

# 异常处理

## SetUnhandledExceptionFilter

```
LPTOP_LEVEL_EXCEPTION_FILTER SetUnhandledExceptionFilter(  
    LPTOP_LEVEL_EXCEPTION_FILTER lpTopLevelExceptionFilter    // 异常处理回调函数  
);
```

当异常没有处理的时候,系统就会调用 `SetUnhandledExceptionFilter` 所设置异常处理函数。

当在 `UnhandledExceptionFilter` 函数获取到了控制权且未在调试下, 该回调函数则会被执行, 如果为 `NULL`, 则使用 `UnhandledExceptionFilter` 的默认处理

### 3、只有程序被调试时, 才会存在未处理异常

`UnhandledExceptionFilter` 的执行流程:

1) 通过 `NtQueryInformationProcess` 查询当前进程是否正在被调试, 如果是, 返回 `EXCEPTION_CONTINUE_SEARCH`, 此时会进入第二轮分发

2) 如果没有被调试:

    查询是否通过 `SetUnhandledExceptionFilter` 注册处理函数 如果有就调用

    如果没有通过 `SetUnhandledExceptionFilter` 注册处理函数 弹出窗口 让用户选择终止程序还是启动即时调试器

    如果用户没有启用即时调试器, 那么该函数返回 `EXCEPTION_EXECUTE_HANDLER`

## 通过其进行反调试

如果在调试状态, 则顶层异常处理是不会被执行的, 可进行反调试

## 反反调试

对 `NtQueryInformationProcess` 的返回值做手脚