

2020. 4. 23

HW #2 – binutils 실습


2015722087 컴퓨터정보공학부

김민철

-ar

이 코드는 자주 사용하던 tar 명령어와 비슷하다. 정적 라이브러리를 만드는 데 사용하는 도구이며, archive 파일을 만든다. 실습에서는 일반적인 파일 두개를 묶어 만들었다.

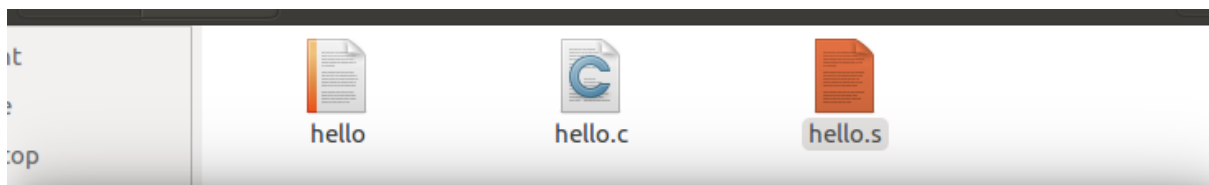
```
os2015722087@ubuntu:~/embedded$ arm-linux-gnueabi-ar r ar_file.a hello.c hello.s
arm-linux-gnueabi-ar: creating ar_file.a
os2015722087@ubuntu:~/embedded$
```



두개의 파일 묶어 묶음 파일이 생성되었다. 압축은 되지 않았지만 여기에 압축 기능이 더해진다면 .tar.gz 와 동일한 기능을 수행하는 것이다.

-as

이 코드는 어셈블리이다. 어셈블리어를 오브젝트 코드로 해석해준다.



```
os2015722087@ubuntu: ~/embedded
os2015722087@ubuntu:~/embedded$ arm-linux-gnueabi-as hello.s -o hello
os2015722087@ubuntu:~/embedded$ file ./hello
./hello: ELF 32-bit LSB relocatable, ARM, EABI5 version 1 (SYSV), not stripped
os2015722087@ubuntu:~/embedded$
os2015722087@ubuntu:~/embedded$
```

처음 사용해본 명령어이지만 일반적인 컴파일러와 동일한 기능을 수행하는 것 같은 느낌이었다.

-g++

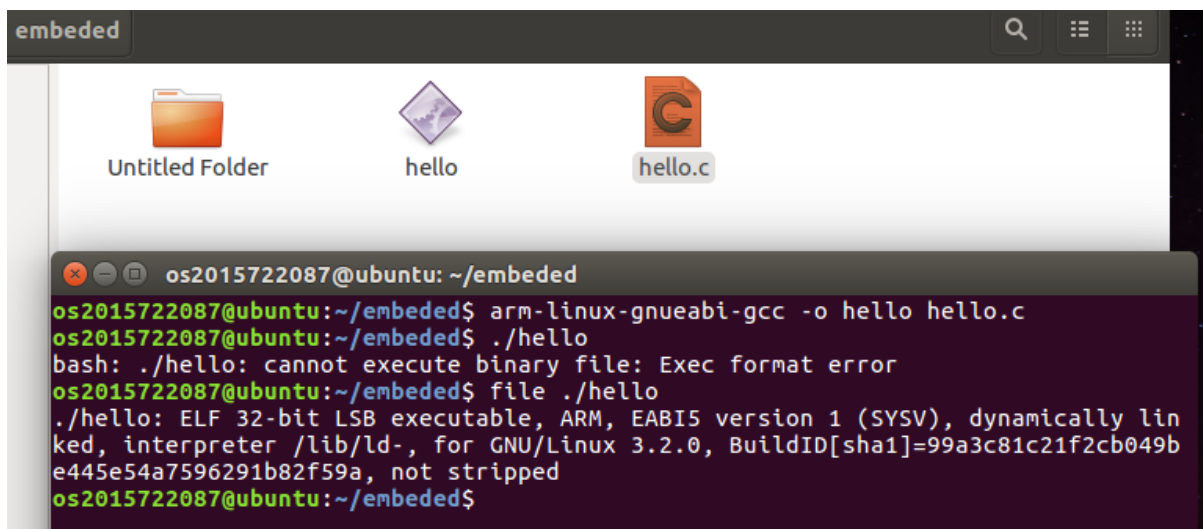
C++ 컴파일러이다.

```
os2015722087@ubuntu:~/embedded$ arm-linux-gnueabi-g++ hello.c
os2015722087@ubuntu:~/embedded$ ./a.out
bash: ./a.out: cannot execute binary file: Exec format error
os2015722087@ubuntu:~/embedded$ file ./a.out
./a.out: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-, for GNU/Linux 3.2.0, BuildID[sha1]=5031ba40b89bbde24f5163b37317ad099b2d31b0, not stripped
```

.c 파일 혹은 .cpp 파일을 컴파일한다.

-gcc

C언어용 컴파일러이나 최근엔 다양한 언어를 지원한다.



The screenshot shows a file manager window titled 'embedded' with three items: 'Untitled Folder', 'hello', and 'hello.c'. Below it, a terminal window shows the following commands and output:

```
os2015722087@ubuntu: ~/embedded
os2015722087@ubuntu:~/embedded$ arm-linux-gnueabi-gcc -o hello hello.c
os2015722087@ubuntu:~/embedded$ ./hello
bash: ./hello: cannot execute binary file: Exec format error
os2015722087@ubuntu:~/embedded$ file ./hello
./hello: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-, for GNU/Linux 3.2.0, BuildID[sha1]=99a3c81c21f2cb049be445e54a7596291b82f59a, not stripped
os2015722087@ubuntu:~/embedded$
```

컴파일을 수행한다.

-ld

링커이다. 위에서 설명한 컴파일러가 추출한 오브젝트 파일을 단일 실행 프로그램으로 병합한다.

```
os2015722087@ubuntu: ~/embeded
os2015722087@ubuntu:~/embeded$ arm-linux-gnueabi-gcc -o hello.o hello.c
os2015722087@ubuntu:~/embeded$ arm-linux-gnueabi-ld hello.o
os2015722087@ubuntu:~/embeded$ arm-linux-gnueabi-ld hello.o -o hello.exe
os2015722087@ubuntu:~/embeded$ ./hello.exe
bash: ./hello.exe: cannot execute binary file: Exec format error
os2015722087@ubuntu:~/embeded$ file ./hello.exe
./hello.exe: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically
linked, interpreter /lib/ld-, for GNU/Linux 3.2.0, BuildID[sha1]=99a3c81c21f2cb
049be445e54a7596291b82f59a, not stripped
os2015722087@ubuntu:~/embeded$
```

많이 쓰던 기능이라 특별히 다른 것은 없는 것 같다.

-nm

오브젝트 파일에 포함된 심볼의 리스트를 확인하는 명령어이다.

```
os2015722087@ubuntu:~/embeded$ arm-linux-gnueabi-nm hello.o
                 U abort@@GLIBC_2.4
0002102c B __bss_end__
0002102c B __bss_end__
00021028 B __bss_start
00021028 B __bss_start__
0001034c t call_weak_fn
00021028 b completed.9905
00021020 D __data_start
00021020 W data_start
00010370 t deregister_tm_clones
000103d8 t __do_global_dtors_aux
00020f10 t __do_global_dtors_aux_fini_array_entry
00021024 D __dso_handle
00020f18 d _DYNAMIC
00021028 D _edata
0002102c B end
```

결과를 해석하기 위해서는 심볼 클래스의 의미를 이해해야한다. 몇 개의 클래스가 존재하지만 nm에서는 두번째 열처럼 한 문자로 표현한다.

-objcopy

오브젝트 파일을 다른 오브젝트 파일로 복사할 때 사용하며, 선택적으로 필요한 부분만 복사하여 파일의 사이즈를 줄이는데 사용하거나, 바이너리 포맷을 바꿀 때 사용한다.

```
os2015722087@ubuntu:~/embeded$ arm-linux-gnueabi-objcopy hello.o objcopy
os2015722087@ubuntu:~/embeded$ file ./objcopy
./objcopy: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically l
inked, interpreter /lib/ld-, for GNU/Linux 3.2.0, BuildID[sha1]=99a3c81c21f2cb04
9be445e54a7596291b82f59a, not stripped
os2015722087@ubuntu:~/embeded$
```

간단한 테스트 코드로하여 복사되는 것을 확인하였다.

-objdump

역어셈블할 때 사용하는 명령어이다. 오브젝트 파일에 대한 정보를 출력한다. 주로 컴파일 도구를 만드는 프로그래머들이 사용한다.

```
os2015722087@ubuntu:~/embeded$ arm-linux-gnueabi-objdump -s hello.o

hello.o:      file format elf32-littlearm

Contents of section .interp:
 10154 2f6c6962 2f6c642d 6c696e75 782e736f  /lib/ld-linux.so
 10164 2e3300
Contents of section .note.ABI-tag:
 10168 04000000 10000000 01000000 474e5500  ....GNU.
 10178 00000000 03000000 02000000 00000000  ....
Contents of section .note.gnu.build-id:
 10188 04000000 14000000 03000000 474e5500  ....GNU.
 10198 99a3c81c 21f2cb04 9be445e5 4a759629  ....!....E.Ju.)
 101a8 1b82f59a
Contents of section .gnu.hash:
 101ac 03000000 02000000 01000000 05000000  ....
 101bc 00480221 00000000 02000000 04000000  .H.!.....
 101cc b82b6b15 7ded110f 2f4e3df6  ....+k.}.../N=.
Contents of section .dynsym:
 101d8 00000000 00000000 00000000 00000000
```

굉장히 많은 내용이 출력되었으나 해석할 수는 없었다.

-readelf

다른 툴을 사용하지 않고도 ELF 파일을 직접 읽기 위한 명령어이다. 하나 이상의 옵션을 사용해야 하며 BFD를 경유하지 않고 파일을 읽기 때문에 objdump보다 상세한 정보를 얻을 수 있다.

```
os2015722087@ubuntu:~/embeded$ arm-linux-gnueabi-readelf -s hello.exe

Symbol table '.dynsym' contains 5 entries:
  Num:      Value              Size Type      Bind   Vis      Ndx Name
   0: 00000000                0 NOTYPE    LOCAL   DEFAULT UND readelf: Error: Reading 0x10
bytes extends past end of file for version need aux (3)
readelf: Error: Reading 0x10 bytes extends past end of file for version need

   1: 00000000                0 NOTYPE    WEAK    DEFAULT UND __gmon_start__
   2: 00000000                0 FUNC      GLOBAL  DEFAULT UND printfreadelf: Error: Reading
0x10 bytes extends past end of file for version need aux (3)
readelf: Error: Reading 0x10 bytes extends past end of file for version need

   3: 00000000                0 FUNC      GLOBAL  DEFAULT UND abortreadelf: Error: Reading
0x10 bytes extends past end of file for version need aux (3)
readelf: Error: Reading 0x10 bytes extends past end of file for version need

   4: 00000000                0 FUNC      GLOBAL  DEFAULT UND __libc_start_mainreadelf: Err
or: Reading 0x10 bytes extends past end of file for version need aux (3)
readelf: Error: Reading 0x10 bytes extends past end of file for version need

Symbol table '.symtab' contains 121 entries:
  Num:      Value              Size Type      Bind   Vis      Ndx Name
   0: 00000000                0 NOTYPE    LOCAL   DEFAULT UND
   1: 00010114                0 SECTION  LOCAL   DEFAULT    1
   2: 00010128                0 SECTION  LOCAL   DEFAULT    2
   3: 00010148                0 SECTION  LOCAL   DEFAULT    3
```

역시 많은 내용이 출력되었으나 해석할 수 없었다.

-size

파일의 섹션별 크기를 출력해주는 명령어이다.

```
os2015722087@ubuntu:~/embeded$ arm-linux-gnueabi-size hello
text      data      bss       dec       hex filename
105        0         0         105       69 hello
```

크기를 상세하게 확인할 때 사용할 수 있을 것 같은 명령어이다.

-strings

파일 내에서 출력할 수 있는 모든 문자열을 출력하는 명령어이다.

```
os2015722087@ubuntu:~/embeded$ arm-linux-gnueabi-strings hello
Number : %d Hello World
GCC: (Ubuntu/Linaro 5.4.0-6ubuntu1~16.04.9) 5.4.0 20160609
aeabi
hello.c
main
printf
.symtab
.strtab
.shstrtab
.rel.text
.data
.bss
.rodata
.comment
.note.GNU-stack
.ARM.attributes
```

단순하다. 파일이 매우 긴 경우에는 사용하지 않을 명령어일 것 같다.

-strip

오브젝트 파일에 있는 심볼을 삭제하는 명령어이다. 일반적으로는 불필요한 심볼을 제거하는데 사용한다.

```
s2015722087@ubuntu:~/embeded$ file ./hello.exe
./hello.exe: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically
linked, interpreter /lib/ld-, for GNU/Linux 3.2.0, BuildID[sha1]=99a3c81c21f2cb
49be445e54a7596291b82f59a, not stripped
s2015722087@ubuntu:~/embeded$ arm-linux-gnueabi-strip hello.exe
s2015722087@ubuntu:~/embeded$ file ./hello.exe
./hello.exe: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically
linked, interpreter /lib/ld-, for GNU/Linux 3.2.0, BuildID[sha1]=99a3c81c21f2cb
49be445e54a7596291b82f59a, stripped
s2015722087@ubuntu:~/embeded$
```

File 명령어를 이용하여 파일 정보를 볼 때 계속해서 not stripped 표시가 있어 의아했는데 이 명령어를 사용하면서 표시를 이해하게 되었다.