Lab1 要点回顾

2008-3- 4

Lab1 要点

- 熟悉bochs的常用命令
- PC启动过程及内存布局
- boot loader的功能
- 段式寻址方式
- 函数调用时的堆栈结构

熟悉bochs的常用命令

- 设置断点的指令
  - 根据物理地址、虚拟地址
- 单步运行的指令
- 如何查看寄存器信息
- 如何查看CPU状态信息
- 如何查看GDT信息

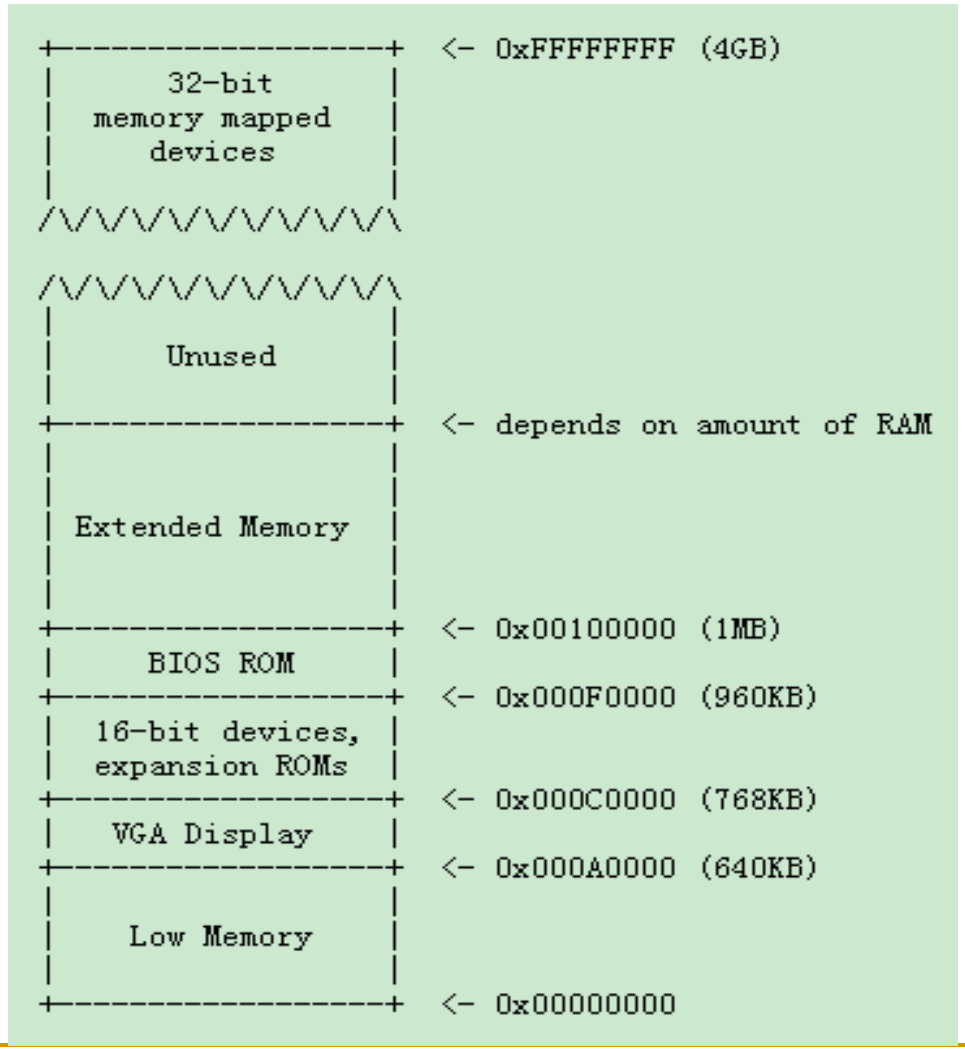
PC启动过程及内存布局

0x100000	4G	空闲
0x0F0000	0x0FFFFF	BIOS
0x0A0000	0xEFFFFF	IO
0x007E00	0x09FFFF	空闲
0x007C00	0x007DFF	启动磁盘MBR
0x000000	0x007BFF	空闲

- PC加电BIOS初始化
  - CS:IP->0xf000:0xfff0
- BIOS读MBR到0x7c00
- BIOS跳转到0x7c00
- MBR加载磁盘上的内核到1M位置
- MBR跳转到内核起始位置

- MBR源文件
  - ./boot/*
- 内核源文件
  - ./kern/*

理解内核在内存中的布局



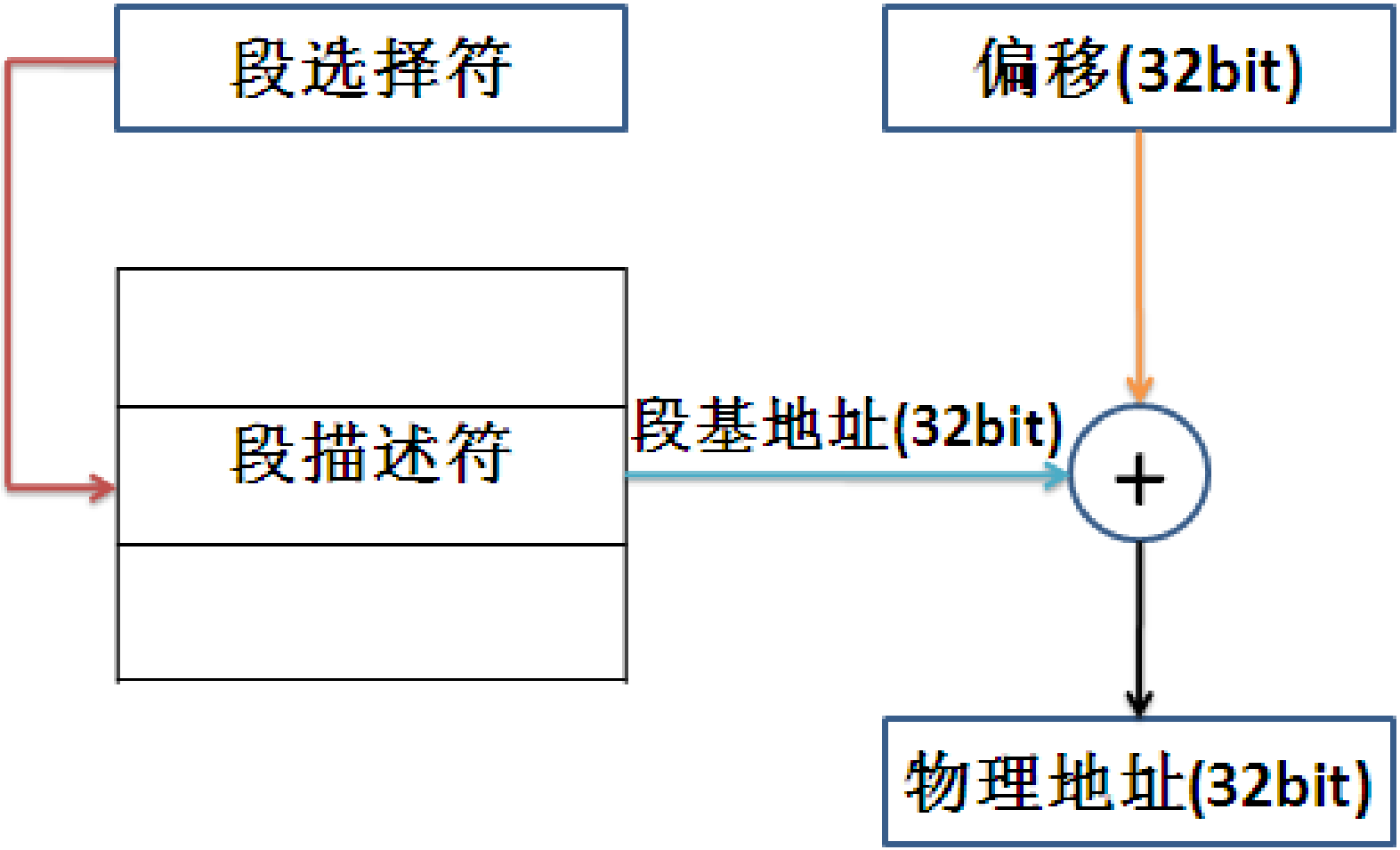
了解boot loader的功能

- 完成系统从实模式到保护模式的转换
- 加载操作系统的kernel并将程序执行转到kernel的开始处

实模式下的段式寻址方式

- 实模式下的逻辑地址由“段基址”和“偏移量”组成
- “段基址”和“偏移量”都是16位，其中，“段基址”由段寄存器CS、DS、SS、ES、FS和GS提供，“偏移量”由BX、BP、SP、SI、DI、IP或者这些寄存器的组合形式来提供
- 实模式下逻辑地址到物理地址的转换公式为：物理地址 ＝ 段基址 × 16 ＋ 偏移量

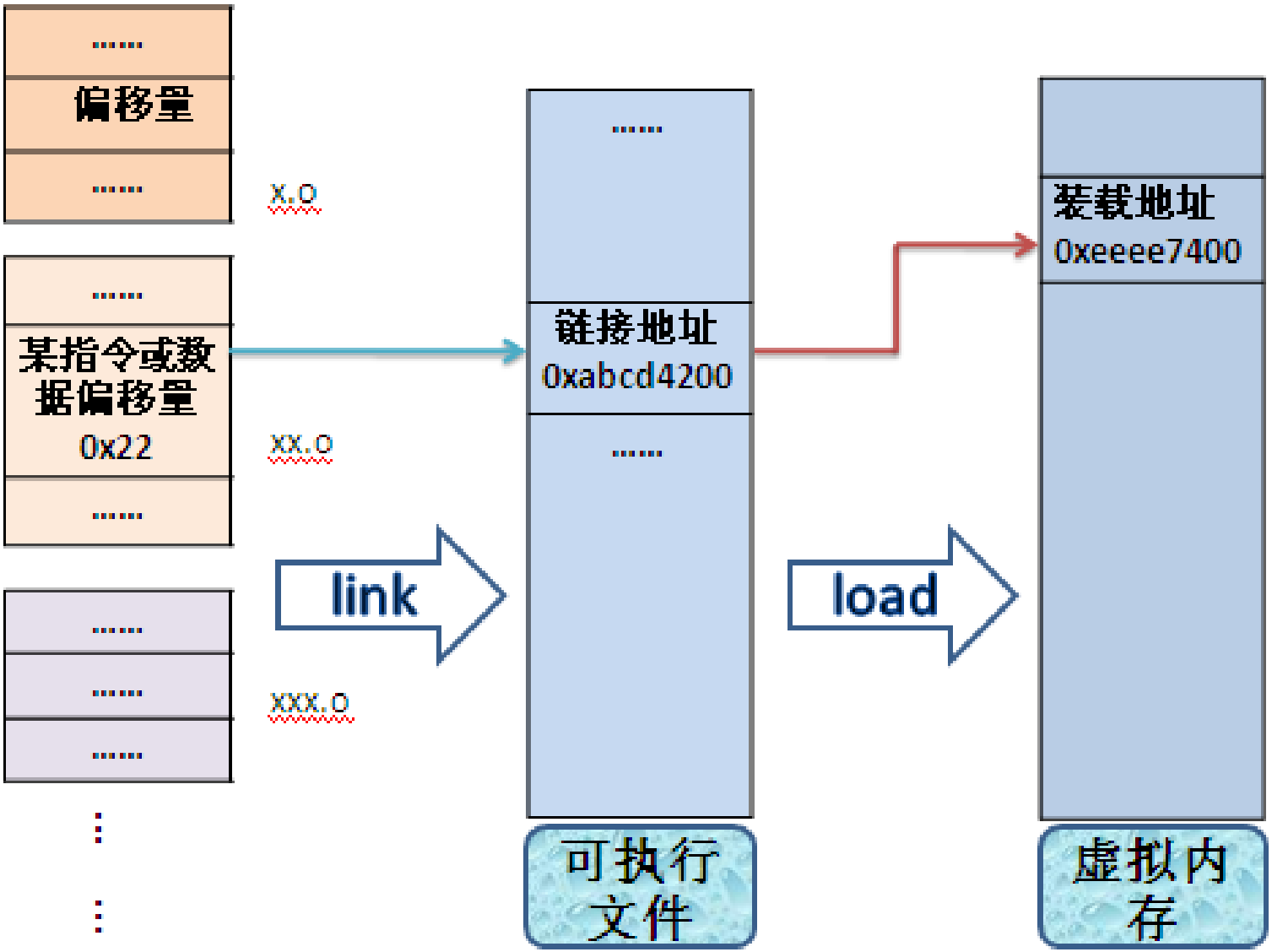
保护模式下的段式寻址方式



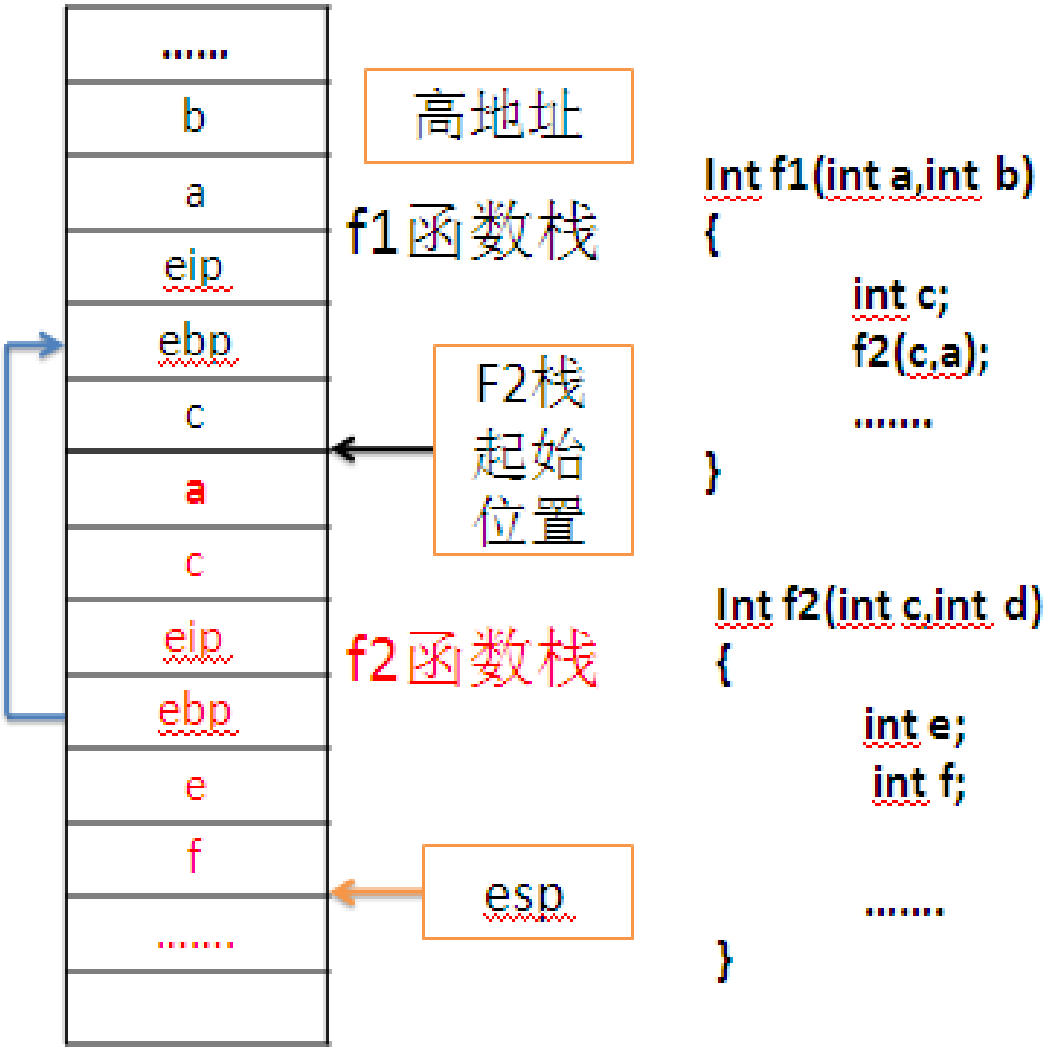
从实模式到保护模式的切换

- 加载GDT
  - lgdt gdtdesc
- 打开保护模式
  - movl %cr0, %eax
  - orl $CR0_PE_ON, %eax
  - movl %eax, %cr0
- 跳转到32位代码开始地址
- ljmp $PROT_MODE_CSEG, $protcseg

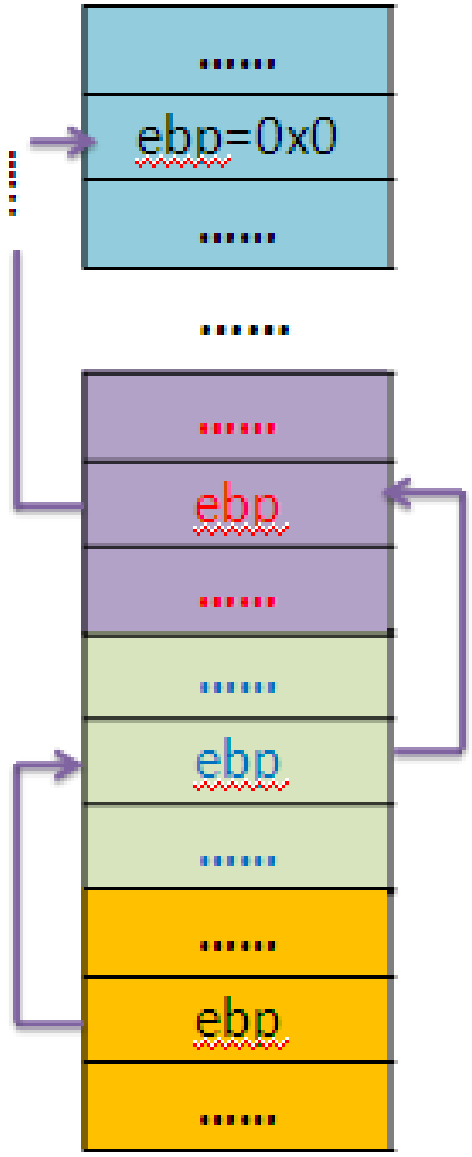
链接地址和装载地址的区别



理解函数调用时的堆栈情况



Back trace函数原理



了解esp和ebp的作用

- esp指向目前栈中已使用空间和空闲空间的分界点，且在x86及很多其他处理器上，esp向下生长
- 被调用函数的ebp装载调用者的ebp的地址，可用于对函数调用关系进行回溯

字符输出格式

Lab中VGA输出一个字符长度为16位
格式如下：

15 底色 12	11 字色 8	7 字符 0

实习命名要求（更正）

- 从课程网站上下载[lab?.tar.gz]后，解压得到代码目录[lab?]
- 将[lab?]重命名为[lab?-组ID]，完成实习
- 将[lab?-组ID]压缩为[lab?-组ID.tar.gz]或[lab?-组ID.tar.bz2]的代码包
- 文档命名为[lab?文档-组ID.doc]
- 代码和文档放入目录[OS实习-lab?-组ID]，将目录压缩成rar或zip提交