

2009년 06월 29일

Make Option들

Compiler Option만 Option이냐! 라고 친다면, make option도 잘 쓰면 엄청 편리한 용법들이 많이 있습니다. 뭐 많은 건 바라지도 않고, 몇 가지만 알아보도록 하세요. 이것들만 잘 알아도 엄청나게 요긴하게 쓰다니깐요.

㉠ -C<directory>

makefile을 계속 읽어서 실행하지 않고, directory로 이동하거라. Change Directory가 그 의미이며, 현재 make에서 특정 target에서 다른 make file을 실행하고 싶을 때 이런 수법을 많이 씁니다.

㉡ -h

으흐흐 헬프.

㉢ -f

make는 실행하게 되면 현재 directory에서 확장자 없는 Makefile 이라는 file을 default로 실행하게 되는데, 특정 file을 -f option을 주어서 실행할 수 있어요. make -f 파일이름.mak 이런 식으로 option을 주면 Makefile 이외의 다른 파일을 실행할 수 있는 거죠.

㉣ -r

make 내부에 미리 정의되어 있는 내장규칙을 모두 없애고 새 삶을 시작하라. 깨끗하게요.

㉤ -t

touch 기능으로 file 생성날짜를 현재 시간으로 바꿔 주며, 이 기능을 이용하면, touch를 해버리니까 수정하지도 않은 파일을 다시 컴파일 시킬 수 있습니다.

㉥ -s

보통 make를 실행하면 엄청난 양의 message를 echo에 의해서 뿌리게 되죠.
silence로서 shut up 하라는 의미입니다.

㉦ -v

make version을 출력합니다.

㉧ -p

내부적인 setting값을 출력합니다.

㉨ -k, -S

k는 keep going, make 실행 중 문제가 생기더라도 닥치고 계속 해라는 뜻이에요. S는 Stop으로서 k를 turn off합니다. 그러니까, S를 주게 되면 k를 무력화하는 거죠.

㉩ -d

Debugging Message를 다 출력해라 라는 option인데 이거 주면 엄청나게 많은 양의 메시지가 뿌려집니다. -d option을 주고 나면 화면에 뿌리지 말고, txt file로 redirect해서 나중에 text editor등으로 보는 게 낫습니다. txt file로 만드는 방법은 redirect를 이용하여, make -d >> debug.txt 이런 식으로 하는 거는 아시겠죠? > 는 원래 파일이 있었다면 내용을 지우고 지금 내용으로 update하는 거구요, >> 는 원래 파일이 있었다면 그 내용의 뒤에 덧붙이는 append 형식이에요. 뭐, 대충 어떻게 돌아가는 구나 정도 debugging하시려면 괜찮은데, 엄청난 양의 메시지를 감내 하실 수 있으면 좋겠어요 ㄹ 차라리 저는 앞에서 얘기한 테크닉4를 이용해서 printf 하듯이 디버깅 하는 게 더 마음 편하더라고요. 개인적인 기호 차이니까, 딱지 걸기 없기.

헉헉.. 겨우 소프트웨어 데꾸빠쥬를 끝냈사와요.

깁니다 길어~

내일부터는 ARM 제어의 세계로 빠져 볼 것이에요~ 안전벨트 단단히 메시라고요.

[option](#), [make](#), [compiler](#)

Linked at

at 2009/06/29 20:57

... bsp;컴파일을 더더더 쉽게. MACRO와 SUFFIX ⑤ 조금 더 Make 테크닉들 [④ Make option들](#) 4) ARM 미장센 - ARM 제어의 구현

③ ARM Assemblby를 파헤쳐 보자 ADS VS GN ... [more](#)

☞ ☞