

C언어 기반 CoAP 프로토콜 코드의 설치 및 실행 방법

경북대학교 컴퓨터학부
이희수, 박재언, 이재혁, 이봉석

2014. 10

- 목 차 -

1. 유닉스 개발환경 구축하기
2. 소스코드 다운로드 및 압축해제
3. 프로그램 수행환경 구성
4. 실행파일 생성
5. 서버와 클라이언트 수행

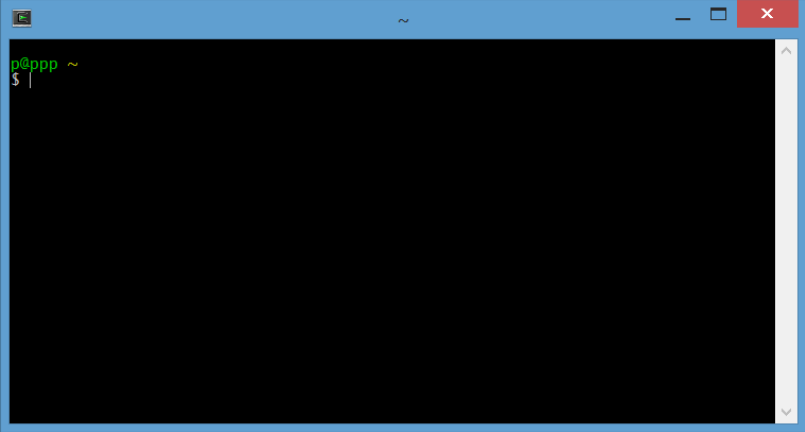
유닉스 환경 구축하기

- 소스코드가 유닉스 기반으로 코드가 작성되어 있어 윈도우 환경에서는 실행하기가 어려움.

- 유닉스 환경이라면 상관없으나 윈도우 환경이라면 유닉스 환경구축 후 진행해야 함.

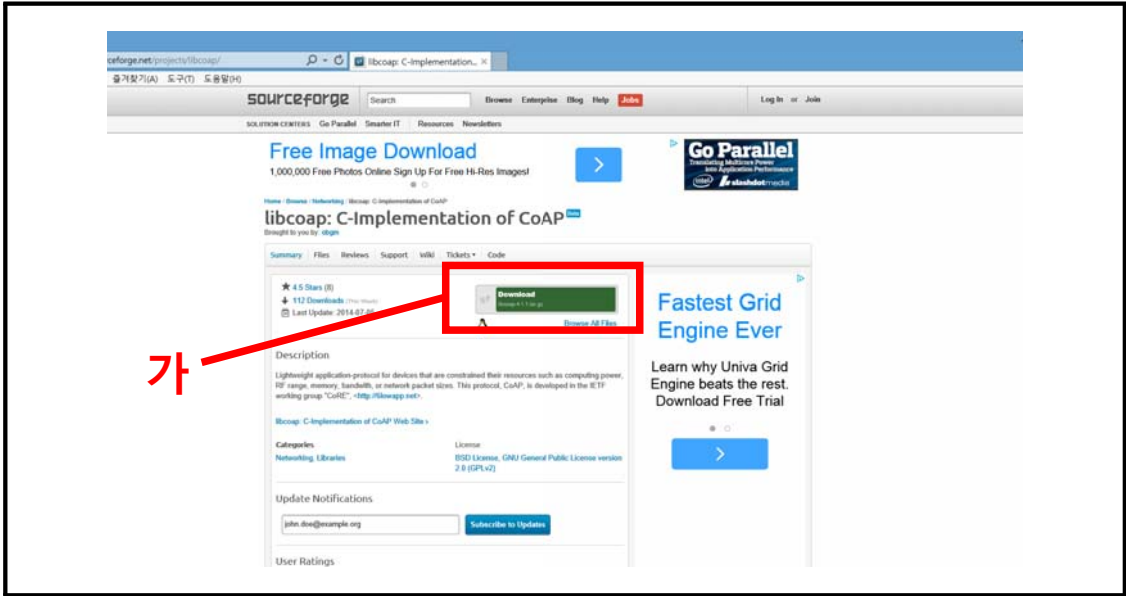
- 환경구축 방법
 - 상용화된 리눅스 OS 직접 설치
 - Virtualbox를 이용한 가상환경 구축
 - Cygwin 설치
 - 여러 방법으로 환경구축이 가능하나 윈도우 환경에서 쉽게 사용이 가능한 Cygwin으로 설명 진행

□ Cygwin 설치

1	<p>http://cygwin.com/install.html에 접속하여 Setup.exe파일을 다운로드하여 실행</p>
2	<p>Setup.exe 파일을 실행시킨 후 패키지 선택 시 Debug, Editors, Devel을 반드시 포함시킨 후 설치</p>
3	<p>설치완료 후 실행시키면 아래와 같이 실행됨.</p>  <p>The screenshot shows a terminal window with a blue title bar. The prompt is 'p@ppp ~' and the cursor is on the line '\$ '.</p>

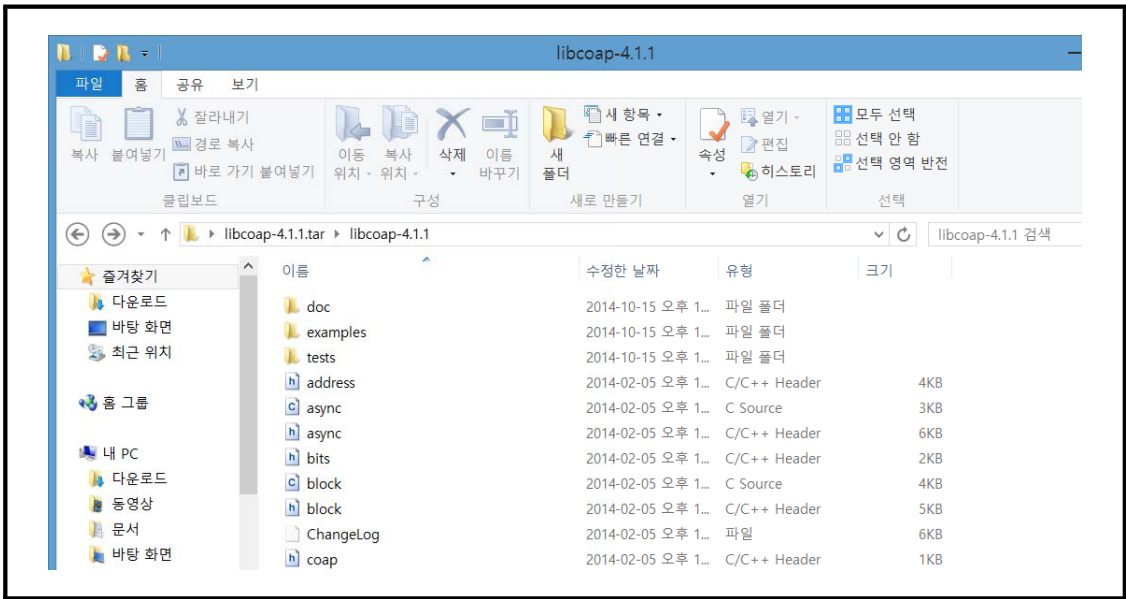
소스코드 다운로드 및 압축해제

□ 소스코드 다운로드



- <http://sourceforge.net/projects/libcoap/>에 접속하여 ‘가’ 부분을 클릭하여 소스코드를 다운

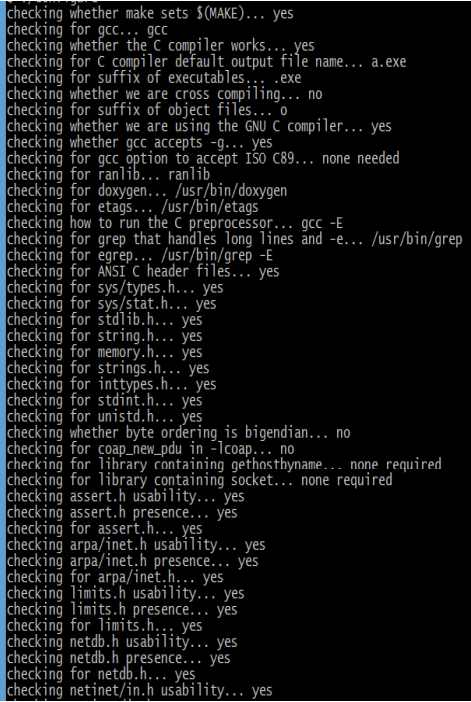
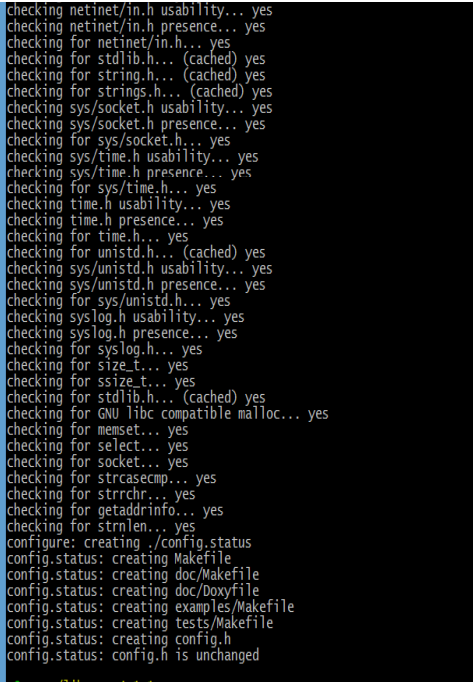
□ 압축해제



- 이후 진행을 위하여 최초 설정된 파일이름과 폴더순서는 변경하지 않고 압축을 해제

프로그램 수행환경 구성

- 개개인의 윈도우즈도 환경변수와 같은 것들의 설정이 다르듯 리눅스 환경도 동일하지 않음
- 다운받은 코드를 수행할 수 있도록 환경을 구성해야 함
- \libcoap-4.1.1.tar\libcoap-4.1.1 폴더에 있는 configure.exe를 실행하여 수행환경을 구성

1		
	위 그림과 같이 ./configure를 입력	
2		
	수행이 끝나면 위 그림과 같이 출력	


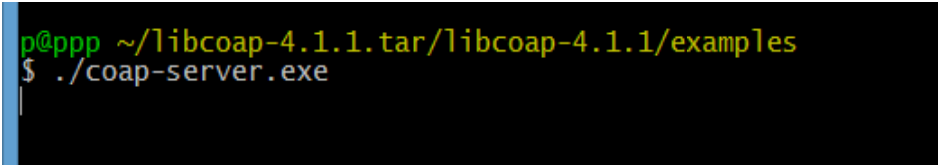
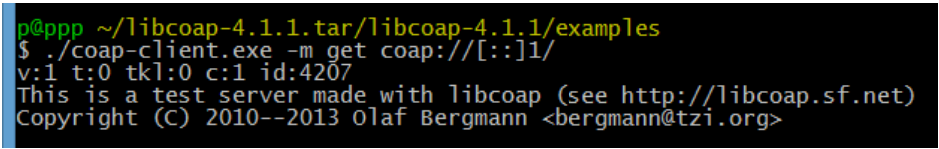
실행파일 생성

- 환경구성이 완료되면 실행파일을 생성해야 함
- make명령어를 사용하여 여러 c코드의 object파일 및 실행파일을 생성
- \libcoap-4.1.1.tar\libcoap-4.1.1 폴더에서 make명령어를 입력

1	<pre>p@ppp ~/libcoap-4.1.1.tar/libcoap-4.1.1 \$ make</pre>
	<p>위 그림처럼 make를 입력</p>
2	<pre>p@ppp ~/libcoap-4.1.1.tar/libcoap-4.1.1 \$ make cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o pdu.o pdu.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o net.o net.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o debug.o debug.c debug.c: In function 'coap_print_addr': debug.c:171:3: warning: implicit declaration of function 'strlen' [-Wimplicit-function-declaration] p += strlen((char *)p, len); ^ cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o encode.o encode.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o uri.o uri.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o coap_list.o coap_list.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o resource.o resource.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o hashkey.o hashkey.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o str.o str.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o option.o option.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o async.o async.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o subscribe.o subscribe.c cc -Wall -Wextra -std=c99 -pedantic -g -O2 -DWITH_POSIX -c -o block.o block.c ar cru libcoap.a pdu.o net.o debug.o encode.o uri.o coap_list.o resource.o hashkey.o str.o option.o as ync.o subscribe.o block.o ranlib libcoap.a make -C doc make[1]: Entering directory '/home/p/libcoap-4.1.1.tar/libcoap-4.1.1/doc' /usr/bin/doxygen Doxyfile >./doxygen.out 2>&1 make[1]: Leaving directory '/home/p/libcoap-4.1.1.tar/libcoap-4.1.1/doc' for dir in examples doc tests; do \ make -C \$dir ; \ done make[1]: Entering directory '/home/p/libcoap-4.1.1.tar/libcoap-4.1.1/examples' cc -g -Wall -g -O2 -I. -DWITH_POSIX -c -o server.o server.c cc -o coap-server server.o -L. -lcoap cc -g -Wall -g -O2 -I. -DWITH_POSIX -c -o client.o client.c client.c: In function 'cmdline_content_type': client.c:592:5: warning: array subscript has type 'char' [-Wchar-subscripts] if (isdigit(*q)) { ^ cc -o coap-client client.o -L. -lcoap cc -g -Wall -g -O2 -I. -DWITH_POSIX -L. rd.c -lcoap -o rd cc -g -Wall -g -O2 -I. -DWITH_POSIX -L. etsi_01.c -lcoap -o etsi_01 make[1]: Leaving directory '/home/p/libcoap-4.1.1.tar/libcoap-4.1.1/examples' make[1]: Entering directory '/home/p/libcoap-4.1.1.tar/libcoap-4.1.1/doc' /usr/bin/doxygen Doxyfile >./doxygen.out 2>&1 make[1]: Leaving directory '/home/p/libcoap-4.1.1.tar/libcoap-4.1.1/doc' make[1]: Entering directory '/home/p/libcoap-4.1.1.tar/libcoap-4.1.1/tests' make[1]: Nothing to be done for 'all'. make[1]: Leaving directory '/home/p/libcoap-4.1.1.tar/libcoap-4.1.1/tests'</pre>
	<p>수행이 끝나면 위 그림처럼 출력</p>

서버와 클라이언트 수행

- make 명령이 수행되고 나면 각종 object 파일과 coap-server.exe와 coap-client.exe 파일이 생성
- 수행 과정

1	
	<p>위 그림처럼 Wlibcoap-4.1.1.tarWlibcoap-4.1.1Wexamples 폴더에 coap-server.exe와 coap-client.exe 파일이 생성됨</p>
2	
	<p>./coap-server을 입력하여 서버를 수행</p>
3	
	<p>./coap-client -m get coap://[::1]/을 입력하여 클라이언트를 수행</p> <p>연결이 성공했을 때 This is a test server made with libcoap (see http://libcoap.sf.net) Copyright (C) 2010--2013 Olaf Bergmann <bergmann@tzi.org> 다음과 같은 메시지가 출력되고, 이것을 통해 서버와 클라이언트가 서로 연결이 되었음을 알 수 있다. (id는 수행시마다 변경됨)</p>