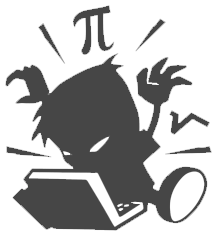


dpull

面包屑小道.

主页

简介

Search

使用Clang/GCC的Sanitizer提升程序质量

Published on 2016/11/25

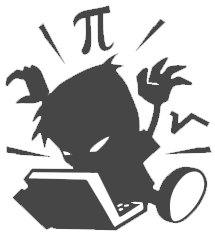
.

在Clang的Controlling Code Generation 或 GCC的Program Instrumentation Options中提供了一系列的参数，可以帮程序解决一些常见的内存问题。

本文使用Clang。

AddressSanitizer

内存错误是C / C++的最常见的问题，可能导致程序诡异崩溃，十分难查。内存错误检查AddressSanitizer可用来查找如内存溢出，内存重叠等问题，会带来2倍的速度消耗。



dpu

面包屑小

主页

简介

Search

开启方法：

- XCode Edit Scheme->Run->Diagnostics->勾选Address Sanitizer
- clang -fsanitize=address test.c

示例：

- AddressSanitizer: stack-buffer-overflow

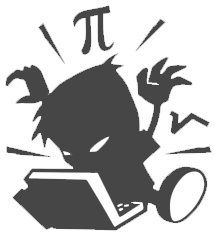
```
long long n = 0;
if (*(long long*)&n == *(long lo
```

- AddressSanitizer: heap-buffer-overflow

```
char *ptr = (char *)malloc(5)
if (ptr[-1] == '\0');
if (ptr[12] == '\0');
```

- AddressSanitizer: heap-use-after-free

```
class Test
{
```



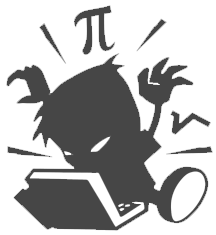

```
public:
    int Release()
    {
        if (--m_nRef == 0)
            delete this;
        return m_nRef;
    }
    int m_nRef = 1;
};

auto p = new Test;
p->Release();
```

- AddressSanitizer: strncpy-param-overlap: memory ranges

```
char sz[64];
sz[0] = '\0';
strncpy(sz, sz, sizeof(sz));
```

UndefinedBehaviorSanitizer

对一些未定义行为的检查，如有



dpu

面包屑小

主页

简介

Search

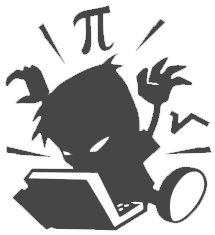
符号整数溢出。可使用 UndefinedBehaviorSanitizer来检查。

```
int num = 100;
for (int i = 0 ; i < 10; ++i)
{
        num *= num; // num在第
        printf("%d\n", num);
}
```

开启方法：

- XCode 在Other Warning Flags（其他也可）添加： -fsanitize=undefined-trap -fsanitize-undefined-trap-on-error
- clang -fsanitize=undefined-trap -fsanitize-undefined-trap-on-error undefined.c

未完待续



dpu

面包屑小

主页

简介

Search