

The rule of 3 5 0

20171703 소프트웨어융합학과 정태원

The rule of Three

클래스는 사용자 정의에 의한 소멸자, 복사 생성자, 복사 할당 연산자를 필요로 합니다. 많은 경우에 이 복사 생성자와 복사 할당 연산자는 값은 전달하거나, 내부의 컨테이너의 내용을 조작하기 위해 쓰입니다. 정의하지 않는다면 컴파일러의 의해 자동 생성됩니다. 컴파일러에 의해 생성되는 복사 생성자는 클래스 타입이 아닌 포인터나 파일 서술자를 다룬다면 정확하지 못하게 됩니다. 흔하게

얕은 복사(shallow copy) 발생합니다. 얕은 복사는 데이터를 새로 담은 깊은 복사가 아닌 주소를 같이 하는 경우가 많아 데이터의 내용을 변경할 때 원하지 않는 결과가 번번히 발생합니다.

복사 생성자를 깊은 복사로 변경 해주는 것은 private으로 데이터를 지정하고 정보가 변경되지 않게 하는데 매우 용이합니다.

The rule of five

위 세가지 생성자, 복사 생성자 그리고 복사 할당자의 유무가 이동 생성자와 이동 할당생성자 자동 생성을

막습니다. 그러기에 이동 생성자를 쓸거면 5가지 모두 정의해야 한다.

위 세가지와는 다르게 굳이 쓰지 않아도 에러는 발생하지 않습니다. 그러나 최적화 기능을 잃게 됩니다.

The rule of zero

소멸자, 복사, 이동 생성자, 복사, 이동 할당연산자가 정의되어있는 class는 OOP 단일 책임 원칙에 따라 새로운 소멸자, 생성자, 할당연산자가 정의되어선 안됩니다. 즉 CGF에 해당하는 내용을 모두 적거나 아니면 모두 적지 않거나 하라는 것입니다.

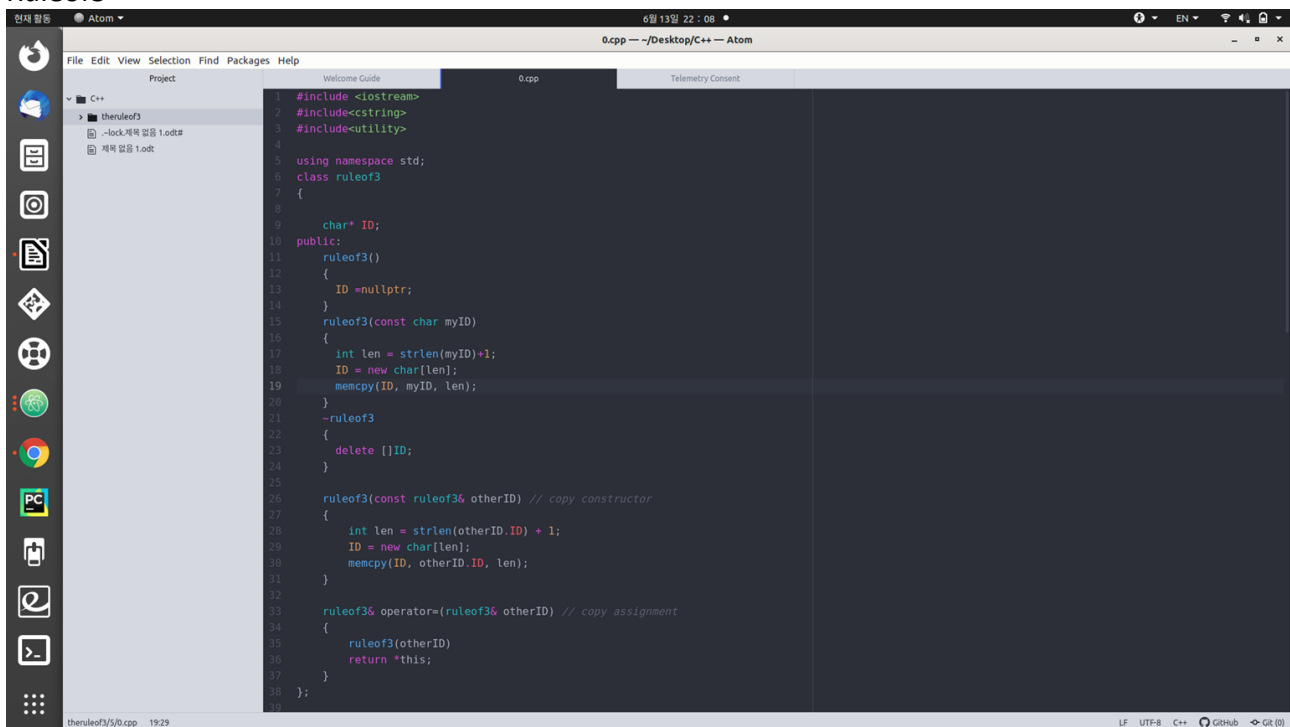
CGF(compile generated Function) – C ++11 부터 이동 생성자, 이동 대입 연산자를 지원함에 따라 컴파일러가 자동으로 생성해 주는 함수들이 생성되었습니다.

아래 다섯 가지가 이에 해당됩니다.

- destructor
- copy constructor
- move constructor
- copy assignmen operator

- move assignment operator

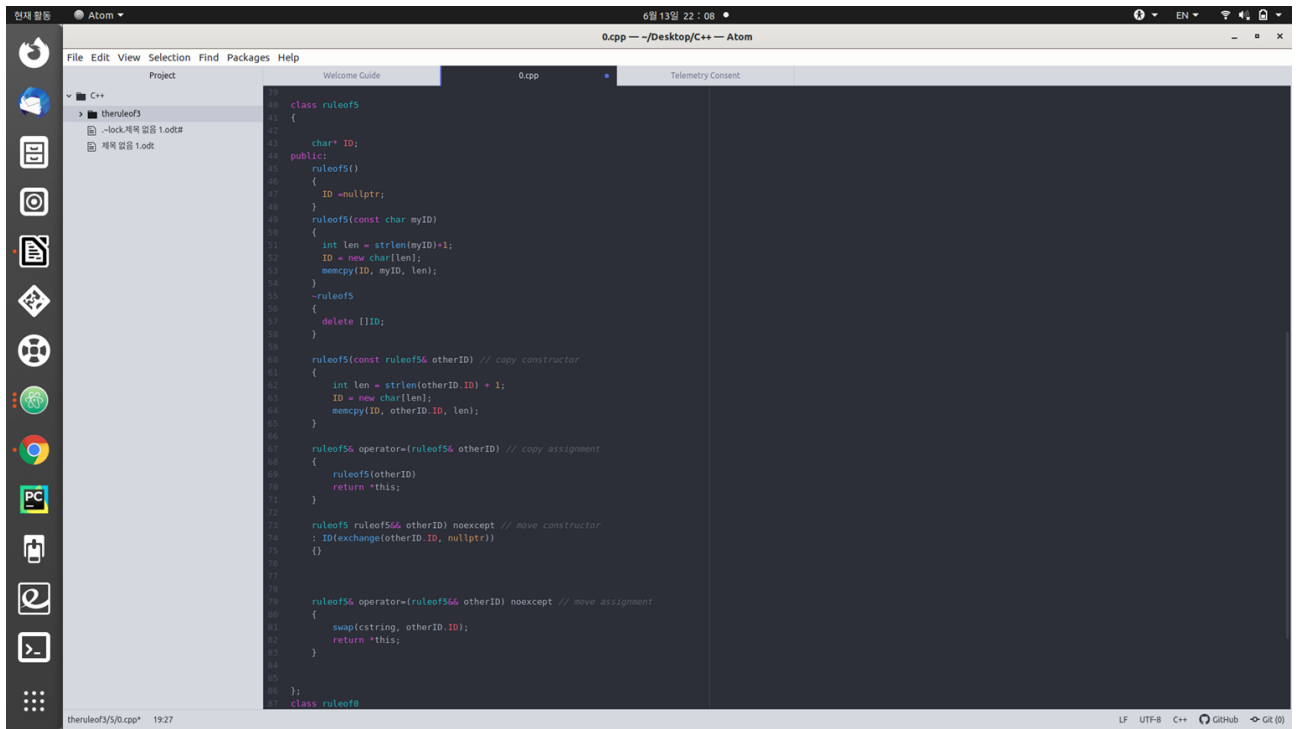
Ruleof3



```
1 #include <iostream>
2 #include <cstring>
3 #include <utility>
4
5 using namespace std;
6 class ruleof3
7 {
8     char* ID;
9 public:
10     ruleof3()
11     {
12         ID = nullptr;
13     }
14     ruleof3(const char myID)
15     {
16         int len = strlen(myID)+1;
17         ID = new char[len];
18         memcpy(ID, myID, len);
19     }
20     ~ruleof3()
21     {
22         delete []ID;
23     }
24
25     ruleof3(const ruleof3& otherID) // copy constructor
26     {
27         int len = strlen(otherID.ID) + 1;
28         ID = new char[len];
29         memcpy(ID, otherID.ID, len);
30     }
31
32     ruleof3& operator=(ruleof3& otherID) // copy assignment
33     {
34         ruleof3(otherID)
35         return *this;
36     }
37 };
38
39
```

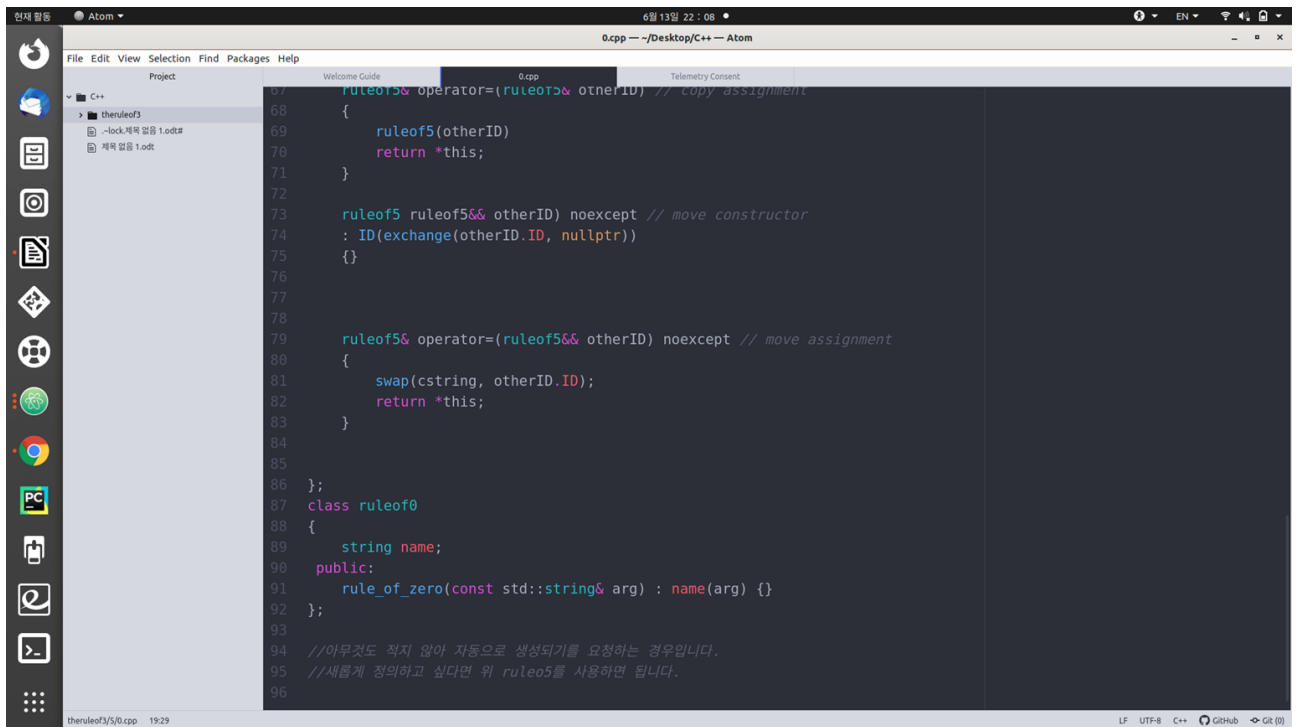
deepcopy는 memcpy를 활용하였습니다.

Ruleof5



moveconstructor는 이동생성자로 shallowcopy 즉 같은 메모리값을 가져와 데이터를 활용하기를 원하는 생성자입니다. 5의 원칙대로 5가지 모두 생성하였습니다.

Ruleof0



정말로 5가지 중에 어느 것도 생성하지 않았습니다. 제로의 원칙은 oop의 단일 책임 원칙에 따라 5가지 중에 어느 것도 생성하지 않는 원칙입니다.

이상입니다.

참고:

<https://nodealdev.tistory.com/58> [nodeal dev story]

https://en.cppreference.com/w/cpp/language/rule_of_three