

执行是交错的。进程 A 运行了一会儿，然后是进程 B 开始运行到完成。然后，进程 C 运行了一会儿，进程 A 接着运行直到完成。最后，进程 C 可以运行到结束了。

图 8-12 的关键点在于进程是轮流使用处理器的。每个进程执行它的流的一部分，然后被抢占(preempted)(暂时挂起)，然后轮到其他进程。对于一个运行在这些进程之一的上下文中的程序，它看上去就像是在独占地使用处理器。唯一的反面例证是，如果我们精确地测量每条指令使用的时间，会发现在程序中一些指令的执行之间，CPU 好像会周期性地停顿。然而，每次处理器停顿，它随后会继续执行我们的程序，并不改变程序内存位置或寄存器的内容。


8.2.2 并发流

计算机系统中逻辑流有许多不同的形式。异常处理程序、进程、信号处理程序、线程和 Java 进程都是逻辑流的例子。

一个逻辑流的执行在时间上与另一个流重叠，称为并发流(concurrent flow)，这两个流被称为并发地运行。更准确地说，流 X 和 Y 互相并发，当且仅当 X 在 Y 开始之后和 Y 结束之前开始，或者 Y 在 X 开始之后和 X 结束之前开始。例如，图 8-12 中，进程 A 和 B 并发地运行，A 和 C 也一样。另一方面，B 和 C 没有并发地运行，因为 B 的最后一条指令在 C 的第一条指令之前执行。

多个流并发地执行的一般现象被称为并发(concurrency)。一个进程和其他进程轮流运行的概念称为多任务(multitasking)。一个进程执行它的控制流的一部分的每一时间段叫做时间片(time slice)。因此，多任务也叫做时间分片(time slicing)。例如，图 8-12 中，进程 A 的流由两个时间片组成。

注意，并发流的思想与流运行的处理器核数或者计算机数无关。如果两个流在时间上重叠，那么它们就是并发的，即使它们是运行在同一个处理器上。不过，有时我们会发现确认并行流是很有帮助的，它是并发流的一个真子集。如果两个流并发地运行在不同的处理器核或者计算机上，那么我们称它们为并行流(parallel flow)，它们并行地运行(running in parallel)，且并行地执行(parallel execution)。

 **练习题 8.1** 考虑三个具有下述起始和结束时间的进程：

进程	起始时间	结束时间
A	0	2
B	1	4
C	3	5

对于每对进程，指出它们是否是并发地运行：

进程对	并发的?
AB	
AC	
BC	

8.2.3 私有地址空间

进程也为每个程序提供一种假象，好像它独占地使用系统地址空间。在一台 n 位地址的机器上，地址空间是 2^n 个可能地址的集合， $0, 1, \dots, 2^n - 1$ 。进程为每个程序提供它自己的私有地址空间。一般而言，和这个空间中某个地址相关联的那个内存字节是不能被