

손에 잡히는 vim : quickfix

김선영

[sunzero@gmail\(dot\)com](mailto:sunzero@gmail.com)

버 전: 2014-11-07

인사이트 출판사 <http://blog.insightbook.co.kr>

가메 출판사 <http://www.kame.co.kr>

저자홈페이지 <http://sunzero.tistory.com>

vim quickfix

- ❖ **make**를 이용하여 컴파일, 링킹 에러 감지
 - ▶ **compiler, linker**의 에러 메시지를 **vim**이 자동으로 해석한다.
 - ↳ 전체조건 : **header**와 선언 관계는 명확해야 한다.
 - ▶ 예제 : **myerr.c**

Practice

❖ 예제 프로그램 : myerr.c

```
#include <stdio.h>
int main()
{
    print_hello();
    return 0;
}
void print_hello()
{
    printf("Hello world\n");
}
```

Practice (con't)

❖ 예제 프로그램 빌드

```
$ make myerr
```

```
cc      myerr.c  -o myerr
```

```
myerr.c:8: warning: conflicting types for 'print_hello'
```

```
myerr.c:4: note: previous implicit declaration of 'print_hello' was here
```

```
$ make -B myerr
```

```
cc      myerr.c  -o myerr
```

```
myerr.c:8: warning: conflicting
```

```
myerr.c:4: note: previous implicit declaration of 'print_hello' was here
```

매번 밖에서 컴파일 후에
다시 들어와서 해당 라인을 찾을 것인가?

build on vim

- ❖ shell 명령을 fork-exec로 실행해도 되지만...

" 생략...

```
nmap <F5> :!gcc -o %< -I. %
```

% : 현재 파일명

%< : 확장자를 제거한 현재 파일명

- ▶ 이건 좋은 방법이 아니다!!!

build on vim : quickfix

❖ 프로그래머의 quick-fix

" 생략...

```
nmap <F5> :mak %<
```

```
nmap <F6> :copen <CR>
```

default target을 쓰는 경우
:mak 만 넣어도 된다.

▶ copen / cclose

▶ cnext / cprev

Practice (con't)

- ❖ quickfix 빌드 : <F5>를 누르자

```
#include <stdio.h>
int main()
{
    print_hello();
    return 0;
}
```

```
void print_hello()
{
    printf("Hello world\n");
}
~
~
~
```

자동으로 에러, 경고 라인으로 위치한다.

(2 of 3): warning: conflicting types for 'print_hello'

8,1

Ⓜ

Practice (con't)

- ❖ quickfix 빌드 : <F6>을 누르자

```
    return 0;
}

void print_hello()
{
    printf("Hello world\n");
}
```

엔터를 치면 해당 라인으로 이동
코드창과 Quickfix List창 사이의 이동키는?

```
myerr.c 8,1 바드
|| cc    myerr.c  -o myerr
myerr.c|8| warning: conflicting types for 'print_hello'
myerr.c|4| note: previous implicit declaration of 'print_hello' was here
~
~
[Quickfix List] 2,1 모뎀
```


Practice2

❖ 예제 프로그램 : myerr2.c

```
#include <stdio.h>
void print_hello()
int main()
{
    print_hello();
    return 0;
}
void print_hello()
{
    write(1, "Hello world\n");
}
```

quickfix를 이용해서 에러를 수정해보자.

quickfix : list

명령어	설명
<code>:mak [make args]</code>	make를 실행한다.
<code>:cnext</code>	next error로 이동한다.
<code>:cprev</code>	previous error로 이동한다.
<code>:copen</code>	분할창으로 quickfix list 를 엽니다.
<code>:cclose</code>	분할창으로 열린 quickfix list 를 닫는다.

map : example

❖ .vimrc 파일

```
" [ map ]  
nmap <F2>      :up<CR>  
nmap <F3>      :tabnext<Enter>  
nmap <F4>      :tabprev<Enter>  
nmap <F5>      :mak %< <Enter>  
nmap <F8>      :copen<ENTER>  
nmap <F9>      :cnext<ENTER>
```