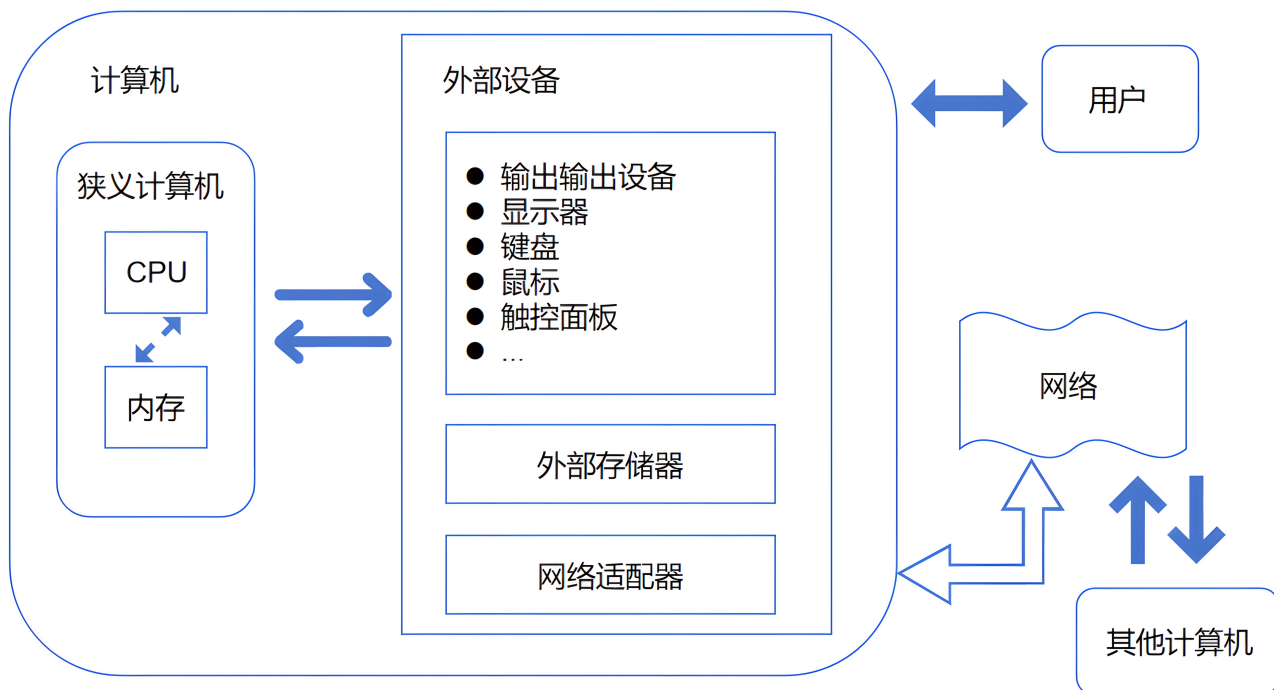


计算机系统硬件



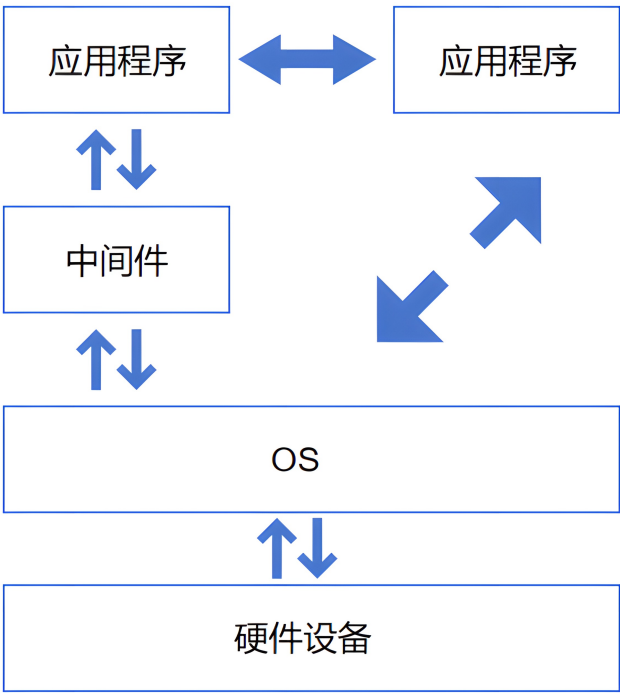
按照上述计算机系统硬件结构，在计算机系统运行时，硬件设备重复执行以下步骤。

- 输入设备或者网络适配器向计算机发起请求
- 读取内存命令，在 CPU 执行，把结果写入负责保存数据的内存区域中
- 将内存中的数据写入 HDD(硬盘驱动器)，SDD(固态硬盘)，或者通过网络发送到其它计算机，或者通过输出设备提供给用户
- 回到第 1 个步骤

如上操作整合而成的有意义的处理即 **<程序>**，程序有：

- 应用程序：用户直接使用的程序
- 中间件：辅助应用程序的程序，数据库系统
- OS: 直接控制硬件设备，为应用程序，中间件提供运行环境的程序，如 Linux

以上三者关系为：



程序在 OS 上以 **<进程>** 为单位进行，一个程序包含一个或多个进程

OS

通过上幅图，可以看出 OS 充当了一个 **<调用外部设备>** 的媒介，如果没有 OS，每个进程将单独处理调用设备(图1)；OS 提供 **<设备驱动程序>** 使得进程通过设备驱动程序访问设备(图2)；进一步地，对于各种不同的设备，Linux 可以通过一个接口处理这些调用(图3)。

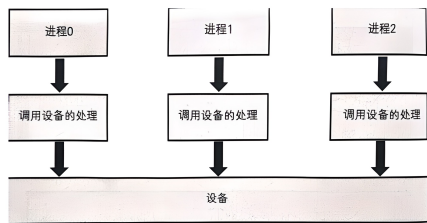


图 1-4 当不存在 OS 时的设备调用

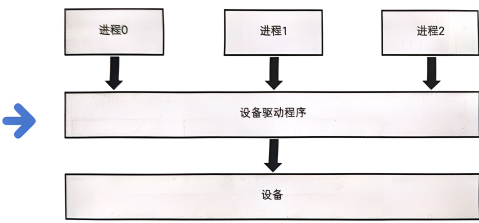


图 1-5 进程通过设备驱动程序访问设备

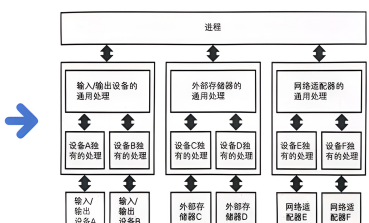
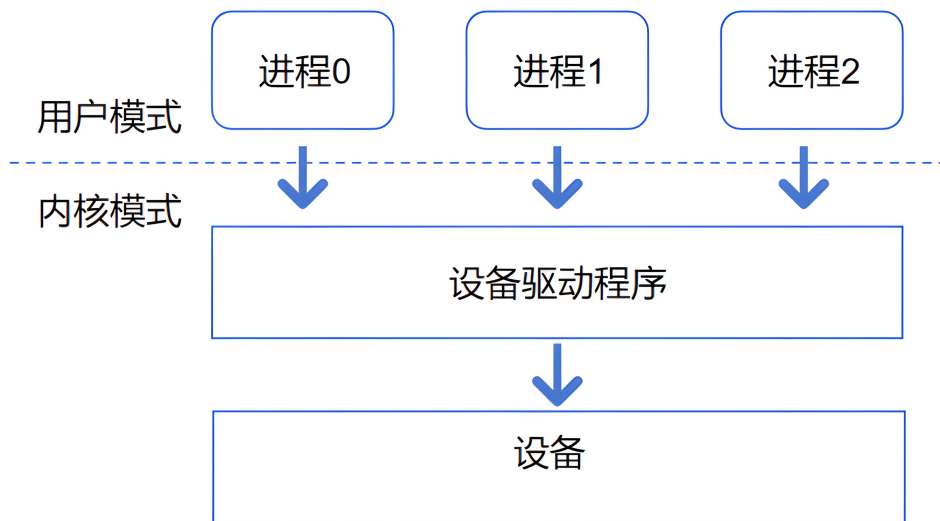


图 1-6 通过同一个接口调用同一类型的设备

这样在 Linux 的管理下，可以避免出现多个进程同时调用相同的设备。

为了进一步保证安全性，Linux 使得 CPU 具有 **<内核模式>** 与 **<用户模式>**，**<只有在内核模式进程才能访问设备>**。



除了设备驱动程序，还有许多不应被普通进程调用的处理程序：

- 进程管理系统
- 进程调度器
- 内存管理系统

这些在内核模式下运行的 OS 的核心处理整合在一起的程序就叫做 <内核>

进程如果需要调用内核提供的功能，需要 <系统调用> 的特殊处理来向内核发出请求

绪论

接下来，参考书将分别以以下几个方面介绍 Linux:

- Linux 在用户模式下运行的功能 and 作为进程与内核通信接口的系统调用
- 创建与终止进程的 <进程管理系统>
- 负责 CPU 资源的 <进程调度器>
- 负责管理内存的 <内存管理系统>
- <文件系统>

普通进程虽然可以通过设备驱动器访问外部存储器的程序，但为了简化这一过程，会利用 <文件系统> 程序(该程序也属于内核)进行访问。

两种方式可以访问外存，也说明了外存的重要性：

- 启动系统时，计算机首先通过 BIOS(基本输入输出系统) 或者 UEFI(统一可扩展固件接口) 这种固化于硬件上的软件来初始化设备
- 然后运行引导程序选择需要启动的 OS
- 从外部存储器读取 OS
- 关闭电源时将运行期间内存上创建的数据写入外部存储器