

基于 Linux 的操作系统研究

CynYang

2023 年 5 月 14 日

1938

总体设计框架

- ▶ 启动模块
- ▶ 进程模块
- ▶ 内存模块
- ▶ 外围功能模块

开发环境

► 宿主 OS

<i>debian</i>	Linux debian v6.1.0.7-amd64

开发环境

- ▶ 宿主 OS
- ▶ 虚拟机

<i>debian</i> <i>bochs</i>	Linux debian v6.1.0.7-amd64 Bochs v2.6.9

开发环境

- ▶ 宿主 OS
- ▶ 虚拟机
- ▶ 编译器

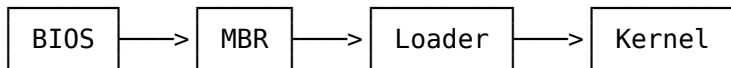
<i>debian</i> <i>bochs</i> nasm gcc	Linux debian v6.1.0.7-amd64 Bochs v2.6.9 Nasm v2.16.01 Gcc v12.2.0

开发环境

- ▶ 宿主 OS
- ▶ 虚拟机
- ▶ 编译器
- ▶ 编辑器

<i>debian</i>	Linux debian v6.1.0.7-amd64
<i>bochs</i>	Bochs v2.6.9
nasm	Nasm v2.16.01
gcc	Gcc v12.2.0
<i>Emacs</i>	GNU Emacs v28.2
<i>Vim</i>	GNU Vim v0.7.2

启动模块



Mbr

- ▶ 0x7C00
- ▶ 512 字节
- ▶ 加载 Loader

实模式下的内存布局

0x100000	JMP F000:E05B	16B	1MB
	System BIOS	64KB-16B	
0xF0000	Expansion Area (maps ROMs for old peripheral cards)	160KB	
0xC8000	Video BIOS	32KB	
0xC0000	Legacy Video Card	128KB	
0xA0000	Ext. System BIOS	1KB	
0x9FC00	Available	~608KB	
0x7E00	MBR	512B	
0x7C00	Available	~30KB	

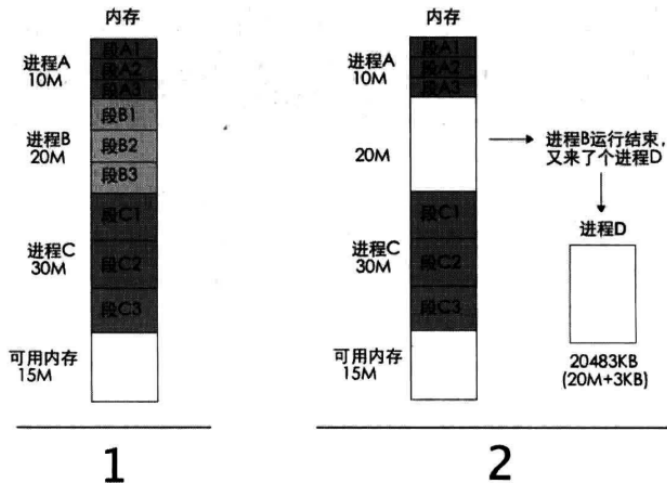
Loader

- ▶ 从实模式到保护模式
- ▶ 打开分页机制
- ▶ 加载 Kernel

从实模式到保护模式

- ▶ 把 0x92 第一个 Bit 置 1
- ▶ 打开 A20 地址线
- ▶ 将 CR0 寄存器的 PE Bit 置 1

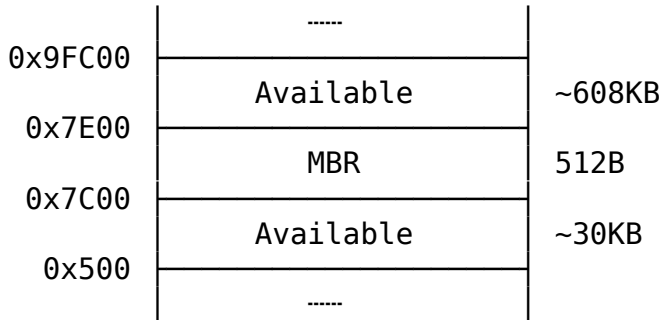
分页机制



打开分页机制

- ▶ 准备好页表目录和页表
- ▶ 将页表地址写入控制寄存器 CR3
- ▶ 将寄存器 CR0 的 PG Bit 置 1

加载 Kernel



进程模块

- ▶ 从特权级 0 到特权级 3
- ▶ 创建用户进程
- ▶ 进程调度

内存模块

- ▶ 初始化内存块描述符
- ▶ 分配/释放内存

外围功能模块

- ▶ 文件系统
- ▶ Shell

文件系统

- ▶ 创建文件
- ▶ 打开/关闭文件
- ▶ 写入/读取文件
- ▶ 删除文件

Shell

- ▶ `Ctrl` + `L` 和 `Ctrl` + `U`
- ▶ `ls`
- ▶ `cd`
- ▶ `mkdir`
- ▶ `rm`