

## 15.21 诊断分段错误¶

### 问题¶

解释器因为某个分段错误、总线错误、访问越界或其他致命错误而突然间崩溃。你想获得Python堆栈信息，从而找出在发生错误的时候你的程序运行点。

### 解决方案¶

`faulthandler` 模块能被用来帮你解决这个问题。在你的程序中引入下列代码：

```
import faulthandler
faulthandler.enable()
```

另外还可以像下面这样使用 `-Xfaulthandler` 来运行Python：

```
bash % python3 -Xfaulthandler program.py
```

最后，你可以设置 `PYTHONFAULTHANDLER` 环境变量。开启 `faulthandler` 后，在C扩展中的致命错误会导致一个Python错误堆栈被打印出来。例如：

```
Fatal Python error: Segmentation fault

Current thread 0x00007fff71106cc0:
  File "example.py", line 6 in foo
  File "example.py", line 10 in bar
  File "example.py", line 14 in spam
  File "example.py", line 19 in <module>
Segmentation fault
```

尽管这个并不能告诉你C代码中哪里出错了，但是至少能告诉你Python里面哪里有错。

### 讨论¶

`faulthandler` 会在Python代码执行出错的时候向你展示跟踪信息。至少，它会告诉你出错时被调用的最顶级扩展函数是哪个。在 `pdb` 和其他Python调试器的帮助下，你就能追根溯源找到错误所在的位置了。

`faulthandler` 不会告诉你任何C语言中的错误信息。因此，你需要使用传统的C调试器，比如 `gdb`。不过，在 `faulthandler` 跟踪信息可以让你去判断从哪里着手。还要注意是在C中某些类型的错误可能不太容易恢复。例如，如果一个C扩展丢弃了程序堆栈信息，它会让 `faulthandler` 不可用，那么你也得不到任何输出（除了程序崩溃外）。