RTOS Compile 환경만들기

▲ Borland C++ 4.5에서 IDE환경으로 RTOS Compile하기

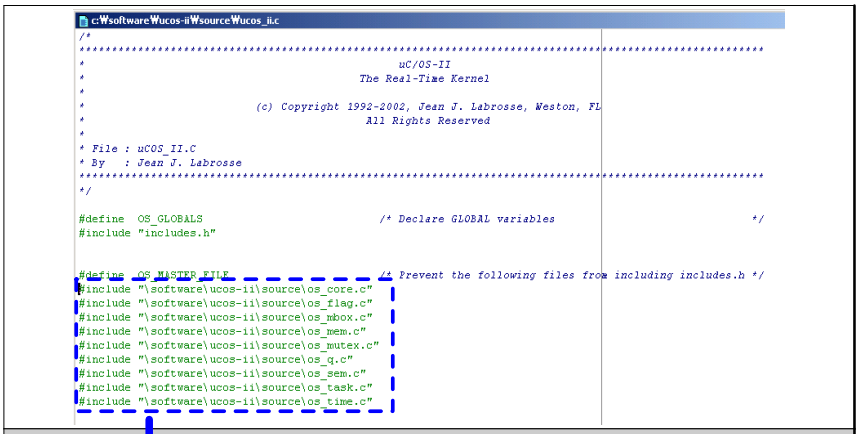
○ 1. 폴더 만들기

- C:\SOFTWARE\SAMPLE 이라는 폴더를 만든다

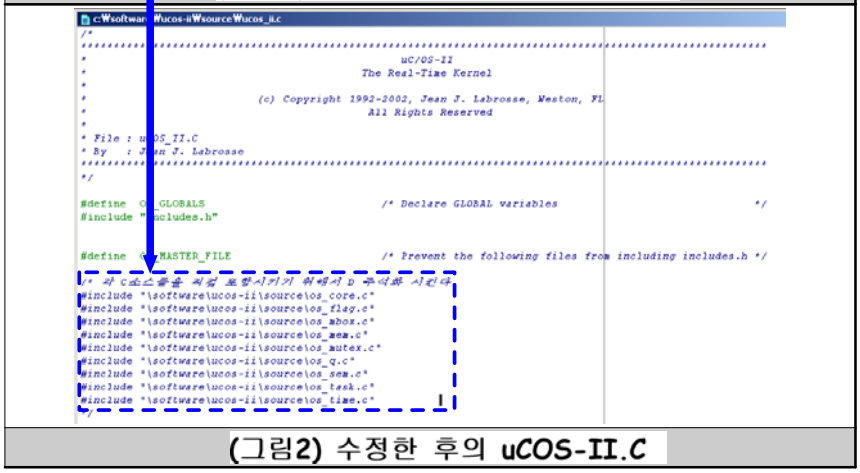
○ 2. 필요한 FILE들을 C:\SOFTWARE\SAMPLE밑으로 COPY한다

파일이 위치한 폴더	복사할 파일
C:\SOFTWARE\uCOS-II\EX1_x86L\BC45\SOURCE (* 예제가 바뀌면 폴더 EX1_x86L부분을 바꾼다)	TEST.C INCLUDES.H OS_CFG.H
C:\SOFTWARE\uCOS-II\Ix86L\BC45	OS_CPU.H OS_CPU_A.ASM OS_CPU_C.C
C:\SOFTWARE\uCOS-II\SOURCE\	uCOS_II.C uCOS_II.C
C:\SOFTWARE\BLOCKS\PC\BC45	PC.C PC.H

○ 3. C:\SOFTWARE\SAMPLE 밑으로 필요한 파일들을 복사한 후 각 환경을 잡아주기 위한 첫 번째 단계로 uCOS-II.C 파일안의 include문 부분들을 Default 시킨다



(그림1) 수정하기 전의 uCOS-II.C



(그림2) 수정한 후의 uCOS-II.C

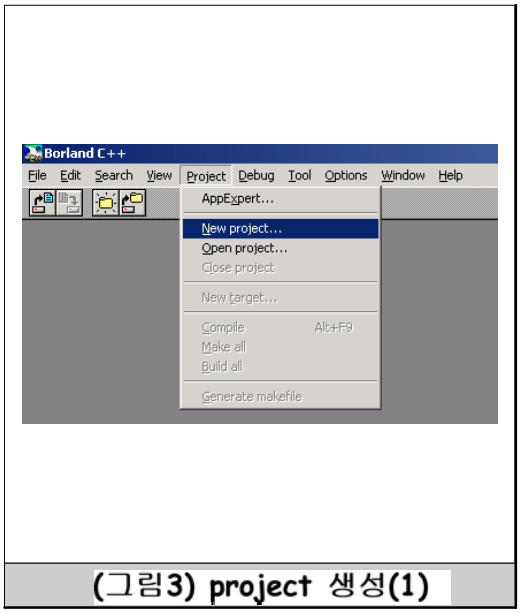
○ 4. uCOS-II.C 파일 안에 default시켰던 OS Dependent한 소스들을 하나씩 uCOS-II.C안에 복사해 붙여 넣는다

- 위의 점선 안의 C 소스들을 하나씩 열어서 uCOS-II.C안에 차례대로 복사한다

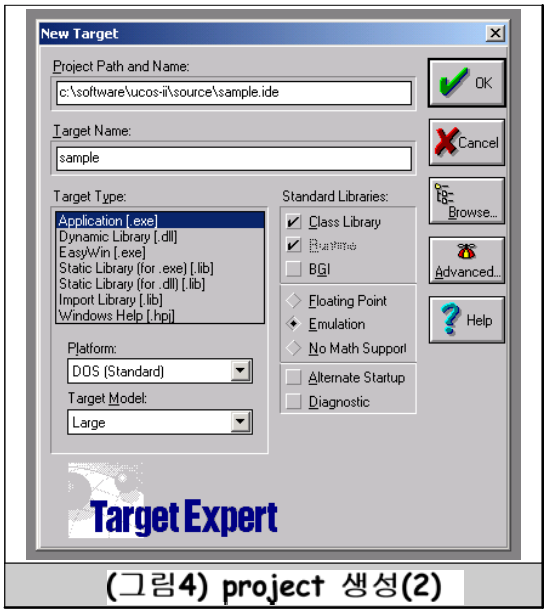
복사해야할 파일	비 고
OS_CORE.C	OS 동작 CORE에 대한 전체적인 함수 정의
OS_FLAG.C	CPU FLAG 및 PROCESS FLAG에 대한 정의
OS_MBOX.C	Message Box에 대한 함수 정의
OS_MEM.C	Memory관리에 대한 함수 정의
OS_MUTEX.C	uCOS-II mutex에 대한 정의
OS_Q.C	queue에 대한 함수 정의
OS_SEM.C	semaphore에 대한 함수 정의
OS_TASK.C	task에 대한 함수 정의

○ 5. Borland C에 대하여 Project에 생성을 실시한다.

- (1) Speedbar에서 Project → New project 생성

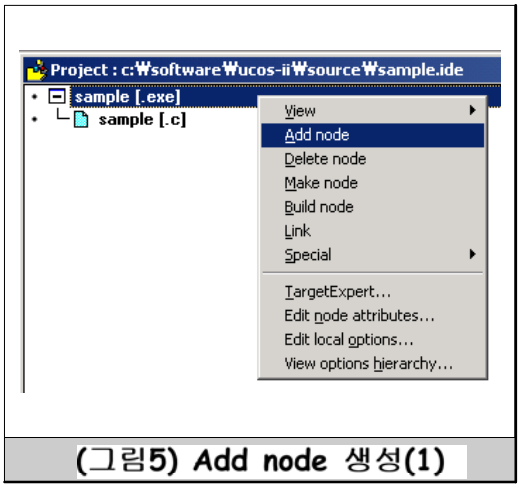


(그림3) project 생성(1)

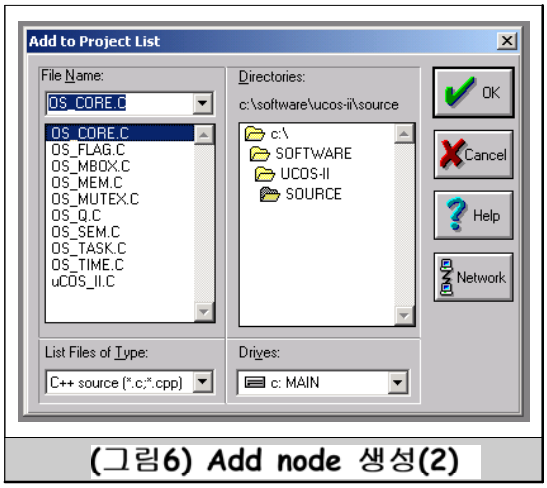


(그림4) project 생성(2)

- (2) Project가 생성되면 project에 RTOS Compile에 필요한 소스들(Nodes)을 Sketch Project 윈도우에서 Add node 메뉴를 이용해서 추가한다.



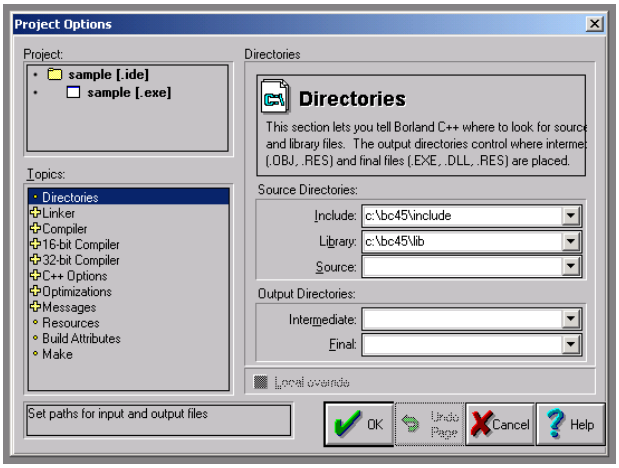
(그림5) Add node 생성(1)



(그림6) Add node 생성(2)

(*) 추가해야 할 소스파일 : os_cpu.c, pc.c, uCOS-II.c, test.c, os_cpu_a.asm

- (3) 소스파일 추가 후 지금 생성하고 있는 "sample.ide" 프로젝트에 대한 Option을 지정해 준다

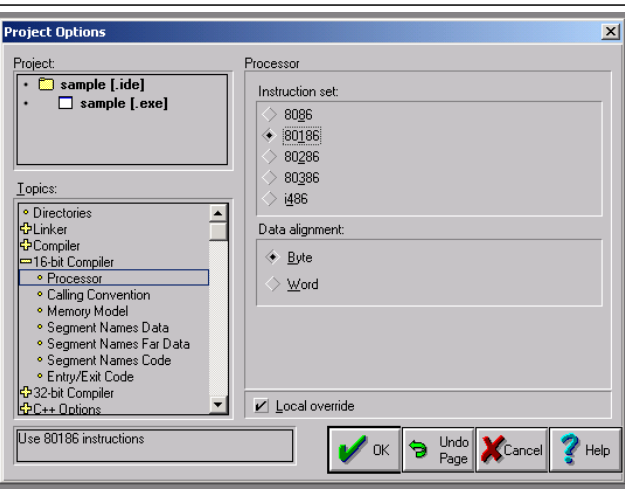


(그림7) project 환경 directory 설정

-project option을 선택하면 많은 하위 option들을 지정하여 다중적인 프로그램 개발시에 아주 편리하게 사용할 수 있는 환경이다.

-include와 library 디렉토리는 초기 상태를 유지한다

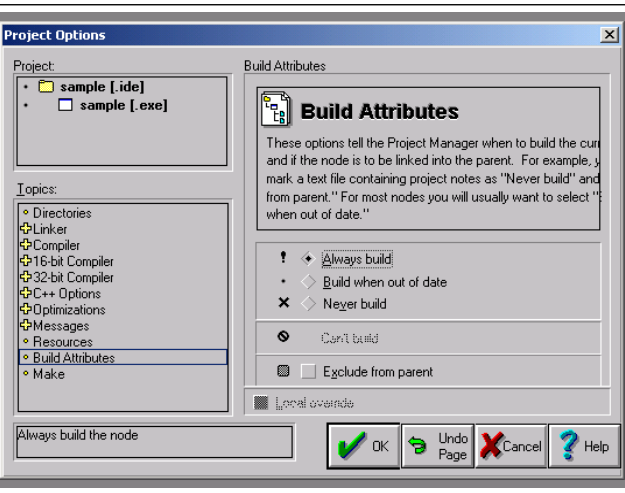
-만약 사용자가 사용하는, 그리고 프로그램에서 필요하는 파일들을 include 시키기 위해서는 include 메뉴에 ";"를 사용해서 추가한다
ex)c:\bc45\include;c:\rcy\include



(그림8) 16bit compiler환경설정

-16bit compiler 메뉴에서 우리는 80186을 상대로 포팅하기 때문에 명령어셋 (instruction set)을 80186으로 선택한다

-데이터 정렬방식은 8bit가 기본이 되는 Byte를 선택한다 (80186은 8bit Data addressing mode이다)

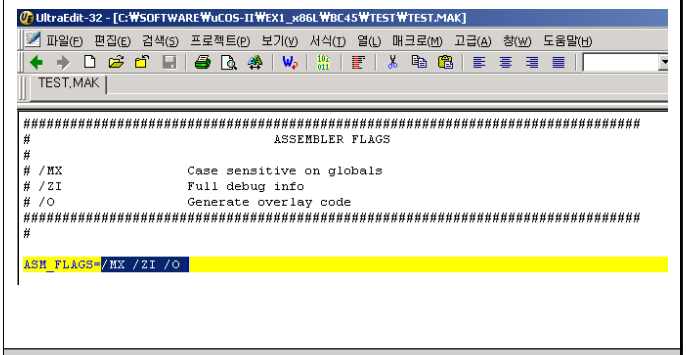


(그림9) build attributes설정

-Borland C++ IDE 환경에서 각 소스로서 Build를 하기 위해서는 언제나 Build 가능하도록 지정한다. 따라서 "Always build"를 선택한다

-선택후에 OK버튼을 눌러 "sample.ide"에 대한 IDE 환경을 저장한다

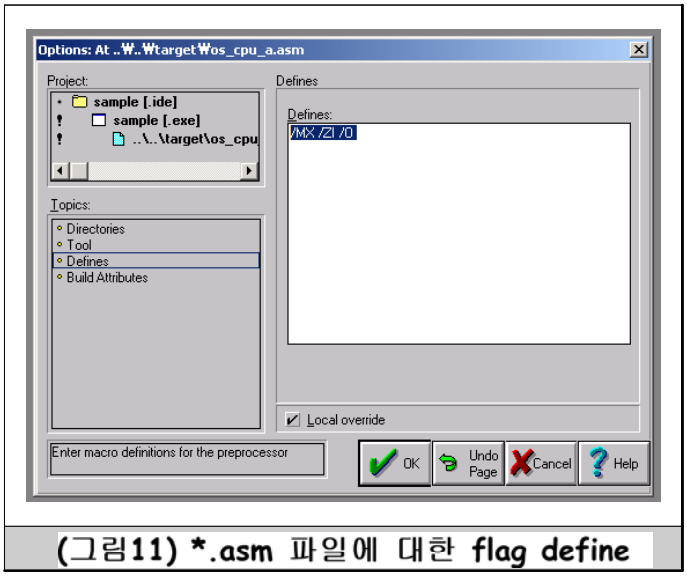
- (4) sample.ide에 대한 환경설정이 끝나면 이곳에 추가한 node들 (c와 asm으로 만든 소스코드들)에 대한 설정이 sample.ide와 동일하게 되어 있는지 확인해 준다. 그리고 asm파일에 대해서는 *.mak파일에 설정되어 있는 flag를 define 부분에 입력해 준다.



(그림10) test.mak파일의 asm flag 부분

-c:\software\uCOS-II\EX1_x86\bc45\test에 위치해 있는 test.mak파일을 주로 사용하는 editor로 열어보라. 그곳에는 uCOS-II를 위한 FLAG들과 설정이 잡혀있다.

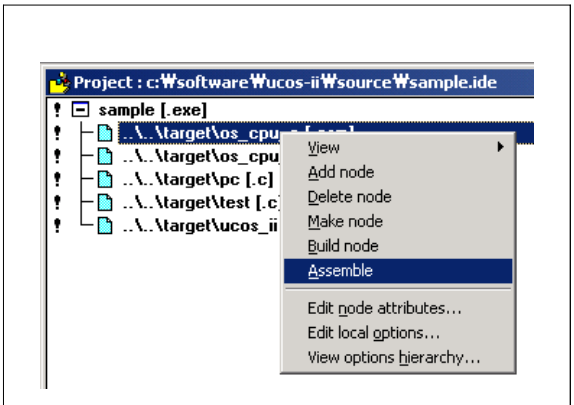
- MAKE TOOL에 대하여 궁금하다면 http://database.sarang.net/study/make/GNU-Make-1.html을 참고하기 바란다.



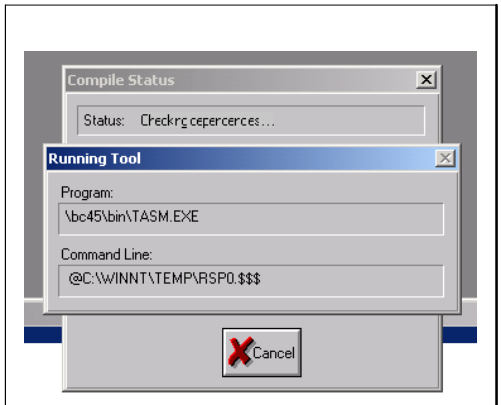
(그림11) *.asm 파일에 대한 flag define

-test.mak파일에서 asm flag 선언의 flag를 따온 후 define부분에 선언해 주면 make tool에서의 그것과 동일하게 동작한다.

- (5) 필요 node들을 project에 추가 시킨 후 os_cpu_a.asm파일부터 assemble을 시작한다. sketch project에 추가되어진 os_cpu_a.asm위에서 마우스 오른쪽 버튼을 눌러 assemble 메뉴를 선택하고 assemble한다.

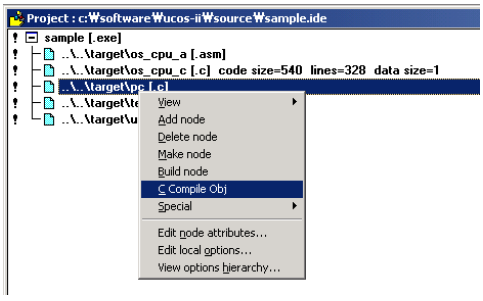


(그림12) sketch project에서 assemble



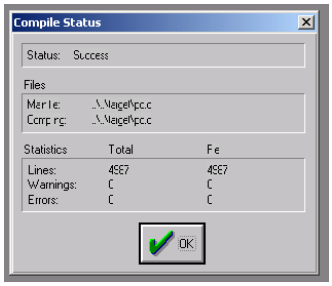
(그림13) assembling

- (6) os_cpu_a.asm 의 assemble이 끝나면 나머지 c소스코드로 된 node들에 대한 complie을 실행한다.



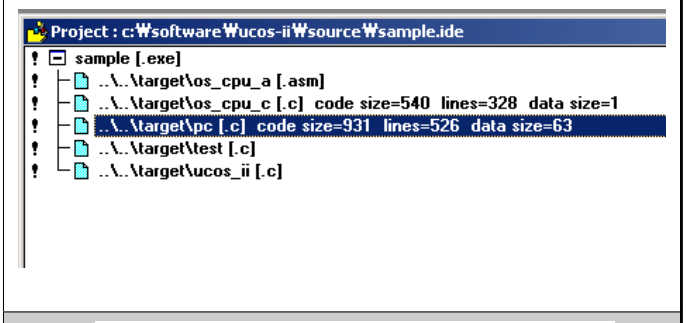
(그림14) sketch project에서 compile obj

-sketch project에서 sub메뉴를 이용하면 c소스마다 일일이 complile 하는 불편함을 없앨 수 있다.



(그림15) compile success

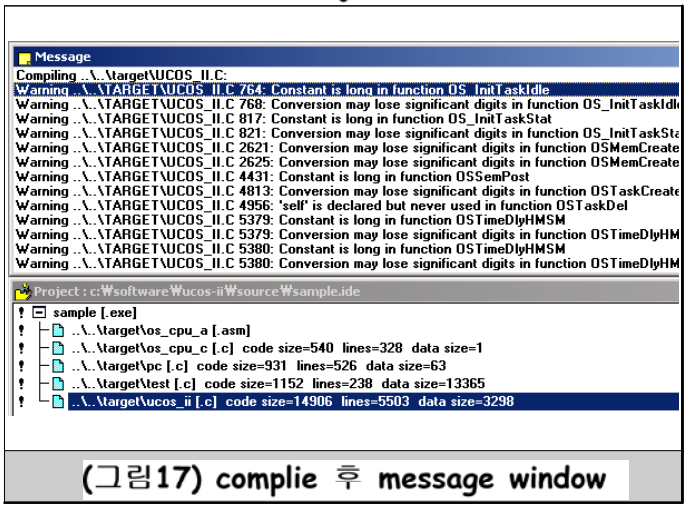
-compiling을 아주 빠르게 해주면서 최종적으로 compile success 메뉴를 띄워준다



(그림16) compile result를 "sketch"

-compile이 끝나면 compile 결과를 보기 좋게 sketch project (아마도 이렇게 간단하고 시안성이 좋게 표시해 주기 때문에 sketch라는 용어를 사용하지 않았나 생각된다.)에 compile 결과를 표시해 준다.

-code size, line, data size에 대한 내용이 표시되어 진다.

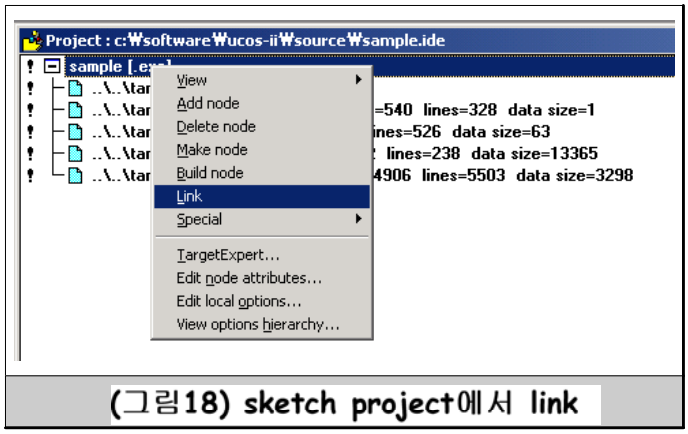


(그림17) complie 후 message window

-다른 c프로그래밍과 같이 compile이 끝나고 난 후에 warning 혹은 error를 표시해 주는 message window가 뜬다

-test.c와 ucos-ii.c파일 compile시 많은 warning이 뜨는데 일단 warning은 무시하고 error를 잡는데 총력을 기울인다.

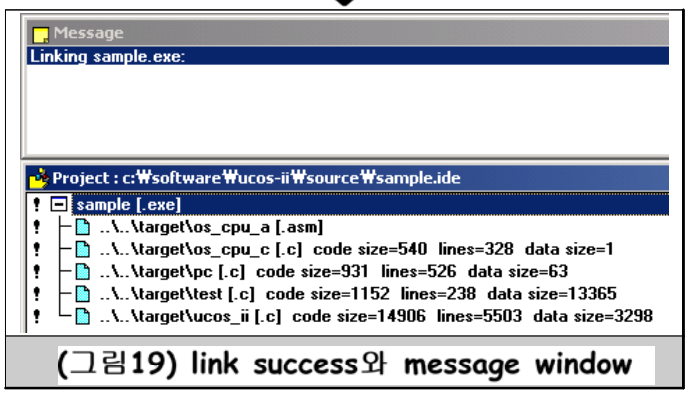
- (7) sample.ide에 add되어 있는 node들에 대한 assemble과 compile이 끝나면 실행하기 위한 excute파일을 만들기 위한 link를 실행한다.



(그림18) sketch project에서 link

-sketch project에서 sub메뉴를 이용하여 최종 excute파일을 생성하기 위해 link를 실행한다.

-sketch project를 이용하면 이와 같이 아주 간편하게 사용할 수 있다

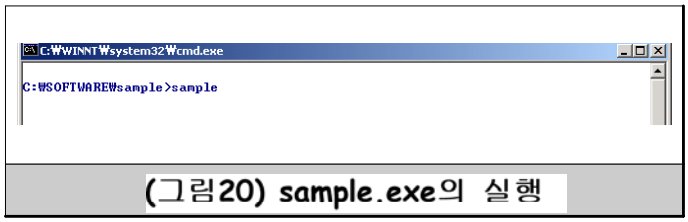


(그림19) link success와 message window

-link가 끝나면 success메뉴가 뜬 후에 message window에 왼쪽과 같은 메시지가 뜬다.

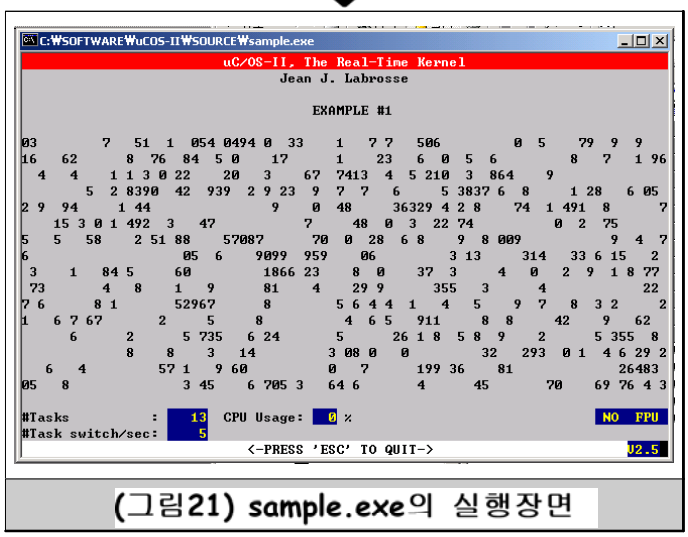
-이제 모든 과정이 끝났다. ide환경이 저장되어 있는 폴더로 이동하여 sample.exe를 실행하여보자

- (8) c:\software\sample\sample.exe를 dos command prompt화면에서 직접 실행하여 보자



(그림20) sample.exe의 실행

-dos command prompt 상태에서 직접 실행한다



(그림21) sample.exe의 실행장면

-uCOS-II의 기본적인 멀티태스킹 수행능력을 보여주는 예제이다.

-10개의 태스크가 화면상 임의 위치에 0에서 9사이의 숫자를 표시한다. 각 태스크는 한 숫자만 표시한다. 즉, 한태스크는 임의 위치에 0만 표시하고 다른 태스크는 1만 표시하는 식으로 예제 프로그램이 실행된다.