ToiletPaper #56

jambit
WHERE INNOVATION WORKS

Undefined behavior in C, Objective C, and C++

Author: Andreas Maier / Software Architect / Business Division Banking & Insurance

✘ Problem

What is the result of `i = ++i + i++`?

Anything can happen: the standard imposes no requirements. The program may fail to compile, may execute incorrectly (crashing or silently generating incorrect results), or may do exactly what the programmer intended. **The exact behavior is undefined**!

The resulting behavior depends upon not only the specific hardware, platform, and compiler but also on specific settings (optimization level, debug/ release version) of the same compiler. This results in code that is not portable and bugs that are hard to detect.

✓ Solution

To get rid of undefined behavior we should understand the concept of sequence point. According to the C standard there is a sequence point
1. at the end of each full expression (typically, at the semicolon `;`)
2. after the evaluation of all function arguments and before execution of the function body 3. after the evaluation of the expression `a`, using the built-in (non-overloaded) operators

`a && b`

`a || b`

`a? b : c`

`a , b`

There are two main cases with undefined behavior **between two sequence points**

1.
```
i = i++ + 1;         // undefined behavior
i = ++i + 1;         // undefined behavior (well-defined in
C++11) ++++i;              // undefined behavior (well-defined
in C++11) f(++i, ++i);       // undefined behavior
f(i = -1, i = -1); // undefined behavior
```

Solution: between two sequence points the value of a scalar object shall be **modified** at **most once**.

2.
```
cout << i << i++; // undefined behavior
a[i] = i++;       // undefined behavior
```

Solution: between two sequence points the prior value of a modified scalar object shall be **accessed only to determine the value** to be stored.

Further aspects

- Enable and heed **compiler warnings**
- Use **static analyzers** (like clang's, cppcheck, etc.) to get even more warnings