Lab 15
Static Library v.s Dynamic Library

TA:Richard Liao

Professor: Hsung-Pin Chang

Operating System Lab

Outline

- **Library Overview**
- **Static Library**
  - Overview
  - Creating Static Library
  - Exercise I
- **Dynamic(shared) Library**
  - Overview
  - Creating Dynamic(shared) Library
  - Exercise II

Library Overview

- A library is a collection of implementations, written in terms of a language, that has a well-defined interface that can interact.

- For instance, people who want to write a higher level program can use a library to make system calls.

Static Library

Overview

Overview

- **Static library (**also known as **archives): libxxx.a**
  - Object files and libraries are linked into the binary program image, at compiler time, i.e., **static linking.**
  - If many programs use the same library. It will waste disk and memory space.
  - If a library function is changed, we need to re-link the applications using this library function.

Overview(cont.)

main.c

```
#include <stdio.h>

extern int  add(int);
extern int  sub(int);
extern int  sum(int, int);

main()
{
      int ret1, ret2, ret3;

      ret1 = add(5);
      ret2 = sub(5);
      ret3 = sum(ret1, ret2);

      printf("\n ret from add()= %d", ret1);
      printf("\n ret from sub()= %d", ret2);
      printf("\n ret from sum()= %d", ret3);
}
```

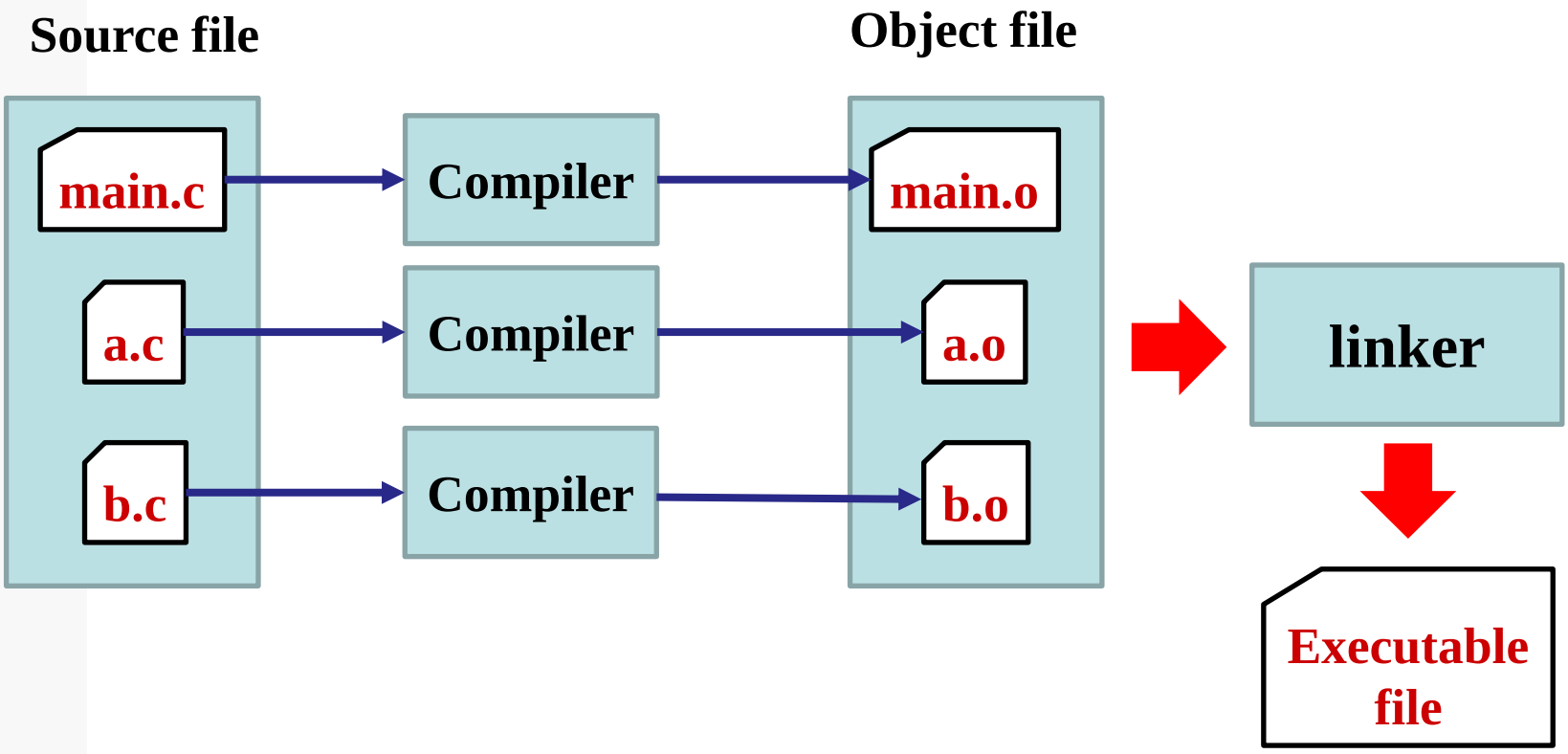
a.c

```
int add(int arg)
{
      arg++;
      return arg;
}
int sub(int   arg)
{
      arg--;
      return arg;
}
```

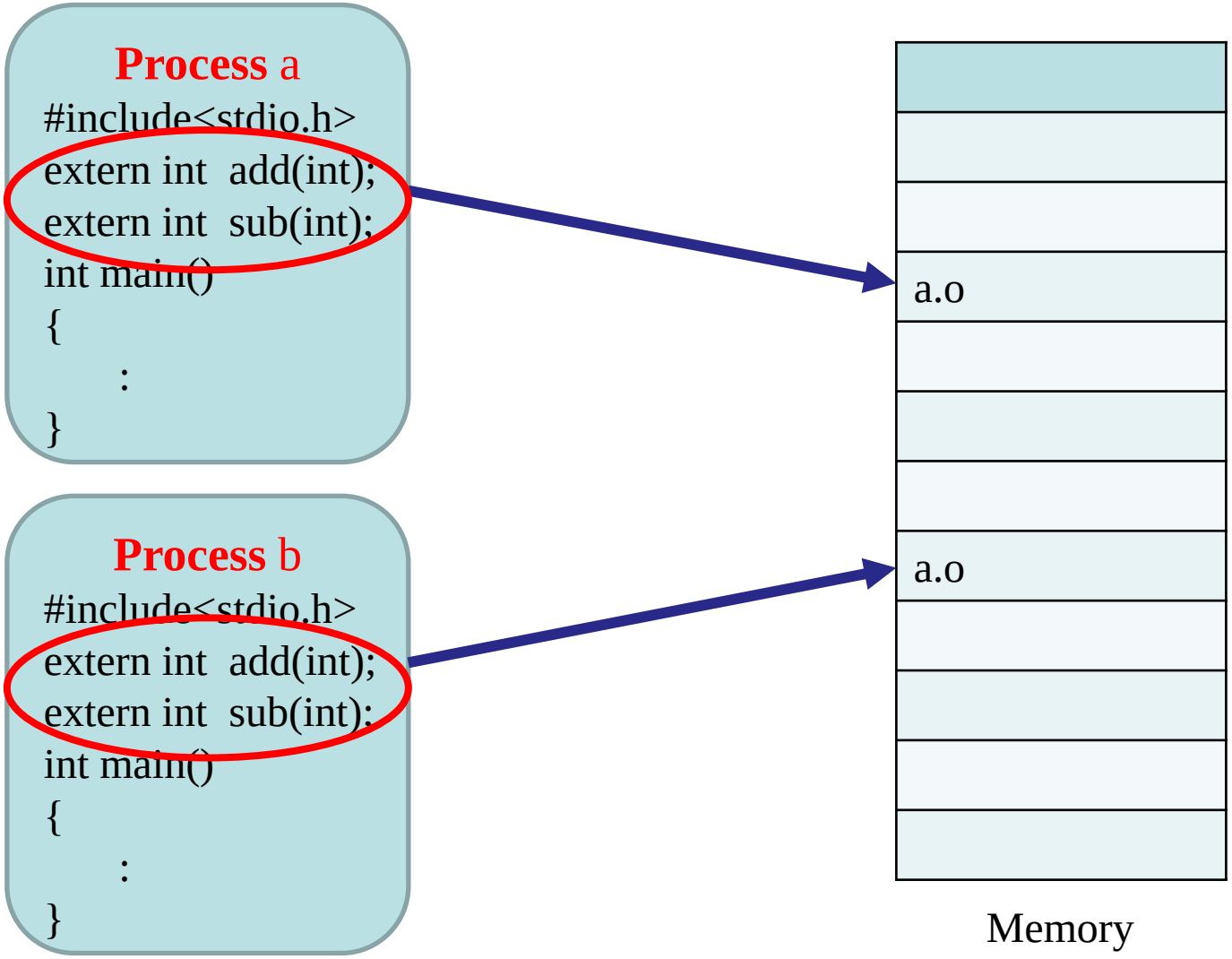
b.c

```
int sum(int  arg1, int arg2)
{
      return (arg1+arg2);
}
```

Overview(cont.)



Overview(cont.)



Static Library

Creating Static Library

Creating Static Library

- **ar** - create, modify, and extract from archives
  - -c: create the archive
  - -r: insert a file member into archive with replacement
  - -d: delete a file member from library
  - -t: list contents of library

eg.

- ***ar –t libtest.a***
- ***ar –d libtest.a b.o***

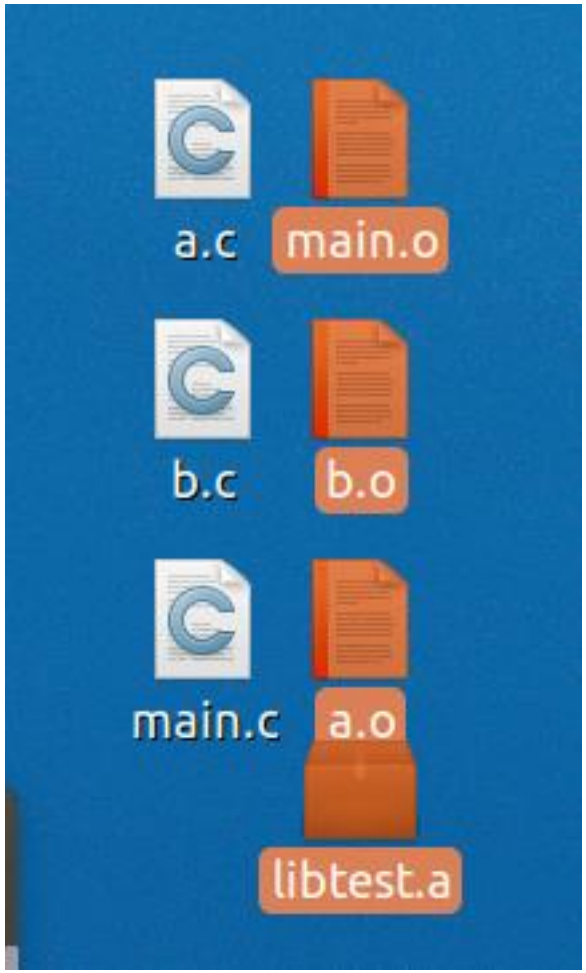
Creating Static Library(cont.)

- First, compile source programs to object files
  - ***gcc –c main.c***
  - ***gcc –c a.c***
  - ***gcc –c b.c***
- Then, create/add/replace object programs into library
  - ***ar –r libtest.a a.o b.o***

Creating Static Library(cont.)

```
richard@richard-oslab:~/Desktop$ gcc -c a.c b.c main.c
main.c:7:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
 main()
 ^~~~
richard@richard-oslab:~/Desktop$ ar -r libtest.a a.o b.o
ar: creating libtest.a
```

Creating Static Library(cont.)



ar –r libtest.a a.o b.o

gcc –c a.c b.c main.c

Creating Static Library(cont.)

- Finally, to link with the library
  - ***gcc –o main main.o libtest.a***

    or
  - ***gcc –o main main.o –L. –ltest***
    - -L. : . = add current directory to the search directory
    - -ltest: means libtest.so or libtest.a

```
richard@richard-oslab:~/Desktop$ gcc -o main main.o libtest.a
```

Creating Static Library(cont.)

gcc –o main main.o libtest.a

Creating Static Library(cont.)

```
richard@richard-oslab:~/Desktop$ ./main

 return 1 from add()= 6
 return 2 from sub()= 4
 return 3 from sum()= 10
```

Static Library

Exercise I

Exercise1(60%)

- Create a static library consists of a object files. The object file should include selection_sort function.

- Create a C program to call the function of the object files and sort the array : 15, 22, 41, 8, 35.

Result

```
richard@richard-oslab:~/Desktop$ ./main
Numbers to be sorted: 15 22 41 8 35
Numbers Sorted: 8 15 22 35 41
```

Dynamic(shared) Library

Overview

Overview

- **Shared library: libxxx.so**
  - A single copy of the library is loaded into memory, many programs can link to the same shared library.
  - The use of shared libraries means that executable programs require less space on memory.
  - The linking will not start until execution time, i.e., **dynamic linking.**

Dynamic (shared) Library

Creating Dynamic Library

Creating Dynamic Library

- First, compile source programs to object files
  - ***gcc –c -fPIC main.c***
  - ***gcc –c -fPIC a.c*** -fPIC : generate *position-independent code*
  - ***gcc –c –fPIC b.c***
- Then, create shared library
  - ***gcc a.o b.o –shared –o libtest.so***
- Finally, to link with shared library
  - ***gcc –o main main.o libtest.so***

Error!

```
richard@richard-oslab:~/Desktop/lab15/example2$ gcc -c -FPIC a.c
richard@richard-oslab:~/Desktop/lab15/example2$ gcc -c -FPIC b.c
richard@richard-oslab:~/Desktop/lab15/example2$ gcc a.o b.o -shared -o libtest.so
richard@richard-oslab:~/Desktop/lab15/example2$ gcc -o main main.o libtest.so
richard@richard-oslab:~/Desktop/lab15/example2$ ./main
./main: error while loading shared libraries: libtest.so: cannot open shared object fil
e: No such file or directory
```

Error!

- Many shared libraries reside in /lib and /usr/lib.
- The error message above occurs because our library resides in the current working directory, which is not part of the standard list searched by the dynamic linker.

Solution

- Method 1:

```
richard@richard-oslab:~/Desktop/lab15/example2$ LD_LIBRARY_PATH=. ./main
```

- Method 2:

```
richard@richard-oslab:~/Desktop/lab15/example2$ gcc -o main main.o ./libtest.so
```

- Method 3:
  - Copy the libtest.so to /lib

Done!

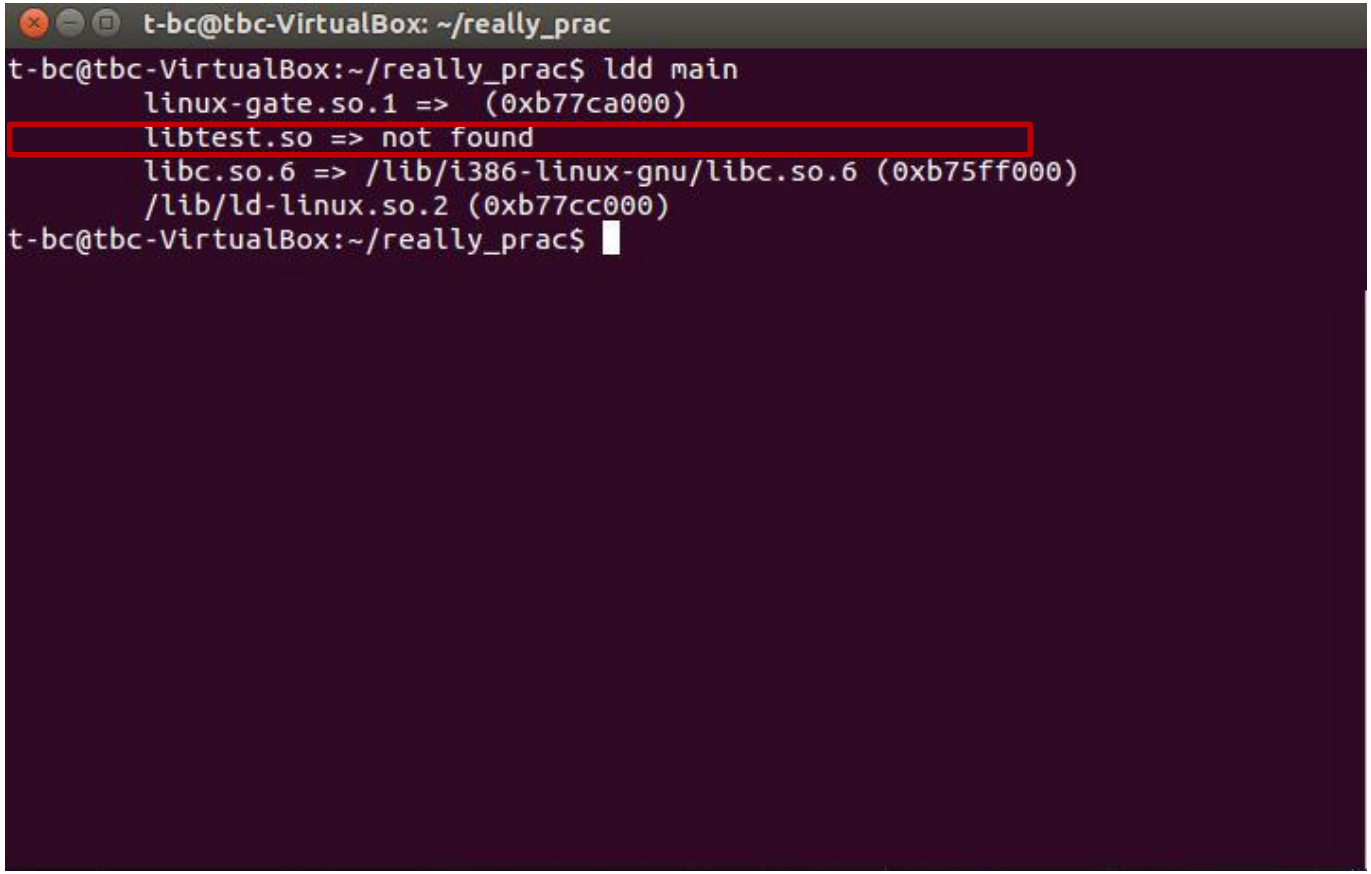
```
richard@richard-oslab:~/Desktop/lab15/example2$ ./main

 return 1 from add()= 6
 return 2 from sub()= 4
 return 3 from sum()= 10
```

list dynamic dependencies

- ***ldd***:
  - Display the shared libraries that a program used.

- In this example, you can type:
  - ***ldd*** (your .out file name)

Not found!


t-bc@tbc-VirtualBox: ~/really_prac
t-bc@tbc-VirtualBox:~/really_prac$ ldd main
linux-gate.so.1 => (0xb77ca000)
libtest.so => not found
libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb75ff000)
/lib/ld-linux.so.2 (0xb77cc000)
t-bc@tbc-VirtualBox:~/really_prac$

Solution

- This error occurs because of the same reason above we have got.

- So you can also type:
  - LD_LIBRARY_PATH=. ldd main

Solution

```
richard@richard-oslab:~/Desktop/lab15/example2$ LD_LIBRARY_PATH=.  ldd main
        linux-vdso.so.1 (0x00007ffc477c7000)
        ./libtest.so (0x00007f6d3b6c9000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f6d3b2d8000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f6d3bacd000)
```

Dynamic Library

Exercise II

Exercise2(30%)

- Use the exercise1’s code to create another library, the library is created as shared library.

- Compare two executable(.out) files size, using the command ***ls -l*** and ***size*** to show the file information.

Result

Static Library

```
richard@richard-oslab:~/Desktop/lab15/exercise1$ ls -l
total 28
-rw-r--r-- 1 richard richard 1640 Aug 29 17:02 a.o
-rw-r--r-- 1 richard richard 1800 Aug 29 17:02 libtest.a
-rwxr-xr-x 1 richard richard 8496 Aug 29 17:02 main
-rw-rw-r-- 1 richard richard  432 Aug 29 17:44 main.c
-rw-r--r-- 1 richard richard 2168 Aug 29 17:02 main.o
```

Dynamic Library

```
richard@richard-oslab:~/Desktop/exercise2$ ls -l
total 40
-rw-r--r-- 1 richard richard  353 Aug 31 17:08 dynamic.c
-rw-r--r-- 1 richard richard 1688 Aug 31 17:08 dynamic.o
-rwxr-xr-x 1 richard richard 8440 Aug 31 17:23 exercise2
-rw-r--r-- 1 richard richard  432 Aug 29 17:01 exercise2.c
-rw-r--r-- 1 richard richard 2168 Aug 31 17:07 exercise2.o
```

Reference

- https://kopu.chat/2017/06/20/%E9%81%B8%E6%93%87%E6%8E%92%E5%BA%8F-selection-sort/
- https://stackoverflow.com/questions/29016166/dynamic-library-size-bigger-than-static-library-and-sum-of-linked-objects-size