

C언어의 사용하지 않는 구조체 필드 탐지

허진영

서울대학교 프로그래밍연구실

2013.01.31

동기

- 코드가 길어질수록 유지 보수 비용 증가
- SW 시스템에서 사용 않는 코드의 비율은 약 25% 정도*
- 안 쓰는 코드로 인한 바이너리 크기 및 메모리 사용량 증가

* Junker, M., Jurgens, E., Hauptmann, B., Vaas, R., Prommer, K. How much does unused code matter for maintenance? ICSE 2012.

원인

- 코드 재사용
 - 기존 제품의 코드를 기반으로 새 제품 구현
- 빈번한 기능 추가 삭제
 - 경쟁사 대비 특화 기능
- 필요할 것 같은 기능을 미리 구현
 - 하지만 안 쓰는 경우도 많다
 - YAGNI (You Aren't Gonna Need It!)
- 보수적인 개발자
 - 일단 동작하는 코드는 건드리면 안됨!
- 코드 리팩토링

사용하지 않는 코드의 예

- 정의는 했지만 사용하지 않는 변수/함수

```
static int g = 0;

void foo(void) {}

void bar(void)
{
    int i = 0;
    i++;

    g = 0;
    g += 1;
}
```

사용하지 않는 코드의 예

- 도달할 수 없는 코드 블록

```
void foo(void)
{
    if (value > 0) return;
    ...
    if (value > 3) {
        ...
    }
}
```

```
void foo(int cmd)
{
    switch (cmd) {
        case 1: ... break;
        case 2: ... break;
        case 3: ... break;
    }
}

int main(void)
{
    foo(1);
    foo(2);
    return 0;
}
```

사용하지 않는 코드의 예

- 실행해도 아무 효과가 없는 함수/코드 블록

```
void foo(int i) { i++; }

void bar(void)
{
#ifdef SOME_OBSOLETE_FEATURE
    ...
#endif
}

int main(void)
{
    foo(1);
    bar();
    return 0;
}
```

사용하지 않는 코드의 예

- 사용되지 않는 구조체 멤버 필드

```
typedef struct
{
    char* from;
    int action;
} Message;

void foo(Message* p)
{
    if (p->from)
    {
        p->action = 0;
    }
}
```

기존 연구

- Peter F. Sweeney, Frank Tip. A Study of Dead Data Members in C++ Applications. PLDI 1998.
- 도달 가능한 함수에서 read 되지 않는 필드 탐지
- Read는 되나 그 코드가 dead code 인 경우는 탐지 불가

앞으로의 계획

- 기존 방식을 보완한 방식 고안
 - dead code에서 접근되는 필드 탐지
- 자동화 구현

감사합니다.