C++ static code analysis: Functions which do not return should be declared as "noreturn"

1 minute

The attribute `noreturn` indicates that a function does not return. This information clarifies the behavior of the function and it allows the compiler to do optimizations.

It can also help the compiler (and static analyzer tools, i.e. us) provide better error messages:

```
__attribute__((noreturn)) void f();

int g(int b) {
  if (b == 5) {
    f();
    printf("Hello world\n"); // This is dead code, the compiler/static analyzer can now detect it
    // There is no returned value, but it is fine, the compiler/static analyzer knows not to warn about it
  } else {
    return 3;
  }
}
```

This rule detects when the attribute `noreturn` can be added to a function.

Noncompliant Code Example

```
void g() { // Noncompliant
  abort();
}
```

Compliant Solution

```
__attribute__((noreturn)) void g() { // or [[noreturn]] for C++
  abort(); // Compliant
}
```