

删除和调度等工作。不同的嵌入式系统对任务管理的要求也不相同,因此,需要根据具体的需求选择任务管理方式。

4.3.1 多道程序技术

嵌入式操作系统可以分为两种类型:单道程序设计和多道程序设计。所谓单道程序设计,就是在操作系统当中,在任何时候只能有一个程序在运行。所谓多道程序设计,就是在操作系统当中,允许多个程序同时存在并运行。在现代操作系统当中,为了提高系统资源的利用率,普遍采用多道程序技术。

图 4-13 是单道和多道程序设计的例子。甲、乙两个程序,它们在运行过程中都要用到 CPU 和输入/输出设备。如图 4-13 所示,我们用不同的方框来表示这两个程序对两种资源的使用情况,方框的长度表示使用的时间。

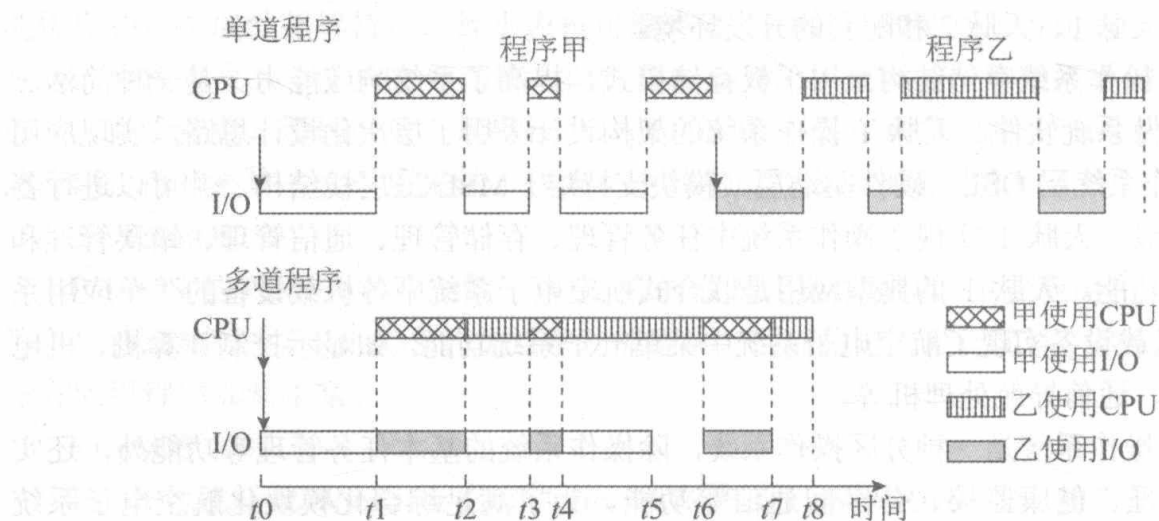


图 4-13 单道与多道程序技术

在单道程序设计的环境下,在任何时候,系统中只能有一个程序在运行,因此,这两个程序的执行只能顺序运行:首先执行程序甲,从 t_0 时刻开始,到 t_6 时刻结束。然后再执行程序乙,从 t_6 时刻开始,一直到它所有的工作都已完成。

在多道程序设计的环境下,允许多个程序同时运行,当一个程序在访问 I/O 设备时,会主动把 CPU 交出来,交由另一个程序去运行,这样就提高了系统资源的使用效率。具体来说,首先,在 t_0 时刻,甲和乙都打算运行,它们的第一件事情都是进行 I/O 操作,由于资源有限,只能满足一个程序的请求。假设甲的请求得到满足,它先开始执行,从 t_0 到 t_1 ,甲一直在使用 I/O 设备,在此期间,乙一直处于等待状态。在到达了 t_1 时刻后,甲已经执行完了 I/O 操作,下一步要执行一小段 CPU 操作。这样,它就把刚刚占用的 I/O 设备释放出来,交给程序乙去使用。因此,在 t_1 到 t_2 期间,程序甲在使用 CPU,程序乙在使用 I/O 设备,它们互不影响。在到达 t_2 时刻后,甲又要执行一小段 I/O 操作,而乙恰巧要执行一小段 CPU 操作,因此,在 t_2 到 t_3 期间,它们相互交换了资源,继续执行。