1. 进程崩溃的本质：CPU内部有一个状态寄存器，如果CPU的计算过程出现了错误，错误信息就会被存储在这个状态寄存器中，CPU也会立马终止进程的代码。
2. 操作系统随时在监视CPU的状态，CPU的状态寄存器记录了异常，操作系统立马就会去给造成异常的进程发送信号，不同的信号包含了不同的错误信息，我们就看到了进程的报错信息。

这里讲一个重要的调试方法，调试是非常重要的。

**如何定位程序究竟是在哪一行奔溃的？**

调试原理：信号之中都有一个core dump标志位。这个标志位的默认值为0，对于大多数进程终止信号，都会将该标志位设置为1，然后将进程的信息转存到core文件中，方便我们后期调试。

调试步骤：

1. ulimit -c 查看core文件的大小上限，如果为0，表示不会生成core文件，我们需要给core文件设置一个合适的大小，直接设置为不限制大小也可以。
2. g++ -g xxx.cpp -o xxx来生成一个可调试的执行文件。
3. 执行该可执行文件，发生段错误，生成core文件。
4. 使用gdb进行调试。

注：有时会出现没有core文件的情况，core文件是由core文件的配置文件控制生成的。配置文件为/proc/sys/kernel/core\_pattern，可以设置pid，uid等属性，有兴趣可以研究一下，一般来说，这些属性用不到的。

有时这个配置文件会设定为将内容重定向到某个文件中。

该文件不能直接修改，标准的修改方法是：  
sudo bash -c "echo core > /proc/sys/kernel/core\_pattern"

这样生成的core文件就在当前目录中了。