硬件设计者和软件开发者约定好有哪些指令可以用（ISA）
 - 只要能够读懂指令并正确完成对应动作（运算、数据操作、输入输出等）的东西就是一台合格的计算机

用C语言模拟机器的功能：

- 变量模拟寄存器
- 变量赋值模拟数据移动指令
- 运算操作模拟运算指令
- ...



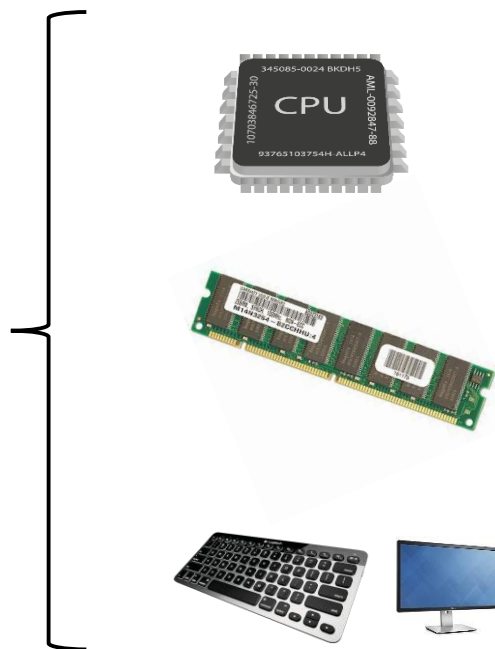
PA的构成

大致对应的器件

PA的四个大阶段



NEMU

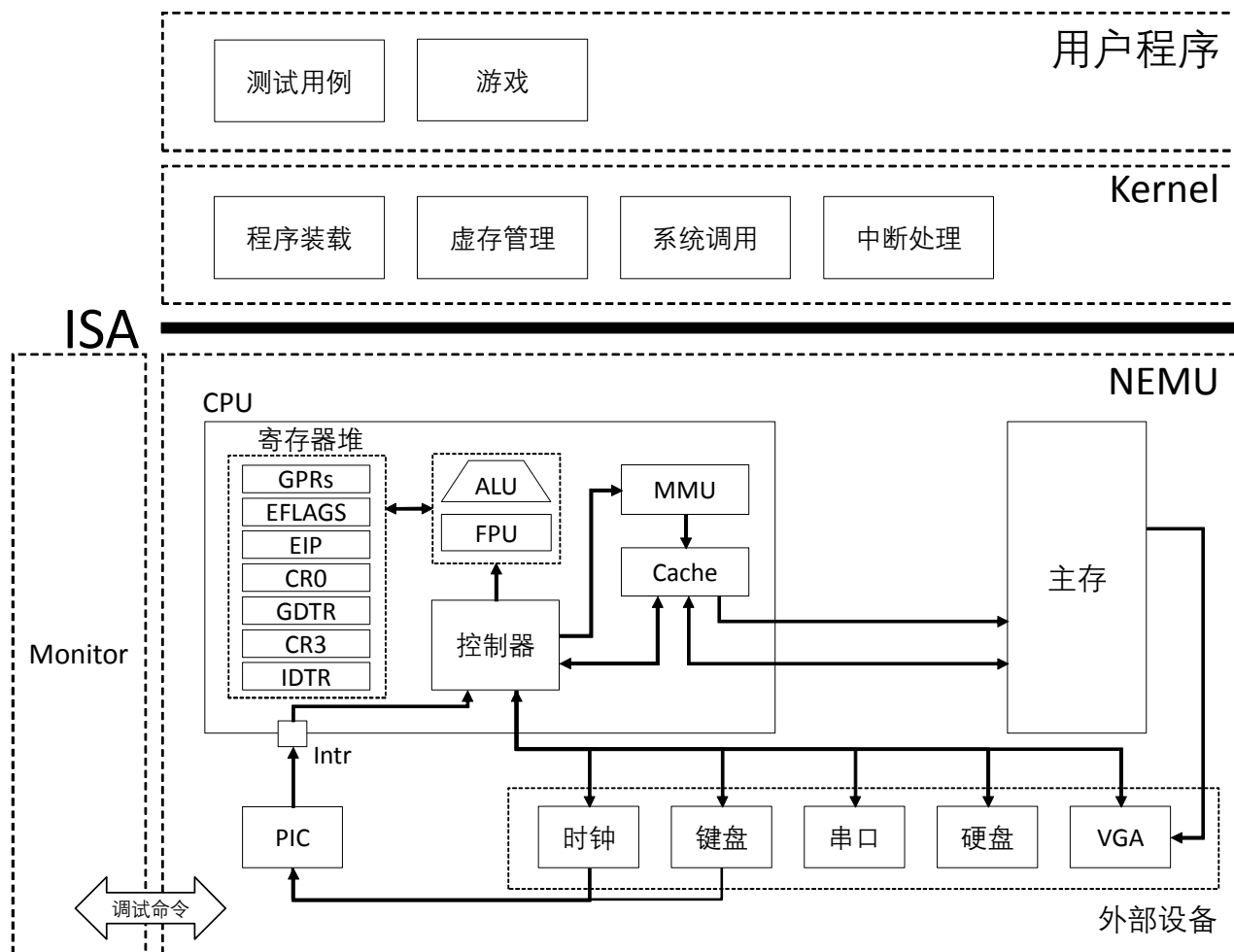


PA 1 数据的表示、存取和运算
PA 2 程序的执行

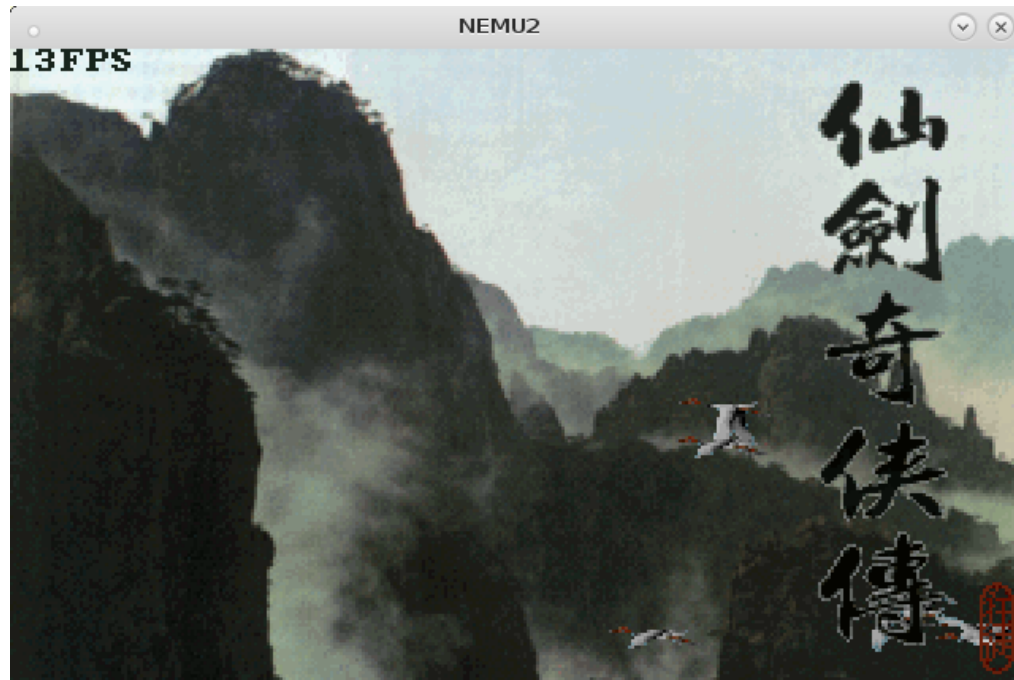
PA 3 存储管理

PA 4 异常、中断与 I/O

PA的构成 – 路线图



PA的成果



当然，最重要的是掌握了计算机系统的重要知识！

下面我们开始讲PA 1