

14.12 调试基本的程序崩溃错误 ¶

问题 ¶

你的程序崩溃后该怎样去调试它？

解决方案 ¶

如果你的程序因为某个异常而崩溃，运行 `python3 -i someprogram.py` 可执行简单的调试。`-i` 选项可让程序结束后打开一个交互式shell。然后你就能查看环境，例如，假设你有下面的代码：

```
# sample.py
```

```
def func(n):  
    return n + 10
```

```
func('Hello')
```

运行 `python3 -i sample.py` 会有类似如下的输出：

```
bash % python3 -i sample.py  
Traceback (most recent call last):  
  File "sample.py", line 6, in <module>  
    func('Hello')  
  File "sample.py", line 4, in func  
    return n + 10  
TypeError: Can't convert 'int' object to str implicitly  
>>> func(10)  
20  
>>>
```

如果你看不到上面这样的，可以在程序崩溃后打开Python的调试器。例如：

```
>>> import pdb  
>>> pdb.pm()  
> sample.py(4)func()  
-> return n + 10  
(Pdb) w  
sample.py(6)<module>()  
-> func('Hello')  
> sample.py(4)func()  
-> return n + 10  
(Pdb) print n  
'Hello'  
(Pdb) q  
>>>
```

如果你的代码所在的环境很难获取交互shell（比如在某个服务器上），通常可以捕获异常后自己打印跟踪信息。例如：

```
import traceback  
import sys
```

```
try:
```

```
func(arg)
except:
    print('**** AN ERROR OCCURRED ****')
    traceback.print_exc(file=sys.stderr)
```

要是你的程序没有崩溃，而只是产生了一些你看不懂的结果，你在感兴趣的地方插入一下 `print()` 语句也是个不错的选择。不过，要是你打算这样做，有一些小技巧可以帮助你。首先，`traceback.print_stack()` 函数会你程序运行到那个点的时候创建一个跟踪栈。例如：

```
>>> def sample(n):
...     if n > 0:
...         sample(n-1)
...     else:
...         traceback.print_stack(file=sys.stderr)
...
>>> sample(5)
File "<stdin>", line 1, in <module>
File "<stdin>", line 3, in sample
File "<stdin>", line 3, in sample
File "<stdin>", line 3, in sample
File "<stdin>", line 3, in sample
File "<stdin>", line 3, in sample
File "<stdin>", line 5, in sample
>>>
```

另外，你还可以像下面这样使用 `pdb.set_trace()` 在任何地方手动的启动调试器：

```
import pdb

def func(arg):
    ...
    pdb.set_trace()
    ...
```

当程序比较大而你想调试控制流程以及函数参数的时候这个就比较有用了。例如，一旦调试器开始运行，你就能够使用 `print` 来观测变量值或敲击某个命令比如 `w` 来获取追踪信息。

讨论

不要将调试弄的过于复杂化。一些简单的错误只需要观察程序堆栈信息就能知道了，实际的错误一般是堆栈的最后一行。你在开发的时候，也可以在你需要调试的地方插入一下 `print()` 函数来诊断信息（只需要最后发布的时候删除这些打印语句即可）。

调试器的一个常见用法是观测某个已经崩溃的函数中的变量。知道怎样在函数崩溃后进入调试器是一个很有用的技能。

当你想解剖一个非常复杂的程序，底层的控制逻辑你不是很清楚的时候，插入 `pdb.set_trace()` 这样的语句就很有用了。

实际上，程序会一直运行到碰到 `set_trace()` 语句位置，然后立马进入调试器。然后你就可以做更多的事了。

如果你使用IDE来做Python开发，通常IDE都会提供自己的调试器来替代pdb。更多这方面的信息可以参考你使用的IDE手册。

