2018

# Operating System Lab 0

Embedded System Lab.

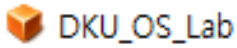
LAB 0 – 프로젝트 구성, 개발환경 구성, GDB

JUNHO PARK, CHOI JONG MOO

## A. 프로젝트 구성

**본 수업에서는 우분투 가상 머신 이미지를 배포하여 실습을 진행하며 배포된 이미지는 아래와 같은 구성을 가지고 있다.**

### 1. 배포 실습 이미지.




다운로드 링크 클릭

### 2. 실습 이미지 구성

- Image name : DKU_OS_Lab
- Operating System : Ubuntu 16.04 LTS
- Kernel Version : 4.13.0-36-generic
- Hardware Platform : x86_64
- ***login & root password : 1234***

### 3. 실습 이미지 수행 방법

참조된 실습 환경 구축 PPT 를 참조한다.

https://goo.gl/tM0Dde

### 4. 실습 내용

```
root@osLab:/home/oslab/DKU_OS_LAB# ls
lab1_sched  lab2_sync  lab3_fs
```

#### i. lab1_sched

FCFS, RR, SPN,SRT, HRRN, MLFQ User Scheduler 구현 과제

#### ii. lab2_sync

Multi threading 환경에서 공유 자원에 대한 race condition 해결 과제

#### iii. lab3_fs

ramdisk 상에서의 fat32 파일시스템 directory entry 추적 실습 과제

### 5. 과제 제출 방법

#### i. 구현 내용

압축하여 heyheygo99@naver.com 으로 제출

#### ii. 레포트

소프트웨어 ICT 관 515 호로 제출

## B. 개발 환경 구성

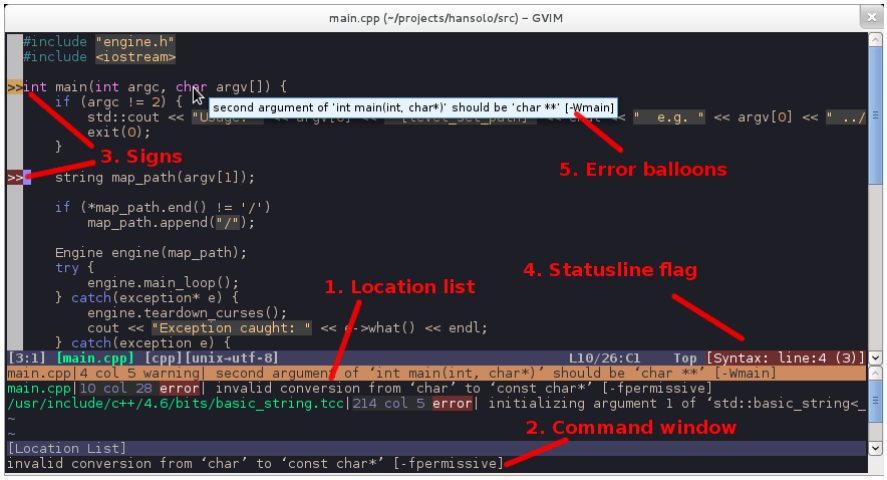
**본 수업에서는 배포된 우분투 실습 이미지의 vim editor 를 사용하여 개발한다. 배포된 vim editor 에는 개발에 유용하게 사용될 수 있는 몇가지 PlugIn 들이 있다.**

***vim editor 내부에서 분리된 창간의 이동은 ctrl + w + 방향키 를 통해 이동할 수 있다.***

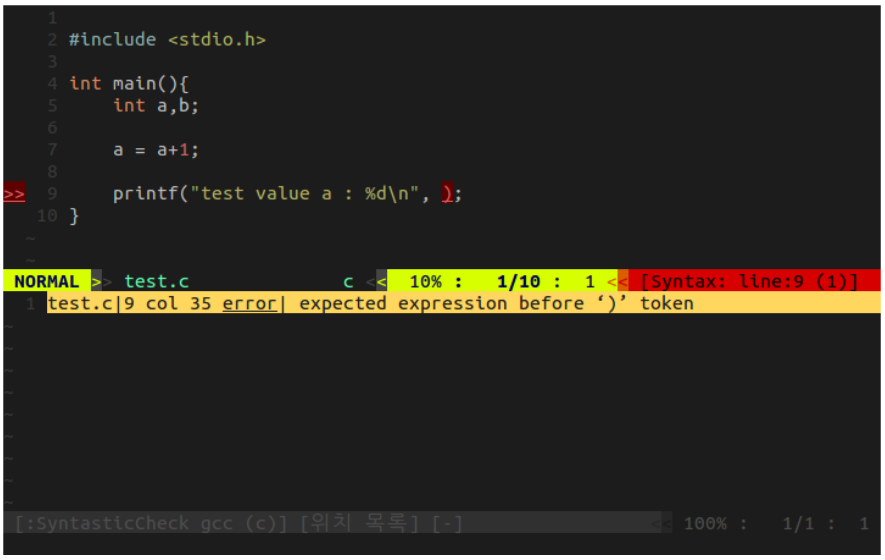
### 1. 설치된 vimrc 플러그인 및 툴

#### i. Syntastic

문법 체크 기능 플러그인으로 소스 작성 후 컴파일 과정에서 error 를 확인하는 것이 아닌 vim 에서 error 및 warning 을 확인할 수 있도록 해주는 플러그인으로 아래와 같이 다양한 기능을 제공한다




예시로 아래와 같은 9 번째 줄 printf 함수에 출력 대상을 지정하지 않은 예제 코드가 있다고 할 때, vim 창의 하단에 어느 소스코드의 몇번 째 line 에




무슨 에러가 발생 하였는지의 정보를 나타내며 상단의 소스코드 화면에서 문제가 발생한 line 을 표시해 준다. 이를 통해 vim 을통해 개발하며 error 를 확인할 수 있다. 소스코드 작성 후, 저장함으로써 에러를 업데이트하여 확인할 수 있다.

- *ctrl + w + 방향키* 를 통해 아래 창으로 이동하여 해당 에러가 난 위치로 갈 수 있다.
- 참고

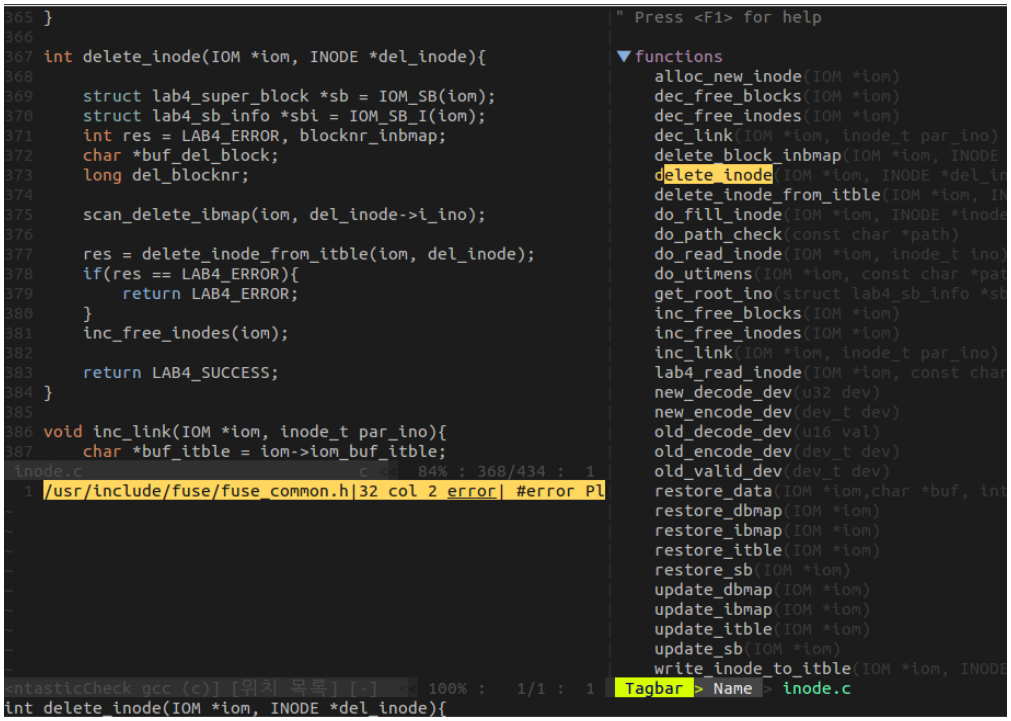
  https://github.com/vim-syntastic/syntastic

### ii. Tagbar

본 과제를 수행하다 보면 test.c 라는 간단한 소스코드가 아닌 다양한 함수 및 매크로, 변수 등을 사용할 수 있다. 이때 현재까지 작성한 소스코드의 손쉬운 관리를 위해 함수, 변수, 매크로 등을 확인하고 해당 위치로 이동하는 기능을 제공하는 플러그인이다.

- F8 을 눌러 해당 기능을 on/off 할 수 있다.
- *ctrl + w + 방향키* 를 통해 우측 창으로 이동하여 해당 함수, 매크로 변수 등으로 이동할 수 있다.
- 참고

  https://github.com/majutsushi/tagbar




### iii. AutoComplPop

자동완성기능을 해주는 플러그인이다.

- 참고

  https://github.com/vim-scripts/AutoComplPop

### iv. NERDTree

탐색기 기능을 하는 플러그인으로써 vim ediitor 에서 작업 도중 다른 파일로 옮겨가야 할 때 유용하게 사용할 수 있다. 아래와 같이 inode.c 라는 파일을 작성 중 file.c 등 다른 파일로 이동해야 할 때, NERDTree 플러그인을 통해 이동 및 조회가 가능하다.

- F7 을 눌러 해당 기능을 on/off 할 수 있다.
- *ctrl + w + 방향키* 를 통해 좌측 창으로 이동하여 다른 위치의 파일들을 열 수 있다.
- 참고

  https://github.com/scrooloose/nerdtree

### v. Ctags

Ctags 는 프로그래밍 소스코드의 태그(전역변수 선언, 함수 정의, 매크로 정의)들의 데이터 베이스(tags 파일)을 생성하는 명령어이다. Ctags 를 사용하고자하는 directory 로 이동하여 “# ctags –R”명령어를 이용하여 tags 파일을 생성한다.

- 사용하려는 directory 에서 tag 파일을 생성해 주어야 한다.

  # ctags -R

- vim editor 의 ex 모드에서 “:tj 함수명 or 구조체명”을 통해서 원하는 source code 를 찾을 수 있다.

  : tj 함수명

  : tj 구조체 명.

- 혹은 vi 를 열어서 source code 를 분석하는 중에 함수 원형이나 구조체이름에 커서를 위치시키고 “ctrl + ]”를 누르면 자동으로 태그를 찾아간다.이전으로 다시 되돌아 오고 싶은경우 “ctrl + t”를 사용해서 돌아올 수 있다.

- 참고

  https://en.wikipedia.org/wiki/Ctags