

2009년 06월 15일

Lib을 까보자~

우리 흔히들 Project를 하다 보면, 3rd party에서 준 lib을 이용하여 Project를 진행할 때가 흔히 있죠. 그러나! 이런 3rd party가 준 lib이 제대로 만들어 지지 않아, link error를 낼 때는 정말 답답하기 그지 없기 마련이지요. Library를 다 까보고 싶지만 참내.

하지만, 걱정 마세요. 이제부터 library를 한번 까볼까 합니다. Lib을 어떻게 까보냐고요? 별거 아닙니다. 우리 Lib 만들 때 월로 만들었나요. archive이용했지요? armar이라는 걸 이용해서 archive를 만들었어요. 그 말인 즉슨, armar을 이용해서 다시 까볼 수도 있다는 말이에요. 얼마나 행복한 순간인가요. 우리 archive의 세계를 여행해 BoA요.

그러면, armar이 어떤 일을 해줄 수 있을 지 armar의 option들을 한번 보지요. command창에 armar이라고 치면 option이 주루룩 나와요.

```
ARM Archiver, ADS1.2 [Build 805]      - archive creation and maintenance tool
Command format:
armar options archive [ file_list ]
wildcards '?' and '*' may be used in file_list
Options:-
-r          Insert files in <file_list>, replace existing members of the same name.
-d          Delete the members in <file_list>.
-x          Extract members in <file_list> placing in files of the same name.
-m          Move files in <file_list>.
-p          Print files to stdout.
-a pos      Insert/move files after file named <pos>.
-b pos      Insert/move files before file named <pos>.
-u          Update older files only, used with -r.
-n          Do not add a symbol table to an object archive.
-s          Force regeneration of archive symbol table.
-t          Print table of contents of archive.
-zs         Show the symbol table.
-zt         Summarize the archive contents (sizes + entries).
-c          Suppress warning when a new archive is created.
-C          Do not overwrite existing files when extracting.
-T          Truncate file names to system maximum length.
-v          Give verbose output.
-create     Force creation of a new archive.
-via file   Take additional arguments from via file.
-sizes      List the size of each member and the library total.
-entries    List sections containing ENTRY points.
-vsn        Print the current Armar Version.
-help       Print this message.
Examples:-
armar -r mylib.a obj1 obj2 obj3...
armar -x mylib.a ?sort*
armar -d mylib.a hash.o
armar -tv ansilib.a
```

좌라라락. option들하고 Example까지 정말 친절하지 아니 않을 수 없지요. 너무 자세한 것들은 빼고요. Example에 나온 몇 가지만 둘러보지요.

armar -r 명령을 주면 이전에 보았듯이, archive (Lib)을 만들어 줍니다.

armar -x 명령을 주면 archive안에서 원하는 것들을 빼낼 수 있지요.

armar -d 명령을 주면 archive안에서 object를 제거할 수 있고요,

armar -tv 이렇게 명령을 여러 개 줄 수도 있는데, t는 object 이름들을 보여주고요, v까지 주면 말 많은 버전으로서, 각 object의 offset과 날짜 뭐 이런 정보까지 보여준답니다.

그러면, 제일 빠른 방법이 ARM standard library 한번 까보면 제일 빠르겠네요.

ADS directory\WLib\Wmlib> 요기에 가면, ARM standard library가 많이 있어요. dir해 보면, 야호~ 이렇게 많은 Lib들이 있네요.

ADS12\Lib\armlib 디렉터리

```
[.]          [...]      c_a__se.b  c_a__se.l  c_a__sn.b  c_a__sn.l c_a__ue.b  c_a__ue.l
c_a__un.b   c_a__un.l   c_t__se.b  c_t__se.l c_t__sn.b  c_t__sn.l c_t__ue.b  c_t__ue.l
c_t__un.b   c_t__un.l f_a.b       f_a.l     f_afm.b   f_afm.l   f_avp.b    f_avp.l f_a_m.b
f_a_m.l     f_a_p.b    f_a_p.l    f_t.b     f_t.l     f_tfm.b   f_tfm.l    f_tvp.b  f_tvp.l
f_t_m.b     f_t_m.l f_t_p.b    f_t_p.l    g_afm.b   g_afm.l   g_avp.b    g_avp.l g_a_m.b
g_a_m.l     g_a_p.b    g_a_p.l    g_tfm.b   g_tfm.l   g_tvp.b   g_tvp.l    g_t_m.b  g_t_m.l
g_t_p.b     g_t_p.l m_afms.b   m_afms.l   m_afmu.b  m_afmu.l   m_avps.b   m_avps.l m_avpu.b
m_avpu.l    m_a_ms.b   m_a_ms.l   m_a_mu.b  m_a_mu.l  m_a_ps.b  m_a_ps.l   m_a_pu.b
m_a_pu.l    m_tfms.b   m_tfms.l   m_tfm.b   m_tfm.l   m_tvps.b  m_tvps.l   m_tvpu.b
m_tvpu.l    m_t_ms.b   m_t_ms.l   m_t_mu.b  m_t_mu.l   m_t_ps.b  m_t_ps.l   m_t_pu.b  m_t_pu.l
```

이중에 제일 만만한 f_t.b를 한번 까 볼 게요. (크기가 제일 작습니다.) 여기 보면 t와 a가 있는데 t는 thumb standard library이고요, a는 ARM standard library예요. ARM과 Thumb 의 차이에 대한 자세한 내용은 ARM 구현 쪽에서 자세히 보시기로 해요.

여하튼 f_t.b에는 어떤 것들이 있을까. 까보죠 머.

ADS12\WLib\Wmlib>armar -t f_t.bfpinit.osetjmp.oprintf1.oprintf2.oscanf1.oscanf2.o

어랏, 어디서 많이 보던 녀들이 들어앉아 있네요. 그러면 이 object들에는 어떤 symbol들이 들어 앉아 있나 궁금하지요? Symbol을 보는 건 -zs option으로 가능해 보여요.

armar -zs f_t.b

ADS12\Lib\armlib>armar -zs f_t.b__16_fp_init

```
from fpinit.o  at offset  400_fp_init
from fpinit.o  at offset  400__16setjmp
from setjmp.o  at offset  1336setjmp
from setjmp.o  at offset  1336__16longjmp
from setjmp.o  at offset  1336longjmp
from setjmp.o  at offset  1336__lib_sel_fp_printf
from printf1.o at offset  2666$Sub$$__16_fp_display_gate
from printf1.o at offset  2666$Sub$$_fp_display_gate
from printf1.o at offset  2666__16_fp_display_gate
from printf2.o at offset  3756_fp_display_gate
from printf2.o at offset  3756__lib_sel_fp_scanf
from scanf1.o  at offset  4798$Sub$$__16_fp_read_gate
from scanf1.o  at offset  4798$Sub$$_fp_read_gate
from scanf1.o  at offset  4798__16_fp_read_gate
from scanf2.o  at offset  5882_fp_read_gate
from scanf2.o  at offset  5882
```

오호, 각각의 object에는 이런 symbol들이 들어 있군요! 으하하. 이런 식이라면, 분명 어느 object에 어떤 symbol이 들어 있는지 제대로 알 수 있겠네요. 이런 symbol들만 찾을 수 있다면, link문제를 일으키는 symbol을 찾아 수정할 수도 있지요.

또는 library를 내 맘대로 수정해서 녀석으로 바꿔 칠 수도 있는 거예요. armar -d optoin을 이용해서 특정 object를 삭제해 버리고, 그 object안에 있던 symbol들을 모두 다시 구현하여,
armar -c option으로 다시 그 archive에 새로운 object를 넣어주면 수정된 archive가 될 수 있겠지요.

예를 들어, 위의 printf2.o를 제거해 버리고,
armar -d f_t.b printf2.o
printf2.c에다가
__16_fp_display_gate, f_fp_display_gate

요 두개 함수를 다시 구현해서 tcc -c로 compiler한 후, (Thumb mode library니까) armar -c f_t.b printf2.o 로 넣는다면 완전 내용이 바뀐 나의 library가 탄생하는 것이지요. 또는 lib을 잘 까 보다 보면, 의외의 수확의 함수들을 발견할 수 있을지도 모릅니다. 와하하.



실은 난데없이 Symbol이니 뭐니 하는 말을 늘어 놓아서 정말 죄송합니다. 하지만, 관심이 있는 분이라면 Symbol은 전역변수의 이름이나, 함수의 이름따위를 의미한다는 걸 눈치 채셨겠지요? 조금 더 참고 기다려 주시면, Symbol의 정체를 까발기는 이야기가 준비되어 있으니깐 조금만 참아주시면 정말 감사하겠습니다.

[lib](#), [library](#), [라이브러리](#), [아카이브](#)

Linked at at 2009/10/01 12:32

... [Lib을 까보자](#) ... [more](#)

Commented by [organizer™](#) at 2009/06/16 00:03

안 그래도 LIB만 딸랑 넘겨줬을 때 일이 잘못되면 어떻게 하나 하고 걱정(?)했었는데, 다 방법이 있군요^^ ;

본문 중의

>> 우리 archive의 세계를 여행해 BoA요.

매우 친숙합니다... <-- 어디서 봤을까요? ㅋㅋ ;

Commented by [히연](#) at 2009/06/16 11:50

ㅋ~ Lib만 딸랑 넘겨줄 때 잘못된 Lib은 박살내 BoA요~ 으흐흐

Commented by 식이.. at 2009/06/16 14:25

이거 너무 유용한데요. ^^

근데 매일 하나씩 강의하나요?

그냥 전부 다 열어주심 안돼요? ㅌ.ㅌ

빨랑 봐야해서요.

Commented by [히연](#) at 2009/06/16 20:13

으흐흐~ 너무 고맙게 봐주셔서 감사합니다~ 자주자주 오세요~

Commented by 감사요 at 2009/06/23 15:56

하루에 5편 이상씩 보고 있네요..

쉽고 재밌게 설명해 주셔서 감사합니다.

Commented by [히연](#) at 2009/06/23 22:36

오호호~ 너무 빨리 보시는거 아니에요~? 아무래도 저두 다시 한번 더 잘 읽어 봐야 하겠어요~

Commented by 주니 at 2009/07/09 11:25

지금 저에게 필요한 내용이 참 많아요 ^^ 감사

Commented by [히연](#) at 2009/07/09 22:58

지금 저에게 필요한 건? 스피드~