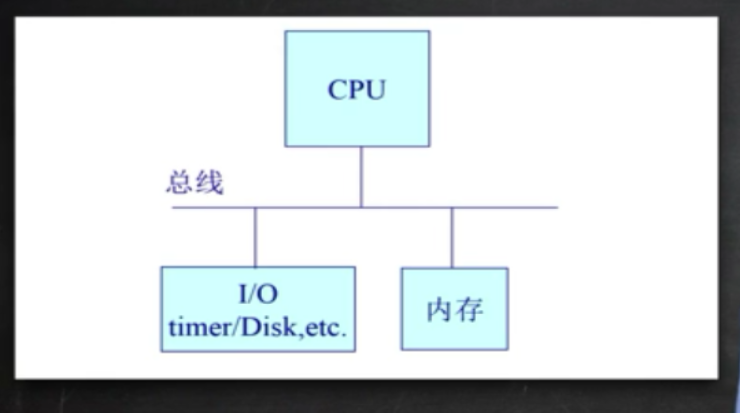
## 启动，中断，异常和系统调用。

操作系统的启动：



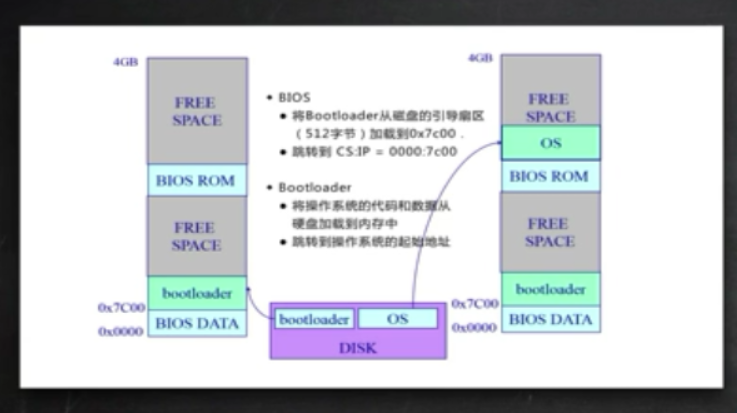
Disk:存放OS

BIOS:基本的IO操作系统

BootLoader：加载os



BIOS 加载 Bootloader ,Bootloader再加载OS



操作系统与设备和应用程序交互：通过系统调用，中断，异常。

系统调用：

应用程序主动向操作系统发出服务请求

异常：

非法指令或者其他坏的处理状态

中断：

来自不同的硬件设备的计时器和网络的中断。

操作系统如何设计和实现中断，异常和系统调用？

源头：

产生时间：

响应方式：

中断和异常处理机制：硬件和软件结合

中断：

硬件处理：

软件处理：

保存当前处理状态

中断服务程序处理

清除中断标记

恢复之前保存的处理状态

异常：异常编号

保存现场

异常处理

杀死产生了异常的程序

重新执行异常指令

恢复现场

系统调用：

程序访问主要是通过高层次的API接口，而不是直接进行系统调用。

OS如何设计并实现系统调用的？

通常情况下，与每个系统调用相关的序号

系统调用接口根据这些序号来维护表的索引

系统调用接口调用内核状态中预期的系统调用

并返回系统调用的状态和其他任何返回值

用户不需要知道系统调用是如何实现的

只需要获取API和了解操作新系统将什么作为返回结果

操作系统接口的细节大部分都隐藏在API中

通过运行程序支持额库来管理

