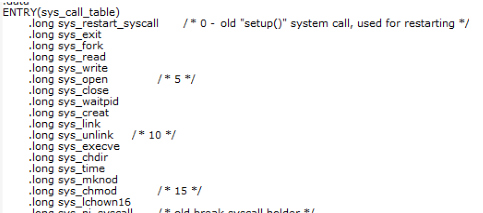
我们在使用与fork有关的函数时，例如fork(),vfork()，clone()，其本质都是利用了do\_fork()来帮我们实现的。在linux系统中有着用户空间和内核空间，我们无法直接在用户空间复制一个进程出来，而是产生一个中断，并且进行现场保护，随后，将系统调用号写入寄存器，由寄存器带入内核，最终让内核帮助我们完成复制进程的操作。

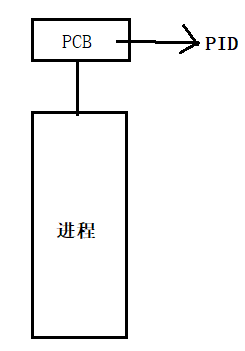
完成进程的复制分为了几个过程：

在执行fork()的时候，产生了一个中断，并且将系统调用号写入寄存器，由寄存器带入内核空间，内核根据系统调用号去和系统调用表进行匹对，让内核去执行复制进程的操作。（每一个系统调用号都能够在系统调用表的相应位置找到）。



系统调用表

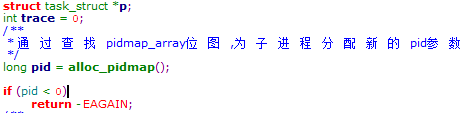
一个进程的模型可以这样表示



PCB是一个进程控制块，PCB里面的PID，即进程标识符。PID能够唯一标识一个进程。

（1）.要想复制一个进程，我们首先就要为新进程获取进程描述符。但是，在获取进程描述符之前，我们要先为子进程分为一个PID参数。

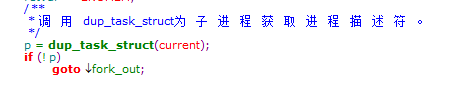
获取PID参数的方法在源码里是利用了pidmap\_array位图。源码如下:



（2）.接下来复制进程描述符，主要是通过copy\_process这个函数实现的。如果所有必须的资源都是可用的,该函数返回刚创建的task\_struct描述符的地址.

copy\_process主要进行了以下操作:

为子进程获取进程描述符



并且这个函数会将父进程的thread\_info结构中的数据复制到子进程的thread\_info结构中。

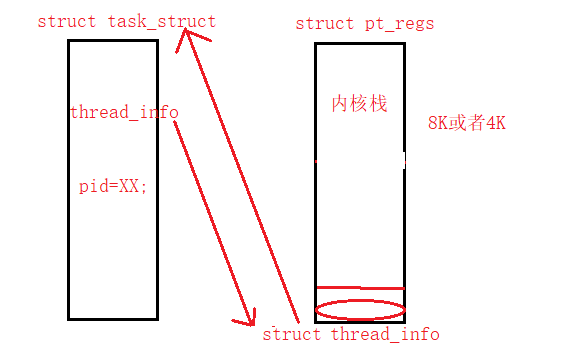
实现将父进程的thread\_info结构中的数据复制到子进程的thread\_info结构中的操作是由alloc\_thread\_info实现的。

alloc\_thread\_info宏会先获取一块空闲内存区，用来存放新进程的thread\_info结构和内核栈。

这块内存区字段的大小是8KB或者4KB。可以简单的理解为先申请了8KB或者4KB的空间去

储存信息。

随后会有一个将current进程描述符的内容复制到tsk所指向的task\_struct结构中，然后把tsk\_thread\_info置为ti，将current进程的thread\_info内容复制给ti指向的结构中，并将ti\_task置为tsk的过程。



将父进程中数据结构中的值复制到新进程中，在这里，我们主要关心的是文件描述符和内存描述符的复制。



将CPU寄存器中储存的值（保存在父进程的内核栈中）用于初始化子进程的内核栈。copy\_thread把eax寄存器对应字段的值（这是fork和clone系统调用在子进程中的返回值）强行置为0。

