Homework #3: Symbol Resolution

Due: Apr. 24, 11:59 PM
Responsible TA: Jaeho Bae (bjho@unist.ac.kr)

1 Homework Description

This homework is designed to reinforce your understanding of the output of the compiler and the symbol resolution performed by the linker. As we learned in class, the compiler outputs relocatable object files with the file extension `*.o` in the Executable and Linkable Format (ELF). To see this output invoke `gcc` with the `-c` flag.

```
$ gcc -c hw3_a.c
```

Also, we can see the contents of the symbol table in these relocatable object files with the linux utility `readelf`.

```
$ readelf -s hw3_a.o
```

We also learned that the linker links these relocatable objects together into an executable file.

Task. `hw3` is an executable whose output prints your student ID to the screen. However, it's doing some things in the background that aren't obvious until we look at the symbol tables.

hw3_a.o

```
Symbol table '.symtab' contains 12 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS hw3_a.c
     2: 0000000000000000     0 SECTION LOCAL  DEFAULT    1 .text
     3: 0000000000000000     0 SECTION LOCAL  DEFAULT    4 .bss
     4: 0000000000000000     4 OBJECT  LOCAL  DEFAULT    4 var_1
     5: 0000000000000000     0 SECTION LOCAL  DEFAULT    5 .rodata
     6: 0000000000000000     4 OBJECT  GLOBAL DEFAULT    3 var_2
     7: 0000000000000008     8 OBJECT  GLOBAL DEFAULT  COM var_3
     8: 0000000000000000    90 FUNC    GLOBAL DEFAULT    1 main
     9: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND puts
    10: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND foo
    11: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND bar
```

hw3_b.o

```
Symbol table '.symtab' contains 14 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS hw3_b.c
     2: 0000000000000000     0 SECTION LOCAL  DEFAULT    1 .text
     3: 0000000000000000     0 SECTION LOCAL  DEFAULT    3 .data
     4: 0000000000000000     0 SECTION LOCAL  DEFAULT    4 .bss
     5: 000000000000000c     4 OBJECT  LOCAL  DEFAULT    3 var_5
```

```
9        6: 0000000000000000     4 OBJECT  LOCAL  DEFAULT    4 var_3.0
10       7: 0000000000000000    12 OBJECT  GLOBAL DEFAULT    3 my_int
11       8: 0000000000000004     4 OBJECT  GLOBAL DEFAULT  COM var_3
12       9: 0000000000000000    31 FUNC    GLOBAL DEFAULT    1 foo
13      10: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND var_2
14      11: 000000000000001f    80 FUNC    GLOBAL DEFAULT    1 bar
15      12: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND sqrt
16      13: 000000000000006f    25 FUNC    GLOBAL DEFAULT    1 foo_bar
```

Your job is to use your knowledge of C, ELF, and linkers to recreate `hw3_a.c` and `hw3_b.c` so that their symbol tables match those shown above with the criteria below.

- The symbol tables must have the same number of lines.
- All rows must match in the `Size`, `Type`, `Bind`, `Vis`, `Ndx`, and `Name` columns (The `Value` field can be mismatched).
- For the symoble table of `hw3_b.o` line 6 the numbers after the decimal point (i.e., `var_3.0`) might not match mine, thats ok.

Running `$ ./make` must complete without any compilation errors and produce an executable named `hw3`. When executed, `hw3` must print your student ID followed by a newline (`\n`) and exit.

2 Environment

To complete this homework, you must access the server at `10.20.18.55` via `ssh` using the account information provided to you. Note that this server can only be accessed from the UNIST internal network, so make sure to use a VPN if you are connecting from outside.

Please also note that grading will be conducted in the provided server environment. In other words, your code must compile successfully on our server, and the symbol table must match correctly in that environment. If either of these conditions is not satisfied, points will be deducted. No exceptions will be made.

3 Submission Guidelines

- You should submit a single zip archive through Blackboard, named in the format `Your_ID-hw3.zip`. For example, if your student ID is 20231234, then your submission file should be named `20231234-hw3.zip`. The zip archive must include the following files:
  - `hw3_a.c` and `hw3_b.c`: your completed versions based on the provided skeleton code [1].
  - `Makefile`: submit the provided `Makefile` [1] as is
  - `report.pdf`: your report must answer every question in this handout. The report may be written in either English or Korean

  If the required naming convention is not followed exactly, the submission will receive a score of zero. No exceptions will be made.
- **Late submission policy.** Late submission will be assessed a penalty of 10% per day (We will only accept late submissions of up to 3 days).

4 Questions

(a) Describe what each field in the symbol table (`Value`, `Size`, `Type`, `Bind`, `Vis`, `Ndx`, and `Name`) means. Also describe the possible values that each field can take.

(b) Run readelf on `hw3`. Do you notice any cases where the linker had to disambiguate between symbols? If so, which ones? Why did this happen?

(c) Write a few sentences explaining your observations from examining the symbol table of the executable `hw3`.

Note

- **Grading policy.** Your work will be evaluated based on three components: (1) how accurately your symbol table matches the expected result, with points deducted for any inconsistencies, (2) the output of `hw3`, which must be your student ID, and (3) the quality of your report. Be sure to write the report as clearly and thoroughly as possible.
- **AI usage.** You may use generative AI tools only in the limited role of **a search engine substitute (e.g., instead of Google**). This means you may ask AI tools for explanations of concepts or for help finding reference material, but **you may not request or use code of any form.**

  AI tools are a part of today's world and will be an even larger part of the one you'll enter after you graduate. Our goal is to help you build skills and understanding that surpass what AI can offer, so that you will not be replaceable by AI. The restrictions that are in place for this course are there because we believe that an over-reliance on AI would prevent you from fully achieving the learning objectives of the course and developing the foundational knowledge you will need to be successful in your future coursework and career. Ultimately, the value you gain from this course will be directly related to the effort you put in. Generally, the more you rely on AI, the less you'll learn. Use AI thoughtfully, if and when it is permitted, but never as a substitute for your own learning.
- **Be careful about plagiarism!** We will conduct strict cross-plagiarism detection for evaluation, including a series of answer generated by ChatGPT, code found online, and your submissions. Therefore, we do not recommend consulting ChatGPT or online solutions. Last semester, we identified several cases of plagiarism using an automated tool. Also, do not share your input values. It is easy to verify from your report PDF whether you solved the bomb on your own.
- **Questions.** If you have any requests or questions (technical difficulties, late submission due to inevitable circumstances, etc.), please ask the TAs on Blackboard. We generally encourage the use of Blackboard for discussions. However, for urgent issues or secret issues, you can send an email to the responsible TA.

References

[1] CSE251: System Programming. 2026. hw3.zip. https://websec-lab.github.io/courses/2026s-cse251/hw/hw3.zip.