

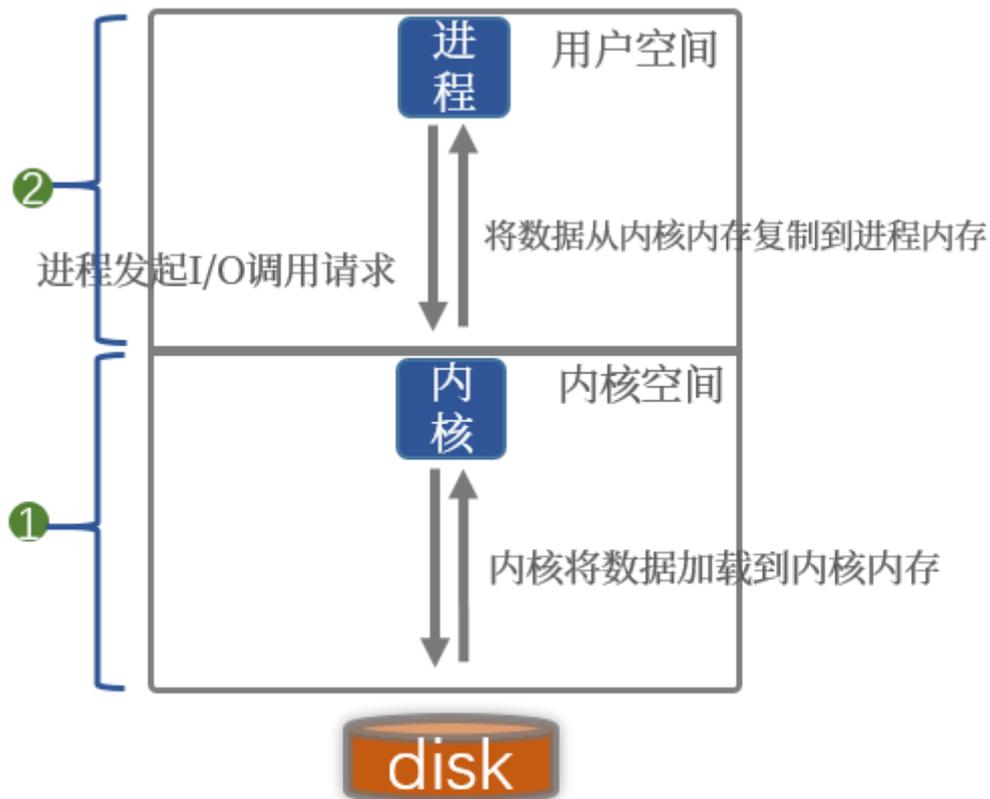
Linux I/O模型

Linux 中的 I/O 模型有五种：`阻塞式(blocking)IO`、`非阻塞式(nonblocking)IO`、`多路复用式(multiplexing)IO`、`事件驱动(signal driven)IO`、`异步(asynchronous)IO`。要理解这几种模型，以及web服务器使用不同I/O模型的区别，必须理解下面相关的概念及术语。下面，将对相关概念及术语进行介绍。

相关概念

进程在内核发起系统调用后，一个完整请求I/O操作包括：

1. 数据从磁盘到内核内存
2. 数据从内核内存到进程内存



同步/异步

同步：进程发起一个调用后，再没得到结果之前，被调用者不会给调用者返回结果。调用者要一直等待被调用者返回最终结果，才能执行后续的操作。

异步：进程发起一个调用后，即便调用者不能立即得到结果，该被调用者也会返回信息，但返回的是未完成状态信息，当调用完成后，内核会自行通知调用者取结果或者由调用者通过轮询的方式查看被调用者是否准备就绪。

同步和异步是站在被调用者如何响应调用者请求的角度理解。异步是被调用者立即返回信息，而同步则是调用者发起调用后，被挂起。

阻塞/非阻塞

阻塞：进程发起调用后，在I/O操作未完成之前，调用者进程进程会被挂起，进入阻塞态，调用者在得到返回结果之后才能继续运行。

非阻塞：进程发起调用后，[被调用函数]完成之前进程不会阻塞，而是立即返回结果。非阻塞但需要忙等待，查看调用是否完成。

调用者等待被调用者返回调用结果时的状态，及调用者在等待返回结果时的行为状态。

五种I/O模型

阻塞式(blocking)IO

进程发起调用之后，被阻塞挂起，不能再处理其他事情。

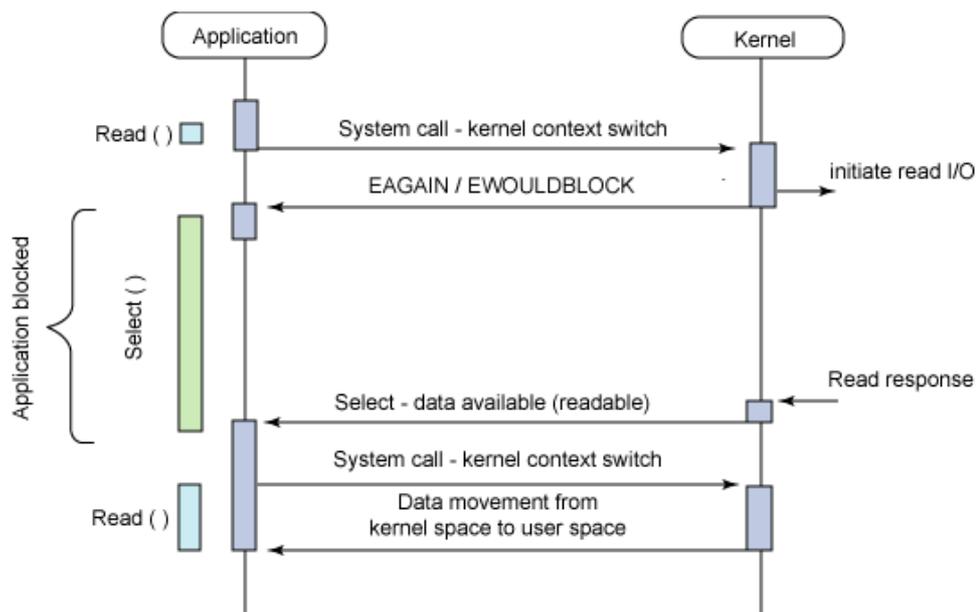
非阻塞式(nonblocking)IO

进程发起调用之后，被调用者立即返回信息，但没有返回最终结果，尽管此时发起调用的进程可以处理其他事情，但需要忙等待，即进程需要不断的询问被调用者是否准备好，可以返回最终结果，所以效率事实上也并不高。

多路复用式(multiplexing)IO

实现了单个进程处理多个IO请求，单个线程通过监控追踪每一个I/O请求的状态，单个进程监视多个文件描述符。所谓IO复用即多个I/O复用一個进程。

Linux内核支持IO多路复用的系统调用有 `select`、`poll`、`epoll`。



上图是`select()`系统调用的过程，`select()`系统调用监视的文件描述符最多是1024个，`poll()`和`select()`在流程上几乎一样，但`poll()`对监视的文件描述符个数没有限制。Apache的`prefork`和`worker`响应模型就使用的Linux内核的`select()`和`poll()`系统调用。

`epoll`在Linux-2.6以上内核支持，`epoll`的特点是：没有文件描述符的限制；采用回调机制；使用`mmap()`文件映射内存来加速与内核空间的消息传递；IO效率不随文件描述符数目增加而线性下降。

事件驱动(signal driven)IO

进程发起调用之后，被调用者立即返回调用者信息，告知接收到请求，之后调用者能够去处理其他事情，等调用者加载完相关数据之后，会通知调用者。此种模型不需要调用者忙等待，当数据从磁盘加载到内核内存后，进程复制数据到进程空间的过程依然是阻塞状态(第二个阶段依然是阻塞状态)。此处进程接收到被调用的信息后，没有被阻塞挂起，所以进程还可以接收其他请求。

Apache的`event`响应模型以及Nginx，都采用的事件驱动的IO模型。`epoll()`系统调用采用的通知机制有：`水平触发`：多次通知、`边缘触发`：一次通知。

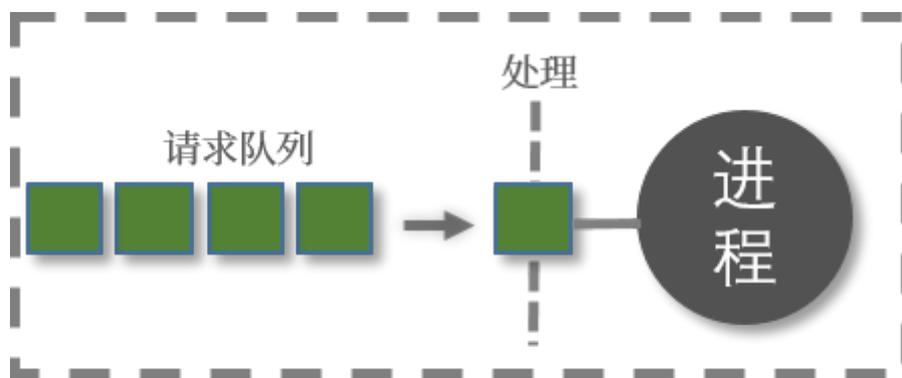
异步(asynchronous)IO

进程发起调用，被调用者立即返回给调用者，此时进程处于非阻塞状态可以接受其他请求。当数据准备好了，内核直接复制数据给进程，然后内核向进程发送通知取数据，两个阶段都是非阻塞状态。

Web服务器响应模型

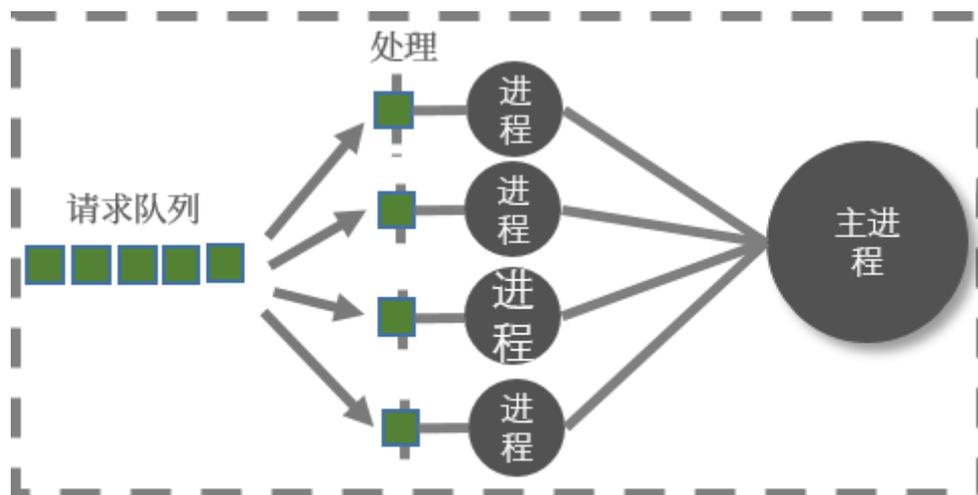
Web服务器响应客户端请求有四种模型：[单进程I/O模型](#)、[多进程I/O并发模型](#)、[复用I/O模型](#)、[复用I/O多线程模型](#)。

[单进程I/O模型](#)：有单个进程处理请求，单个进程同一时间只能处理一个请求。



单进程I/O模型

[多进程I/O模型](#)：有一个主进程生成多个子进程，每个子进程处理一个请求。Apache中的prefork和worker模型都使用的多进程I/O并发模型。



多进程及线程I/O并发模型

复用I/O多线程模型：由一个主进程生成多个子进程，每个子进程响应多个请求。在复用I/O多线程模型中，有一个监控连接的线程负责监控。单个进程监控多个连接的状态，如果某个连接需要操作时，进程就操作某个连接，代理可以监控连接状态。Apache中的event响应模型就是用了此种模型。

