

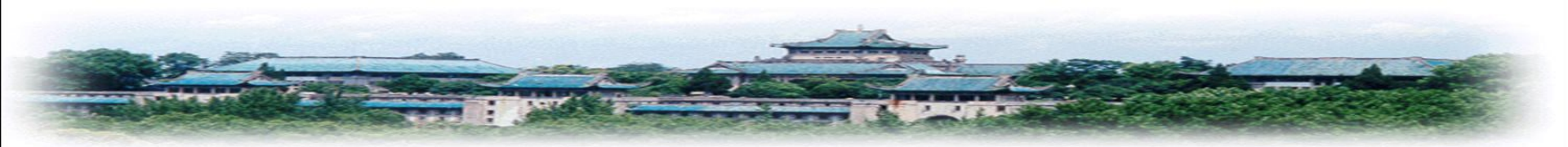
# 操作系统设计及实践

《操作系统原理》配套实验

操作系统课程组

2024年9月

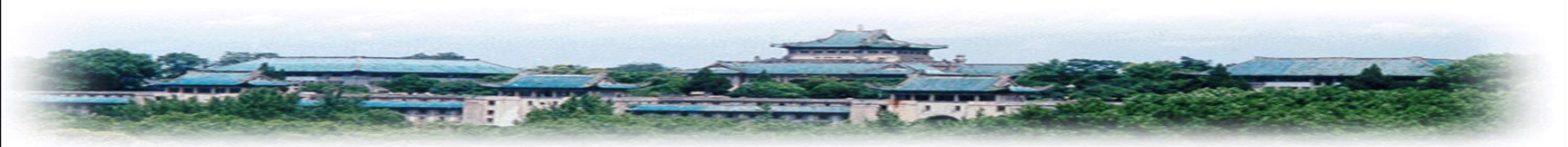




# 操作系统设计实验系列（三）

## 分页机制





# 一、实验目标

- 掌握内存分页机制
- 对应参考阅读章节：第三章3.3节
- 对应参考代码为
  - 基本分页机制/f/pmtest6.asm
  - 读取内存，并合理填充/g/pmtest7.asm
  - 体会分页， /h/pmtest8.asm





## 二、本次实验内容

1. 认真阅读章节资料，掌握什么是分页机制
2. 调试代码，掌握分页机制基本方法与思路
  - 代码3.22中，212行--237行，设置断点调试这几个循环，分析究竟在这里做了什么？
3. 掌握PDE，PTE的计算方法
  - 动手画一画这个映射图
  - 为什么代码3.22里面，PDE初始化添加了一个PageTblBase(Line 212)，而PTE初始化时候没有类似的基地址呢（Line224）？
4. 熟悉如何获取当前系统内存布局的方法
5. 掌握内存地址映射关系的切换
  - 画出流程图
6. 基础题：依据实验的代码，
  - 自定义一个函数，给定一个虚拟地址，能够返回该地址从虚拟地址到物理地址的计算过程，如果该地址不存在，则返回一个错误提示。
  - 完善分页管理功能，补充alloc\_pages, free\_pages两个函数功能，试试你能一次分配的最大空间有多大，如果超出了有什么办法解决呢？
7. 进阶题（选做）
  - 设计一个内存管理器，提示，均按照页为最小单位进行分配、对于空闲空间管理可采用位图法或其他方法进行分配，分配策略不限。





## 三、完成本次实验要思考的问题

1. 分页和分段有何区别？在本次实验中，段页机制是怎么搭配工作的？
2. PDE、PTE，是什么？例程中如何进行初始化？CPU是怎样访问到PDE、PTE，从而计算出物理地址的？
3. 开启分页机制之后，在GDT表中、在PDE、PTE中存的地址是物理地址、线性地址，还是逻辑地址，为什么？
4. 为什么PageTblBase初始值为2M+4K？能不能比这个值小？
5. 怎么读取本机的实际物理内存信息？
6. 如何进行地址映射与切换？
7. 如何实现alloc\_pages, free\_pages



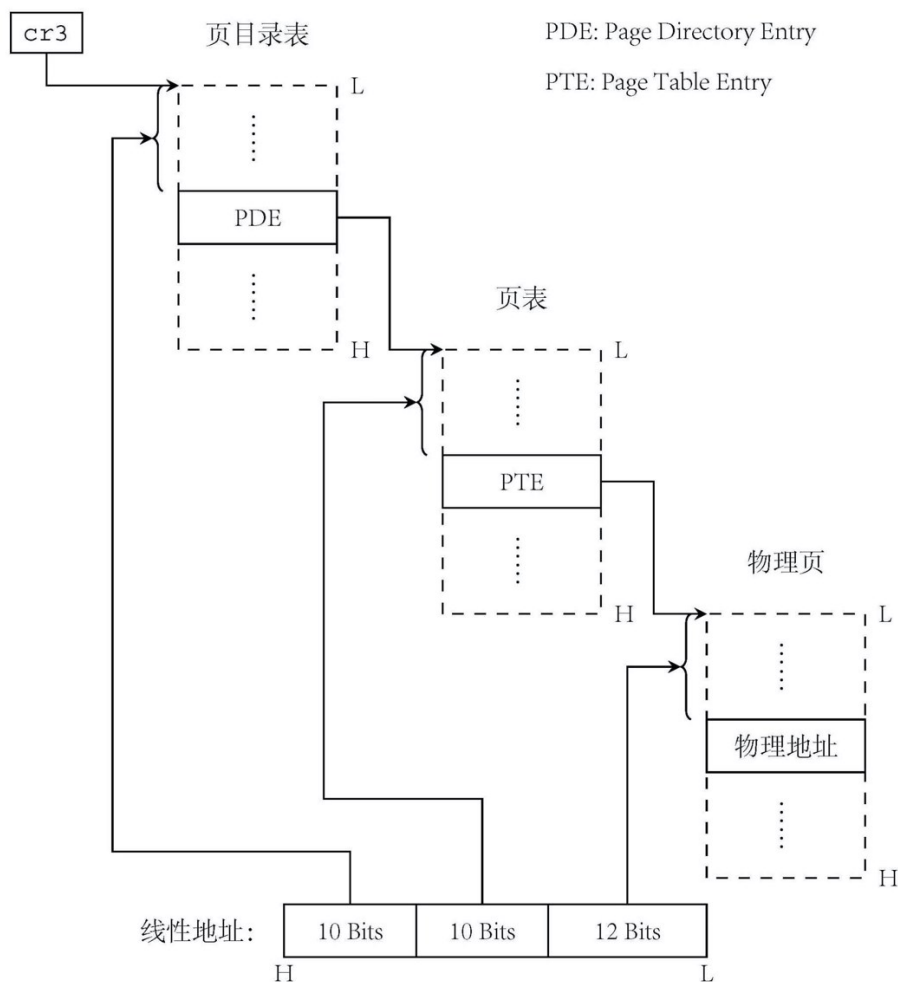


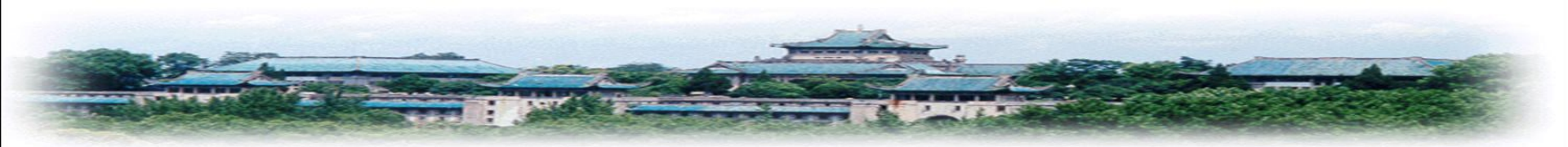
## 四、基本知识点回顾

### • 分页机制

#### 启动分页的步骤

- 初始化页目录表、页表、CR3的内容，使各部分指向正确的位置。
- 修改CR0的PG位，使分页机制打开。





## 四、基本知识点回顾

- 如何取得当前物理内存信息
  - BIOS中断: int 15h
  - 循环读取ARDS结构 (Address Range Descriptor Struct)
  - 注意
    - 这里是在实模式下调用的, Why?
    - 实模式、保护模式下内存变量的访问问题?
- 如何计算物理页的个数
  - 根据ARDS, 计算总的内存大小
  - 内存/页面大小, 向上取整





## 四、基本知识点回顾

- 一点汇编知识:

- Stosd (dword, 4字节)

- 将eax的内容(4字节)存储到es:edi指向的内存单元中, 同时edi的值根据方向标志的值(DF==0)增加或者减少 (DF==1)
    - 相应的还有stosb (1字节), stosw (2字节)

- 如何在bochs里面对.com文件进行调试

- 使用Magic break:
    - 修改bochsrc文件, 在文件末尾添加"magic\_break: enabled=1"
  - 然后在需要添加调试的语句前面增加xchg bx bx, 则执行时候bochs就会停下来。
    - 调试时候n是下一句, 会跳过函数调用, 如果要进去, 需要s (step in) 进入那个函数







谢 谢！



武汉大学