在现行使用的操作系统中，如Window 95及之后的Windows版本，使用的就是“[抢占式](http://baike.baidu.com/view/1448318.htm" \t "http://baike.baidu.com/_blank)多任务”了，它的总控制权在操作系统手中，操作系统会轮流询问每一个任务是否需要使用 CPU ，需要使用的话就让它用，不过在一定时间后，操作系统会剥夺当前任务的 CPU 使用权，把它排在询问队列的最后，再去询问下一个任务……。

具有最高优先级的线程就是当前正在运行的那个线程。驻留在CPU内正在运行的线程会在什么时候中止呢？一是属于它的[时间片](http://baike.baidu.com/view/359518.htm" \t "http://baike.baidu.com/_blank)已经到期，二是加入了另一个优先级更高的线程。

通过不断从一个线程到另一个线程的切换，给人的感觉是线程的执行是同时进行的。线程并非随时需要运行，经常发生的情况是某个线程需要等待用户的输入，另一个线程的信息或等待I/O请求，否则便无法执行下去。有些线程也许本来就处于[挂起状态](http://baike.baidu.com/view/1268241.htm" \t "http://baike.baidu.com/_blank)。

----在抢占式多任务系统中，操作系统必须具有从任何一个运行的程序上取走控制权和使另一个程序获得控制权的能力。一个**抢占式系统**下运行的应用程序不用担心独占系统，因为系统会合理地给每个运行的任务分配时间块。将这种系统称做抢占式系统是因为它不管在所给时间块内会发生什么事，都抢占运行程序的控制权。OS/2和UNIX都是抢占式系统。

抢占式多任务处理（preemptive multitasking）是将同时进行的各项任务（[task](http://baike.baidu.com/view/1912341.htm" \t "http://baike.baidu.com/_blank)），依照重要程度来排定优先顺序。在抢占式多任务系统中，操作系统（[operating system](http://baike.baidu.com/view/171639.htm" \t "http://baike.baidu.com/_blank)）必须具有从任何一个运行的程序上取走控制权和使另一个程序获得控制权的能力。

一个抢占式系统下运行的应用程序不用担心独占系统，因为系统会合理地给每个运行的任务分配时间块。将这种系统称做抢占式系统是因为它不管在所给时间块内会发生什么事，都抢占运行程序的控制权。OS/2和UNIX都是抢占式系统。