链滴

I/O 与零拷贝

作者：WTF

原文链接：https://ld246.com/article/1613978365718

来源网站：链滴

【整理】【原文：[http://www.52im.net/thread-3280-1-1.html 】

<h2 id="1-什么是I-O-">1、什么是 I/O？</h2>
I/O 就是简单的数据 copy。
Q1：既然是 copy 数据，那么从哪里 copy 到哪里？
数据从外部设备 copy 到内存就是 Input，
数据从内存 copy 到外部设备就是 OutPut。
内存与外部设备不停的来回 copy 数据就是 Input and Output，简称 I/O。
<h2 id="2-常见I-O操作">2、常见 I/O 操作</h2>

 各种语言的打印操作
 读写文件
 网络通信
 鼠标键盘
 屏幕显示
 ……

<h2 id="3-I-O与CPU">3、I/O 与 CPU</h2>
CPU 执行机器指令的速度是纳秒级别的，而通常的 I/O 如磁盘工作，一次磁盘 seek（寻道）大在毫秒级别。
Q1：既然二者速度相差这么大，该如何设计以便更高效的利用系统资源？
既然有速度差异，而且进程在执行完 I/O 操作前不能继续向前推进，那就只有一个办法，等待 wat。先做其他事情，等 I/O 操作结束了再切换回来。
<h2 id="4-执行I-O时底层发生了什么">4、执行 I/O 时底层发生了什么</h2>
在支持线程的操作系统中，实际上被调度的是线程而不是进程。
我们暂时假设操作系统只有进程这样的概念进行讨论：
现在内存中有两个进程，进程 A 和进程 B，当前进程 A 正在运行

进程 A 中有一段读取文件的代码，通常我们定义一个用来装数据的 buff，然后调用 read 之类函数：read(buff);
这是一种典型的 I/O 操作，当 CPU 执行到这段代码的时候会向磁盘发送读取请求。
由于外部设备执行 I/O 操作是相当慢的，因此在 I/O 操作完成之前进程是无法继续向前推进的这就是所谓的阻塞 block。
操作系统检测到进程向 I/O 设备发起请求后就暂停进程的运行，
怎么暂停运行呢？
<pre><code class="highlight-chroma">只需要记录下当前进程的运行状态并把CPU的PC寄存器向其它进程的指令就行了。
</code></pre>
进程有暂停就会有继续执行，因此操作系统必须保存被暂停的进程以备后续继续执行，显然我们以用队列来保存被暂停执行的进程。
如下图，进程 A 被暂停执行并被放到阻塞队列中

<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2021/02/image-ab9eb5b1.png?imageView2/2/interlace/1/format/jpg">
这时操作系统已经向磁盘发送了 I/O 请求，因此磁盘 driver 开始将磁盘中的数据 copy 到 A 的 buffer 中。虽然这是进程 A 已经被暂停执行了，但这并不妨碍磁盘向内存中 copy 数据。
注意： 现代磁盘向内存 copy 数据是无需借助 CPU 的帮助，这就是 DMA（Direct Memory Access）
实际上：操作系统中除了有阻塞队列外也有就绪队列，所谓就绪队列就是只队列里的进程准备就可以被 CPU 执行了。
Q1：为什么不直接执行非要有个就绪队列？
即使只有 1 个核的机器上也可以创建出成千上万个进程，CPU 不可能同时执行这么多进程，因必然存在这样的进程，即使其一切准备就绪也不能被分配到计算资源，这样的进程就被放到了就绪队。
当进程 A 被暂停后 CPU 是不会闲下来的，操作系统会在就绪队列中找下一个可以执行的进程，就是这里的进程 B。此时操作系统将进程 B 从就绪队列中取出，找出进程 B 被暂停时执行到的机器指的位置，然后将 CPU 的 PC 寄存器指向该位置，运行进程 B。
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2021/02/image-e744c07b.png?imageView2/2/interlace/1/format/jpg"></>
此时进程 B 在被 CPU 执行，磁盘在向进程 A 的内存空间中 copy 数据，在操作系统的调度下，据 copy 和指令执行在同时进行。
此后磁盘终于将全部数据都 copy 到了进程 A 的内存中，这磁盘通知操作系统任务完成了。

如何通知？中断
操作系统接收到磁盘中断后发现数据 copy 完毕，进程 A 重新获得继续运行的资格，于是操作系将 A 从阻塞队列放到了就绪队列。
此时 B 继续执行，A 继续等待，进程 B 执行了一会后操作系统认为 B 执行的时间够长了，因此进程 B 放到就绪队列，把进程 A 取出来并继续执行。
注意： 操作系统把 B 放到的是就绪队列，因此进程 B 被暂停运行仅仅是为时间片到了而不是因为 I/O 阻塞。
这种进程执行 I/O 操作被阻塞暂停执行的方式被称为阻塞式 I/O，blocking I/O。
这里的讨论对于线程一样成立。
<h2 id="5-零拷贝">5、零拷贝</h2>
上面的讲解中我们直接把磁盘数据 copy 到了进程空间中，但实际上一般情况下 I/O 数据是要首 copy 到操作系统内部，然后操作系统再 copy 到进程空间中。
因此我们可以看到这里起始还有一层经过操作系统的 copy，对于性能要求很高的场景起始也是以绕过操作系统直接进行数据 copy 的，这也是上文描述的场景，这种技术称为 Zero-copy，零拷贝
【整理】【原文：[http://www.52im.net/thread-3280-1-1.html 】