

## 一、程序分类

---

程序按其运行环境分为：

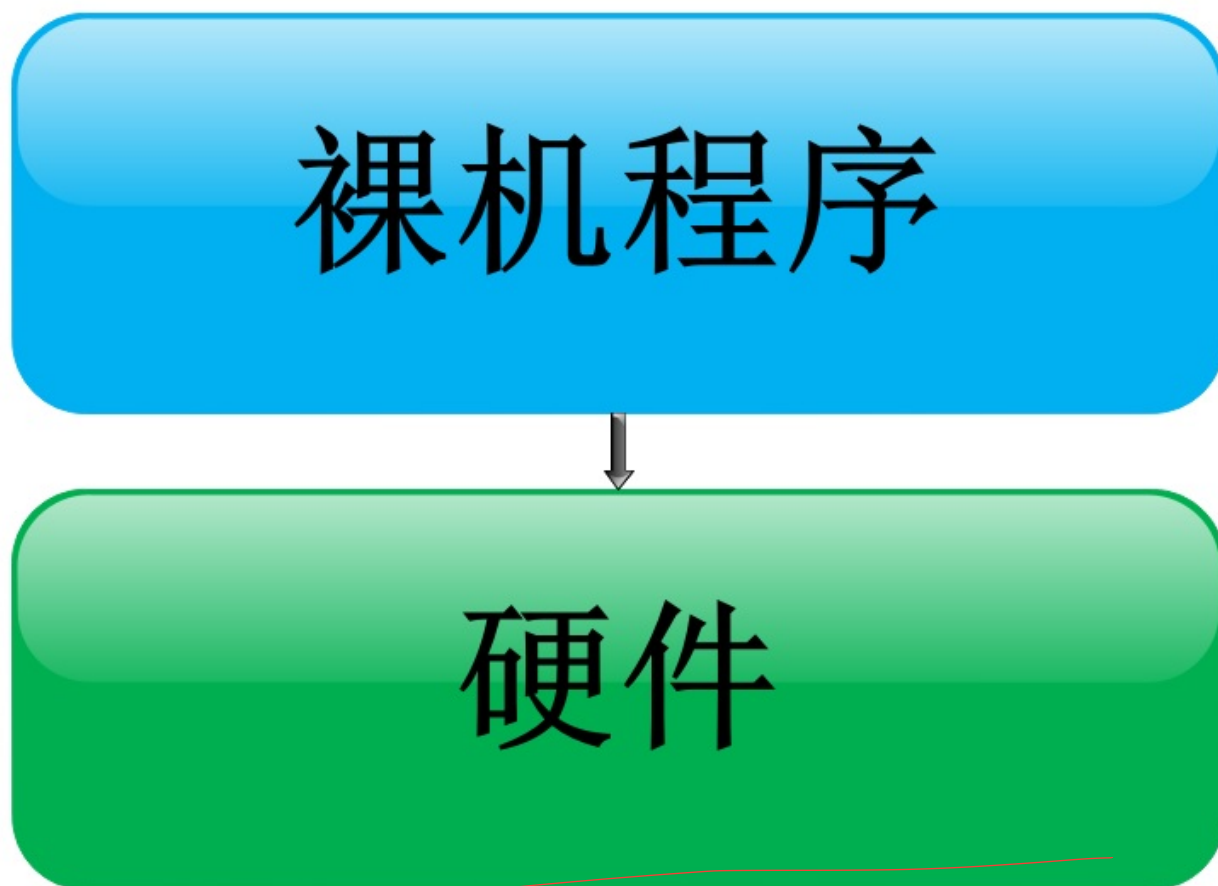
1. 裸机程序：直接运行在对应硬件上的程序
2. 应用程序：只能运行在对应操作系统上的程序

## 二、计算机系统的层次结构

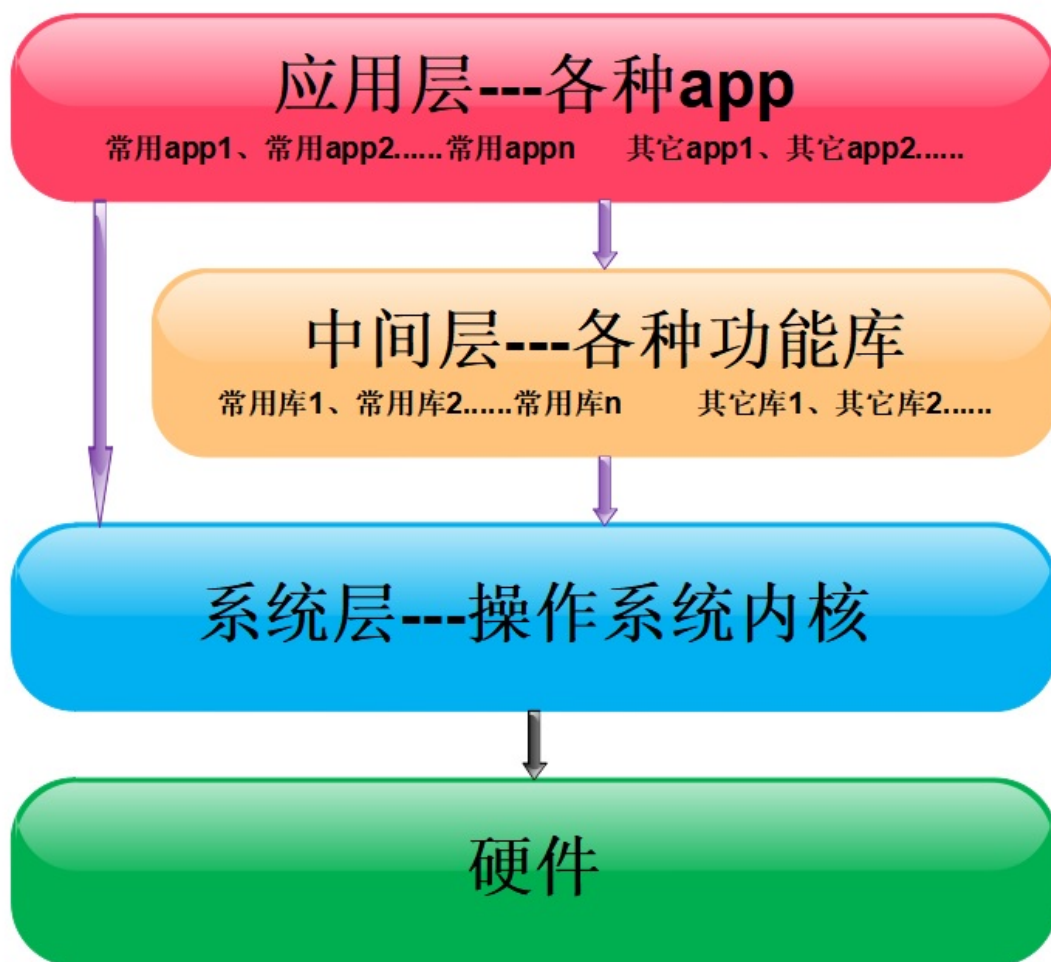
---

计算机系统两种层次结构：

### 2.1 无操作系统的简单的两层结构



### 2.2 有操作系统的复杂的四层结构



### 三、什么是操作系统

狭义的操作系统：给应用程序提供运行环境的裸机程序，也被称为操作系统内核

广义的操作系统：一组软件集合，它包含：

1. 最核心的一个裸机程序 -----内核 (kernel)
2. app开发常用的一些功能库 (如：C语言标准函数库、线程库、C++标准类库、QT类库等等)
3. 一些管理用的特殊app (如桌面、命令行、app包管理器、资源管理器、系统设置、一些常用后台服务程序)

### 四、操作系统内核的实现模式

内核：操作系统最核心的那个裸机程序，主要负责硬件资源的驱动和管理。

一个操作系统内核主要包括如下几个子模块：

1. 任务管理：多任务支持、任务调度、任务间通讯
2. 内存管理：物理内存管理，虚拟内存实现
3. 设备驱动：各种外部设备的I/O支持
4. 网络协议支持
5. 文件系统支持

## 6. 启动管理

两种典型的内核实现模式：

1. 单内核（宏内核）：所有子模块代码编译到一个比较大的可执行文件（镜像文件）中，各子模块代码共用同一套运行资源，各模块间的交互直接通过函数调用来进行
2. 微内核：只将任务管理、内存管理、启动管理最基本的三个子模块编译到一个微型的可执行文件中，其它子模块则各自编译成独立的后台服务程序，这些服务程序与微型内核以及app间主要通过各种IPC进行通讯

单内核特点：效率高，稳定性低，扩展性差，安全性高，典型操作系统：UNIX系列、Linux

微内核特点：效率低，稳定性高，扩展性高，安全性低，典型操作系统：Windows，QNX

# 五、什么是设备驱动程序

英文：Device Driver

简称：驱动（Driver）

一种添加到操作系统中的特殊程序，主要作用是协助操作系统完成应用程序与对应硬件设备之间数据传送的功能

简言之，设备驱动程序就是操作系统中“驱动”对应硬件设备使之能正常工作的代码。

一个驱动程序主要完成如下工作：

1. 初始化设备，让设备做好开始工作的准备
2. 读数据：将设备产生的数据传递给上层应用程序
3. 写数据：将上层应用程序交付过来的数据传递给设备
4. 获取设备信息：协助上层应用程序获取设备的属性、状态信息
5. 设置设备信息：让上层应用程序可以决定设备的一些工作属性、模式
6. 其它相关操作：如休眠、唤醒、关闭设备等

其中最核心的工作就是设备数据的输入和输出，因此计算机外部设备（外设）也被称为IO设备