计算机系统综合实践

（面向IA-32的模拟器“NEMU”设计）

天津大学

智能与计算学部

魏继增

提纲

- 什么是NEMU?
- 实验内容
- 实验环境和相关工具
- 实验基本步骤
- 小贴士

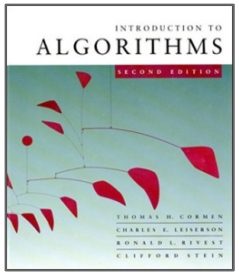
课程简介

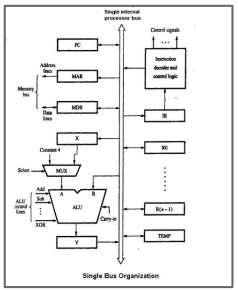
- 本课程是一门集中实践类课程。学生在学习了“计算机系统基础”课程基础之上，通过高级语言构建一台支持**IA32 ISA**的虚拟计算机系统——**NEMU**。学生通过该任务可以将所其涉及的硬件和软件基本概念串联起来，从而实现深入理解计算机系统的全貌和相关软硬件知识体系，理解计算机系统中每一个抽象层次及相互转换关系，建立计算机软硬件协同工作的概念。最终，锻炼学生的计算机系统思维，培养计算机系统能力。

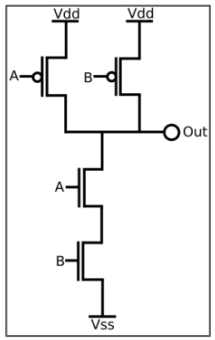
课程目标

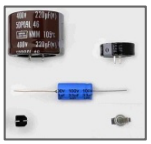
- 通过开发基于**IA32 ISA**的虚拟计算机系统（复杂工程问题），锻炼计算思维能力和系统能力；掌握计算机各抽象层次之间的逻辑关系和转换机理；建立完整软硬件协同工作的概念；能够准确描述计算机系统在运行程序的过程中哪些任务由软件完成，哪些任务由硬件完成，哪些任务需要软硬件协同处理。
- 通过对计算机系统设计问题的分析求解，训练编写复杂程序的能力。利用已有知识和软件工程规范，从分析入手逐步分解复杂工程问题，设计并构建实现方案，编写的程序符合软件工程规范。

计算机系统栈*









Application

Algorithm

Programming Language

Operating System/Virtual Machines

Instruction Set Architecture

Micro-architecture

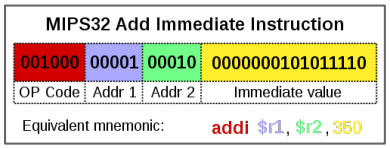
Register-Transfer Level

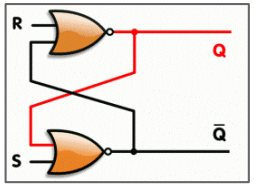
Gates

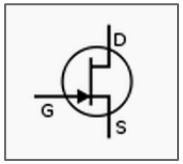
Circuits

Devices

Physics

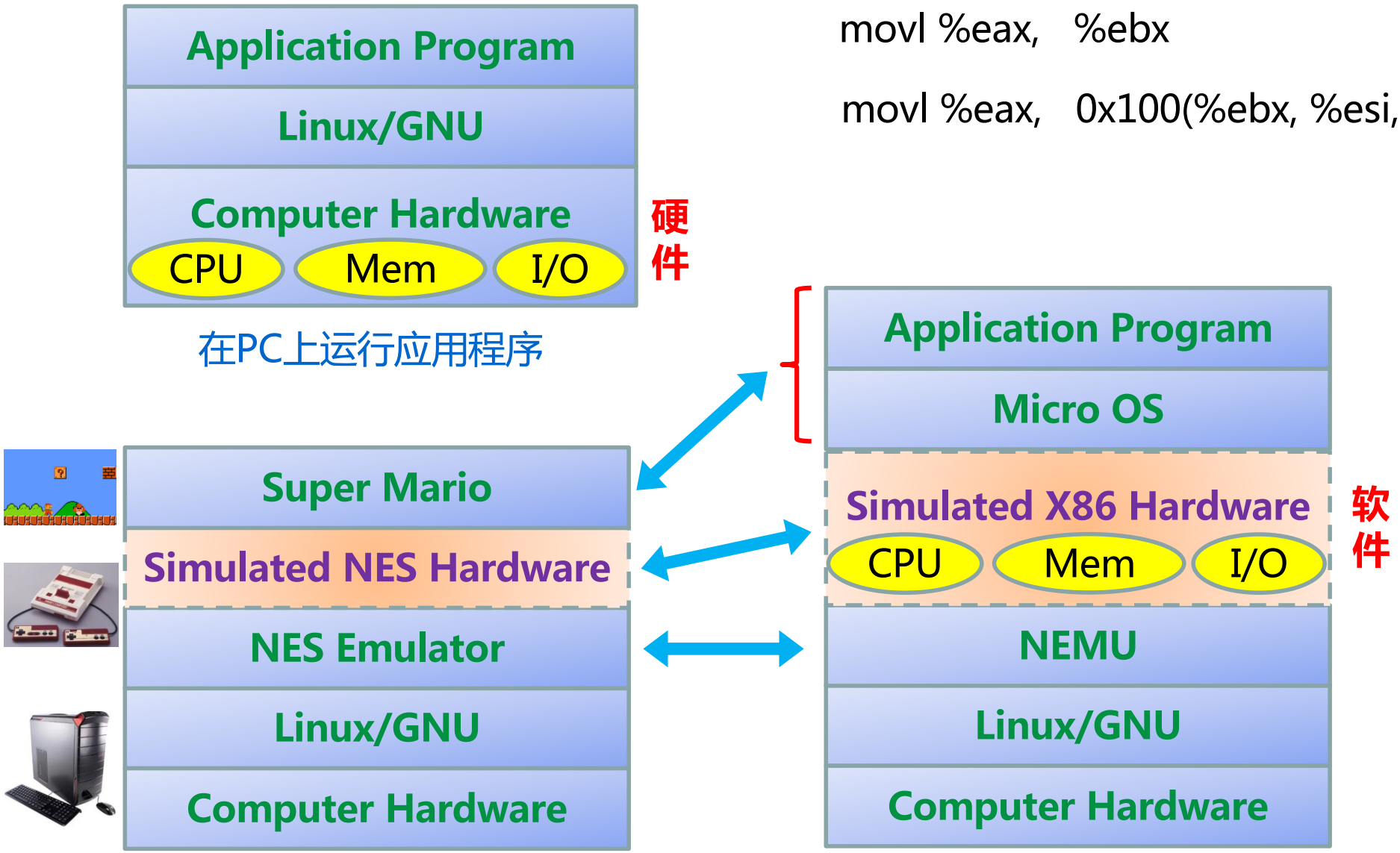






* from Computer Architecture, Princeton University

什么是NEMU？



movl %eax, %ebx

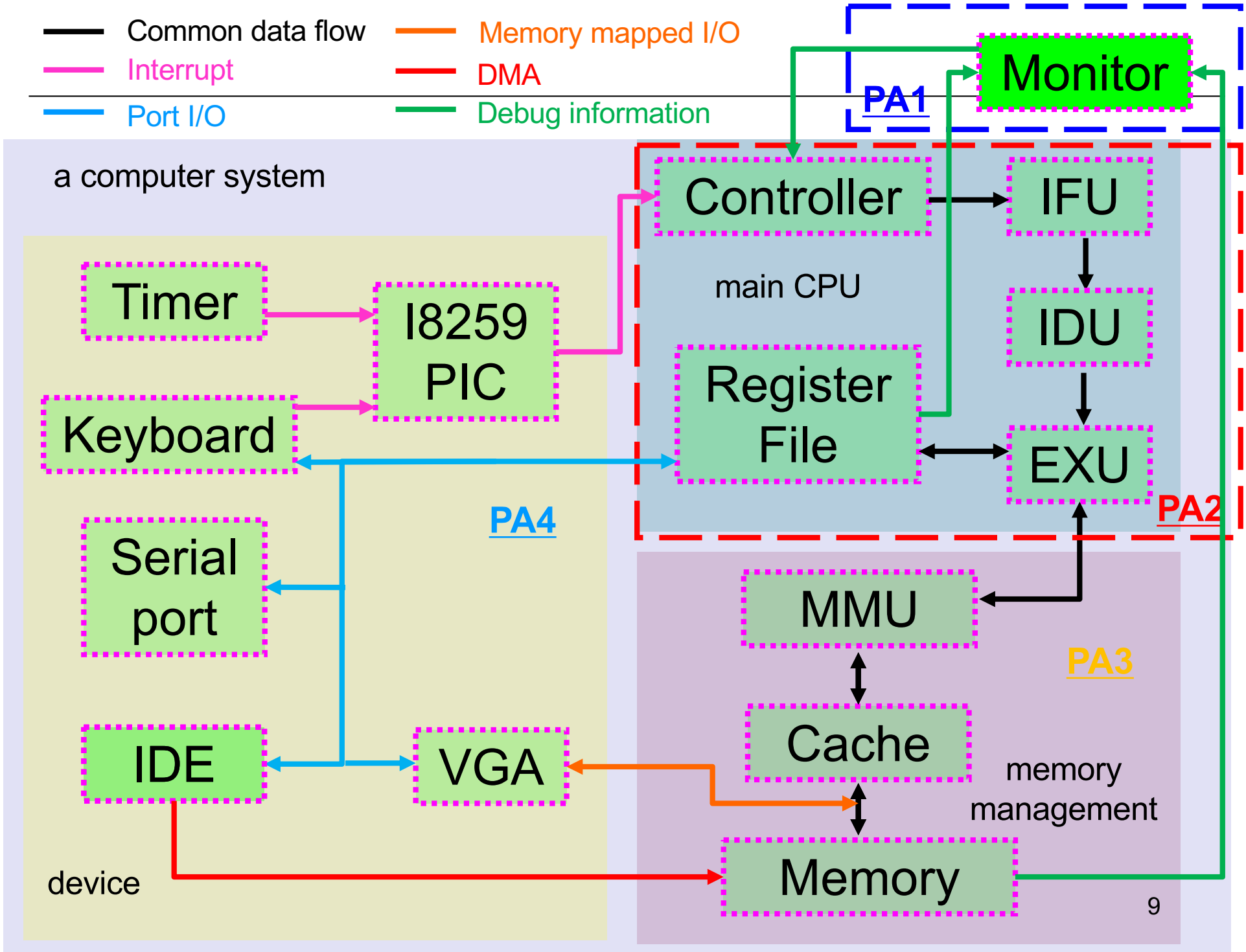
movl %eax, 0x100(%ebx, %esi, 4)

在虚拟机上运行应用程序

什么是NEMU？（cont.）

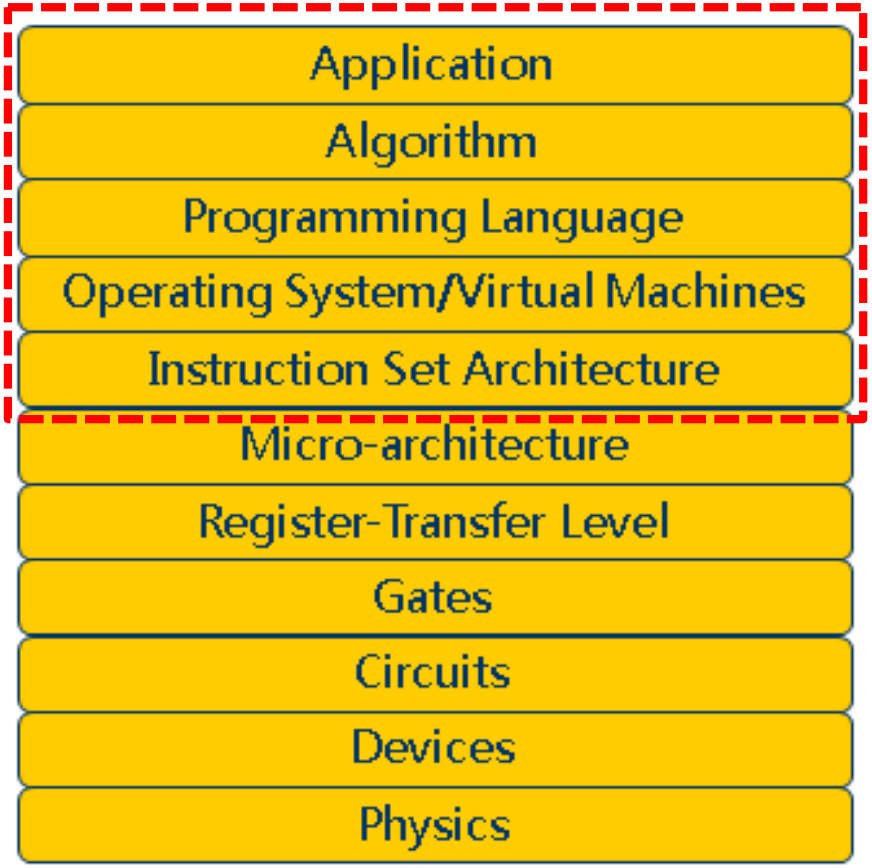
- 理解"程序如何在计算机上运行"的根本途径是实现一个完整的计算机系统

- **NEMU**任务
  - 功能完备(但经过简化)的**IA-32**（**32**位）全系统模拟器
  - 包括**4**个连贯的实验内容
    - 简易调试器(过渡实验)
    - 指令系统
    - 存储管理
    - 中断与**I/O**


Common data flow
Interrupt
Port I/O
Memory mapped I/O
DMA
Debug information
PA1
Monitor
a computer system
Controller
IFU
main CPU
IDU
Timer
I8259
PIC
Register
File
EXU
Keyboard
PA2
PA4
Serial
port
MMU
PA3
Cache
memory
management
IDE
VGA
Memory
device
9

什么是NEMU？（cont.）

- 前导课程
  - 计算机组成原理II
- 与理论课紧密结合
  - 知识点覆盖度广: 约95%
    - 只有动态链接没有涉及
  - 并有部分延伸
  - 覆盖系统栈中ISA以上层次
- 觉得自己上课听懂了?
  - 做一做NEMU实验就知道



NEMU特性

- 简易调试器(位于monitor中)
  - 单步执行, 打印寄存器/内存, 表达式求值, 监视点
- CPU核心
  - 完整的指令周期
  - 支持x86保护模式下的大部分常用指令(不支持实模式)
    - 不支持x87浮点指令
- 存储管理
  - DRAM(包含row buffer和burst的物理特性)
  - 两级联合cache
  - MMU
    - IA-32分段机制, IA-32分页机制(包含TLB)
    - 不支持保护机制

NEMU特性（cont.）

- 中断/异常
  - IA-32中断机制
  - 不支持保护机制
- 设备
  - 时钟, 键盘, VGA, 串口, IDE, I8259 PIC
    - 大部分功能都不可编程
  - 端口I/O, 内存映射I/O

软硬结合的计算机系统

- 实验中后期会结合OS kernel进行
  - 一个单核单任务微型操作系统的内核
    - 2个设备驱动
      - Ramdisk, IDE
    - ELF32加载器
    - 分页存储管理
    - 简易文件系统
      - 文件数量, 大小皆固定, 没有目录
    - 6个系统调用
      - open, read, write, lseek, close, brk

- 软(kernel)硬(NEMU)结合

实验的终极任务

- 在**NEMU**中运行仙剑奇侠传

- 重新审视计算机系统栈

提纲

- 什么是NEMU?
- 实验内容
- 实验环境和相关工具
- 实验基本步骤
- 小贴士

PA1 – 简易调试器

- 简易调试器(类似于GDB)
  - 寄存器结构
  - 单步执行, 打印寄存器/内存, 表达式求值, 监视点
  - 涉及7个必做任务，2个选做任务，3道思考题
  - 预计消耗30小时，约400行代码

PA2 – 指令系统

- 指令系统
  - 实现指令周期，支持保护模式下的常用指令
  - 支持浮点数处理（不是浮点指令）
  - 进一步完善简易调试（打印变量、栈帧链）
  - 实现用户程序的加载
  - 黑客小挑战：运行时代码劫持（选做）
  - 涉及**5**个必做任务，**2**个选做任务，**6**个思考题
  - 预计消耗**60**小时，约**800**行代码

PA3 – 存储管理

- 存储管理
  - Cache
  - IA-32分段机制
  - IA-32分页机制
  - 快表TLB
  - 涉及4个必做任务，3个选做任务，11道思考题
  - 预计消耗45小时，约500行代码

PA4 – 中断与I/O

- 中断与I/O
  - IA-32中断机制
  - 系统调用（运行Hello World程序）
  - 设备与I/O（运行打字小游戏）
  - HAL、文件系统（运行仙剑奇侠传）
  - 预计消耗30小时，约300行代码

本课程仅需要完成PA1 ~ PA3

鼓励兴趣和精力的同学可挑战PA4

考核方式

- **PA1**：简易调试器（**20%**）
- **PA2**：指令系统（**40%**）
- **PA3**：存储管理（**40%**）
- 每次实验中代码实现占**70%**，实验报告占**30%**。
- 对于代码实现，完成所有必做任务起评分**90**分，完成所有必做任务+选做任务起评分**100**分。

提纲

- 什么是NEMU?
- 实验内容
- 实验环境和相关工具
- 小贴士

实验平台

- 智能与计算学部虚拟仿真实验平台（推荐）
  （**http://172.28.45.56**）
  - 完成实验
  - 实验验收
  - 提交实验报告
  - 提交工程源码
  - 答疑
- 自主搭建实验平台
  - 仍需在虚拟仿真实验平台完成提交、验收

实验环境

- 操作系统：**Ubuntu18.04**
- 编译环境：**GNU GCC-4.4.7**
- 环境已在虚拟仿真平台为同学配置完成，可直接进行实验。

其它工具

- 开发工具：**vim + ctags + tmux**
  - **vim**：编辑器之神，强烈推荐大家学习使用
  - **ctags**：代码阅读辅助工具
    - 也可选用**wine + source insight**
  - **tmux**：终端复用器，用于终端切分窗口

- 版本控制工具：**git**
  - 开发过程的跟踪（防抄袭）
  - 工程的版本控制（强烈推荐使用）

参考资料

- 必读资料
  - NEMU实验指导书PA0~PA4
  - i386程序员手册
  - “袁春风，计算机系统基础，机械工业出版社”（第1、2版均可）

- 其它资料
  - 虚拟仿真平台使用说明（附件1）
  - NEMU实验环境配置（附件2）——使用虚仿平台不需要该资料
  - Linux入门教程（附件3）
  - git入门教程（附件4）
  - man入门教程（附件5）
  - i386手册勘误（附件6）
  - Make手册

下载地址

- 实验指导书
  - http://pan.tju.edu.cn:80/link/DEFBF54C42F499FE2685A4201765609D
  - 访问密码：EJzx

- 附件与手册
  - http://pan.tju.edu.cn:80/link/BB9BDE48DF1C21FFB1461F6B732ED265
  - 访问密码：Yzju

框架代码

- github打包下载
  - https://github.com/tjuics2020/NEMU2020
  - git clone下载

代码维护

- 虚拟仿真平台可以上网，定期将代码工程保存到邮箱、网盘等地方

- 建立**github**远程仓库，通过“**git push**”命令定期将代码工程推送到远程仓库保存（推荐）

提纲

- 什么是NEMU?
- 实验内容
- 实验环境和相关工具
- 实验基本步骤
- 小贴士

实验基本步骤

- 请按照“附件1 - 虚拟仿真实验平台使用手册”进入实验平台。

- 参考实验指导书“PA0 – 实验前的准备”的第三节熟悉基本实验步骤。

提纲

- 什么是NEMU?
- 实验内容
- 实验环境和相关工具
- 实验基本步骤
- 小贴士

NEMU实验过程中可能遇到的困难

- 道理我都懂，但真正做的时候很难啊！
- 没错，让任何计算机系统工程落地生根天生就是很难的一件事情！很多计算机大师都觉得难，不是只有你觉得难，这很正常！
- 究竟难在哪里？
  - 技能：陌生的环境/工具/框架（**Linux、Vim、GDB，Git . . .**）
  - 前导课程：代码挂了、程序出了难以理解的错误
  - 系统观：不明白框架代码是什么意思，不明白为什么突然执行到这里了

遇到困难怎么办？ —— NEMU教会你很多

- 通过基本调试原则调整心情
  - 机器永远是对的
  - 未测试代码永远是错的
- 正确使用搜索工具
  - 用百度百科查看词条简介（没办法，**wikipedia**上不去）
  - 用**man**、百度找教程（没办法，**Google**上不去）
  - 用**stackoverflow**、**CSDN**、知乎找解决方案
- 仔细阅读各种手册（**RTFM**），了解系统行为
- 一边读手册，一边读框架代码（切记！切记！切记！）
- 通过编译选项**-Wall**和**-Werror**消除潜在**bug**（已设置好）
- 使用**assert**尽早暴露错误
- 使用**git**尽早提交能跑的代码
- 铭记**KISS**法则，尽早测试，先完成后完美

除了上面这些，我还能做什么

- 人总会犯错误，**bug**是无法避免的
- 锻炼自己的系统观
  - 可以快速定位错误在系统栈中的位置
  - 了解系统如何工作，克服恐惧感
- 实践！实践！实践！
  - 做实验 **->** 踩坑 **->** 调**bug ->**总结经验
  - 做更大的实验/项目 **->** 踩更深的坑 **->** 调更难的**bug ->** 总结更宝贵的经验
  - 本质就是积攒“经验值”和打游戏一个道理**^_^**

你能从NEMU实验中收获什么

- 从本质上理解一个程序在计算机上是如何运行的
- 串联相关软硬件知识，建立系统观
- 不再惧怕大型程序（都是纸老虎！），能看懂、会开发
- 掌握Linux系统下的软件开发、管理方法
- 提高文档阅读能力（特别是英文文档）
- 获得成就感，增强学习信心！

你知道吗？“NEMU”就是一个本科生大四时候独立设计的。相信你也可以！

Welcome to NEMU!