

CUnit을 이용한 unit testing

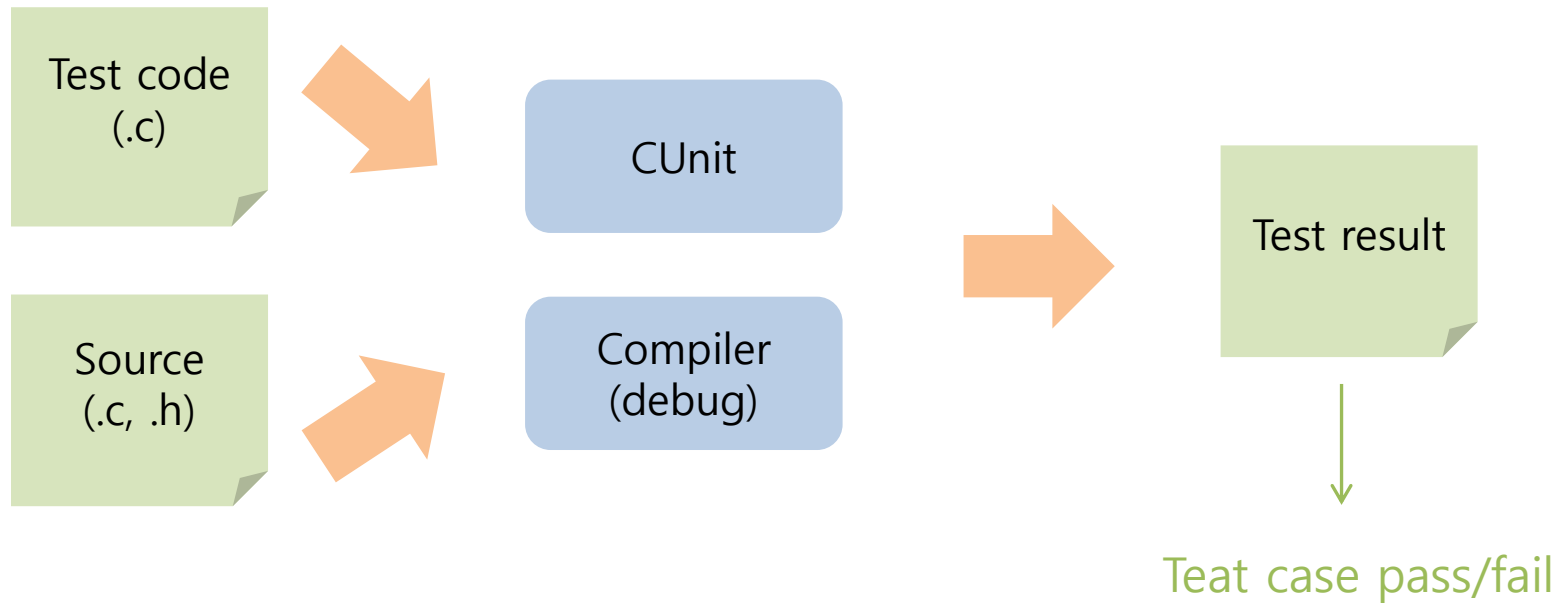
2013.11.28

윤상현

CUnit 환경

- Cygwin
- CUnit

CUnit을 이용한 테스트



CUnit 설치

- <http://sourceforge.net/projects/cunit/> 에서 CUnit-2.1-2-src.tar.gz 다운
- 압축을 풀어 cygwin의 home/username폴더에 복사
 - e.g) C:\cygwin\home\shyoon
- Cygwin을 실행하여, 해당 경로에서 아래 순서 대로 실행 (설치)
 - ./configure
 - make
 - make install

```
~/CUnit-2.1-2
shyoon@shyoon-PC ~/home
$ cd shyoon/

shyoon@shyoon-PC ~
$ gcc
gcc: fatal error: no input files
compilation terminated.

shyoon@shyoon-PC ~
$ ls
CUnit-2.1-2  CUnit-2.1-2-src.tar.bz2

shyoon@shyoon-PC ~
$ cd CUnit-2.1-2/

shyoon@shyoon-PC ~/CUnit-2.1-2
$ ls
aclocal.m4  config.sub  cunit.pc.in  INSTALL  Makefile.in  README
AUTHORS     configure  CUnit.spec.in  install-sh  Man  Share
ChangeLog   configure.in  depcomp  Jamrules.in  missing  TODO
config.guess  COPYING  doc  ltmain.sh  mkninstalldirs
config.h.in  CUnit  Examples  Makefile.am  NEWS

shyoon@shyoon-PC ~/CUnit-2.1-2
$ ./configure
```

```
~/CUnit-2.1-2
config.status: creating CUnit/Headers/Makefile
config.status: creating CUnit/Headers/CUnit.h
config.status: creating CUnit/Sources/Makefile
config.status: creating CUnit/Sources/Framework/Makefile
config.status: creating CUnit/Sources/Automated/Makefile
config.status: creating CUnit/Sources/Basic/Makefile
config.status: creating CUnit/Sources/Console/Makefile
config.status: creating CUnit/Sources/Curses/Makefile
config.status: creating CUnit/Sources/Test/Makefile
config.status: creating doc/Makefile
config.status: creating doc/headers/Makefile
config.status: creating Examples/Makefile
config.status: creating Examples/AutomatedTest/Makefile
config.status: creating Examples/BasicTest/Makefile
config.status: creating Examples/ConsoleTest/Makefile
config.status: creating Examples/CursesTest/Makefile
config.status: creating Man/Makefile
config.status: creating Man/man3/Makefile
config.status: creating Share/Makefile
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands

shyoon@shyoon-PC ~/CUnit-2.1-2
$ make
```

```
~/CUnit-2.1-2
make[3]: Leaving directory '/home/shyoon/CUnit-2.1-2/doc/headers'
make[3]: Entering directory '/home/shyoon/CUnit-2.1-2/doc'
make[3]: Nothing to be done for 'all-am'.
make[3]: Leaving directory '/home/shyoon/CUnit-2.1-2/doc'
make[2]: Leaving directory '/home/shyoon/CUnit-2.1-2/doc'
Making all in Man
make[2]: Entering directory '/home/shyoon/CUnit-2.1-2/Man'
Making all in man3
make[3]: Entering directory '/home/shyoon/CUnit-2.1-2/Man/man3'
make[3]: Nothing to be done for 'all'.
make[3]: Leaving directory '/home/shyoon/CUnit-2.1-2/Man/man3'
make[3]: Entering directory '/home/shyoon/CUnit-2.1-2/Man'
make[3]: Nothing to be done for 'all-am'.
make[3]: Leaving directory '/home/shyoon/CUnit-2.1-2/Man'
make[2]: Leaving directory '/home/shyoon/CUnit-2.1-2/Man'
Making all in Share
make[2]: Entering directory '/home/shyoon/CUnit-2.1-2/Share'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/shyoon/CUnit-2.1-2/Share'
make[2]: Entering directory '/home/shyoon/CUnit-2.1-2'
make[2]: Leaving directory '/home/shyoon/CUnit-2.1-2'
make[1]: Leaving directory '/home/shyoon/CUnit-2.1-2'

shyoon@shyoon-PC ~/CUnit-2.1-2
$ make install
```

```
~/CUnit-2.1-2
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/home/shyoon/CUnit-2.1-2/Man'
make[2]: Leaving directory '/home/shyoon/CUnit-2.1-2/Man'
make[1]: Leaving directory '/home/shyoon/CUnit-2.1-2/Man'
Making install in Share
make[1]: Entering directory '/home/shyoon/CUnit-2.1-2/Share'
make[2]: Entering directory '/home/shyoon/CUnit-2.1-2/Share'
make[2]: Nothing to be done for 'install-exec-am'.
test -z "/usr/local/share/CUnit" || /usr/bin/mkdir -p "/usr/local/share/CUnit"
/usr/bin/install -c -m 644 CUnit-List.dtd CUnit-List.xml CUnit-Run.dtd CUnit-Run.xml Memory-Dump.dtd Memory-Dump.xml '/usr/local/share/CUnit'
make[2]: Leaving directory '/home/shyoon/CUnit-2.1-2/Share'
make[1]: Leaving directory '/home/shyoon/CUnit-2.1-2/Share'
make[1]: Entering directory '/home/shyoon/CUnit-2.1-2'
make[2]: Entering directory '/home/shyoon/CUnit-2.1-2'
make[2]: Nothing to be done for 'install-exec-am'.
test -z "/usr/local/lib/pkgconfig" || /usr/bin/mkdir -p "/usr/local/lib/pkgconfig"
/usr/bin/install -c -m 644 cunit.pc '/usr/local/lib/pkgconfig'
make[2]: Leaving directory '/home/shyoon/CUnit-2.1-2'
make[1]: Leaving directory '/home/shyoon/CUnit-2.1-2'

shyoon@shyoon-PC ~/CUnit-2.1-2
$
```

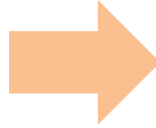
Test 대상 소스 수정

- main 함수 삭제
 - Test source에 main()을 정의하고, 이를 실행 시킴
 - 소스코드에 main()이 두 개 있을 수 없으므로 **test 대상 소스코드에 있는 main()을 삭제해야 됨**
 - 기존의 main()에서 단순히 함수호출을 하는 것이 아니라면 추가 작업을 해야 할 수 있음
 - 데이터 타입 선언을 test.c에 추가
 - main()이 일반적인 함수의 기능을 가지고 있었으면, 다른 함수로 재정의
 - ...
- 그 외 수정할 부분
 - .c file을 .h파일로 변환.
 - Multi definition error 해결
 - 대상 소스코드 마다 case by case로 추가 작업이 필요할 수 있음
 - 가능하면 원본 소스의 기능을 손상시키지 않는 수준에서 수정

```
int main() {

    char userInput, a;

    Bool DoorSensorInput = F;
    Bool Time_TempInput = F;
    Bool ModeButtonInput = F;
    Bool _10sec_10C_increButtonInput = F;
```

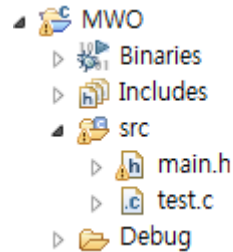


```
int main_controll() {

    char userInput, a;

    Bool DoorSensorInput = F;
    Bool Time_TempInput = F;
    Bool ModeButtonInput = F;
```

테스트용 main()과 중복, 함수 명 변경
추가적으로 test main()에 기존의 main()
기능을 추가해야 할 수도



Test 대상 파일의 이름을 변경
main.c -> main.h

Test code 작성

- Test를 위한 main작성 필요
- Main에서 test suite을 호출, test 대상 코드의 unit을 실행시킴
 - Test suite: test case set
 - Test case: 입력 값과 원하는 출력 값, C 함수 형태
- Test main()의 구조
 - Registry initialize
 - Registry에 test suite 추가
 - Test suite에 test cases 추가
 - Test run

Test 소스 코드 예제

```
#include "CUnit/Automated.h"
#include "CUnit/Basic.h"

#include "main.h"

//test cases
void simple_test_00(void){
    //CU_ASSERT_STRING_EQUAL(result, "expected")
    s.DoorState = 7;
    DoorSensorInterface(F);
    CU_ASSERT_EQUAL(s.DoorState, F);
}

void simple_test_01(void){}
void simple_test_02(void){}

int main()
{
    CU_pSuite pSuite = NULL;

    //initialize registry
    if(CUE_SUCCESS != CU_initialize_registry())
        return CU_get_error();

    pSuite = CU_add_suite("testing a suite", NULL, NULL);
    if(NULL==pSuite){
        CU_cleanup_registry();
        return CU_get_error();
    }

    //add test cases to the suite
    if(NULL==CU_add_test(pSuite, "simple_test0", simple_test_00)
        || (NULL==CU_add_test(pSuite, "simple_test1", simple_test_01))
        || NULL==CU_add_test(pSuite, "simple_test2", simple_test_02)){
        CU_cleanup_registry();
        return CU_get_error();
    }

    //make xml file
    CU_set_output_filename("CUnit");
    CU_list_tests_to_file();
    CU_automated_run_tests();

    //basic run
    CU_basic_set_mode(CU_BRM_VERBOSE);
    CU_basic_run_tests();

    CU_cleanup_registry();

    return CU_get_error();
}
```

CUnit header file include, run방식에 따라 다른
Test 대상 소스 include

Test cases

Test main()

Registry에 test suite 추가

Test case 예제

- ASSERT는 변수의 값이 예상과 같은지를 확인하는 용도

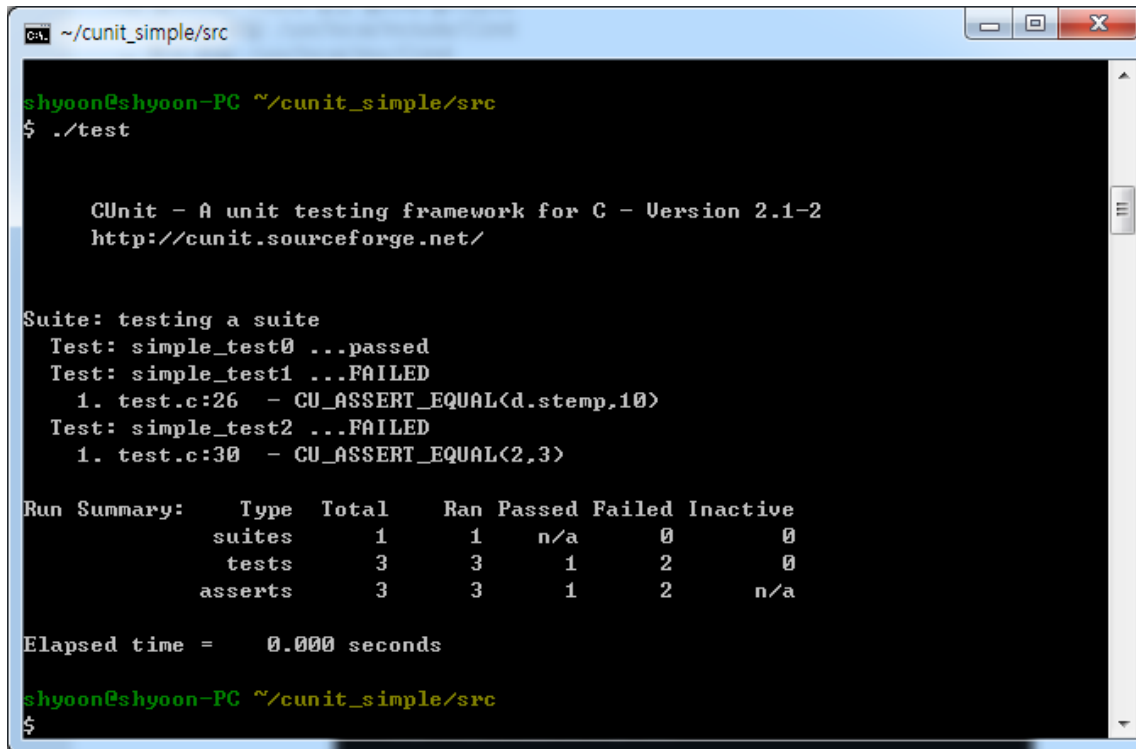
```
void simple_test_00(void){  
    //CU_ASSERT_STRING_EQUAL(result, "expected")  
    s.DoorState = T;  
    DoorSensorInterface(F);  
    CU_ASSERT_EQUAL(s.DoorState, F);  
}  
  
void simple_test_01(void){  
  
    d.stime = 0;  
    _10secIncrease(T);  
  
    CU_ASSERT_EQUAL(d.stemp, 10);  
}  
  
void simple_test_02(void){  
    CU_ASSERT_EQUAL(2,3);  
}
```

Test 하고자 하는 data 값 초기화
함수에 parameter 입력
함수 실행 후 예상 값 확인

http://cunit.sourceforge.net/doc/writing_tests.html#tests 참조

Test 실행

- Cygwin에 대상 소스와 test 소스를 넣고 아래를 입력
 - gcc test.c -o test.exe -L/usr/local/lib -lcunit (컴파일)
 - ./test



```
CA: ~/cunit_simple/src
shyoon@shyoon-PC ~/cunit_simple/src
$ ./test

CUnit - A unit testing framework for C - Version 2.1-2
http://cunit.sourceforge.net/

Suite: testing a suite
Test: simple_test0 ...passed
Test: simple_test1 ...FAILED
1. test.c:26 - CU_ASSERT_EQUAL(d.stemp,10)
Test: simple_test2 ...FAILED
1. test.c:30 - CU_ASSERT_EQUAL(2,3)

Run Summary:
  Type   Total   Ran   Passed   Failed   Inactive
  suites    1     1    n/a      0        0
  tests     3     3     1       2        0
  asserts   3     3     1       2    n/a

Elapsed time = 0.000 seconds

shyoon@shyoon-PC ~/cunit_simple/src
$
```

Basic mode의 실행결과를
볼 수 있음