作者：代兴   邮箱：503268771@qq.com Vx 公众号：东方架构师

静态库的含义

静态库，把多个目标文件打包，合并成 1 个文件。
静态库的符号重定位处理和单个目标文件处理类似。
程序代码使用静态库生成可执行程序，静态库的内容合并到可执行程序。

用 C 程序分析静态库

编写代码：bird.h

```
#ifndef _BIRD_H_
#define _BIRD_H_

extern int bird_height;

extern void bird_fly();

#endif
```

编写代码：bird.c

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int bird_height = 0xB1B2B3B4;

void bird_fly()
{
    printf("Bird is flying at height %#X \n", bird_height);
}
```

编写代码：dog.h

```
#ifndef _DOG_H_
#define _DOG_H_

extern int dog_speed;

extern void dog_run();

#endif
```

编写代码：dog.c

```
#include <unistd.h>
```

```
#include <stdio.h>
#include <stdlib.h>

int dog_speed = 0xD1D2D3D4;

void dog_run()
{
    printf("Dog is running at speed %#X \n", dog_speed);
}
```

编写代码：main.c

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include "bird.h"
#include "dog.h"

int main_int = 0x61626364;

void main_print_param(char *name, void *addr, int value)
{
    printf(" %15s  addr = %p  value = %#X \n", name, addr, value);
}

void main_print_func(char *name, void *addr)
{
    printf(" %15s  addr = %p   \n", name, addr);
}

int main()
{
    // 重置变量。
    main_int = 0xF1F2F3F4;

    // 调用函数。
    bird_fly();
    dog_run();

    // 查看变量的地址、值。
    printf("\nparam addr and value : \n");
    main_print_param("bird_height", &bird_height, bird_height);
    main_print_param("dog_speed", &dog_speed, dog_speed);
    main_print_param("main_int", &main_int, main_int);

    // 查看方法的地址。
    printf("\nfunc addr :  \n");
    main_print_func("bird_fly", bird_fly);
    main_print_func("dog_run", dog_run);
    main_print_func("main", main);
    // 休眠。方便查看内存布局。
    sleep(80000);
    return 0;
}
```

编译代码：

```
gcc bird.c -c -o bird.o
gcc dog.c -c -o dog.o
ar -cr my_static.a bird.o dog.o

gcc main.c my_static.a -o main

readelf -a my_static.a > my_static.a.elf.txt
readelf -a main > main.elf.txt

objdump -D my_static.a > my_static.a.dump.txt
objdump -D main > main.dump.txt
```

运行代码：

```
[root@local static]# ./main
Bird is flying at height 0XB1B2B3B4
Dog is running at speed 0XD1D2D3D4

param addr and value :
     bird_height  addr = 0x601048  value = 0XB1B2B3B4
       dog_speed  addr = 0x60104c  value = 0XD1D2D3D4
        main_int  addr = 0x601044  value = 0XF1F2F3F4

func addr :
        bird_fly  addr = 0x4006d4
         dog_run  addr = 0x4006f1
            main  addr = 0x40061b
```

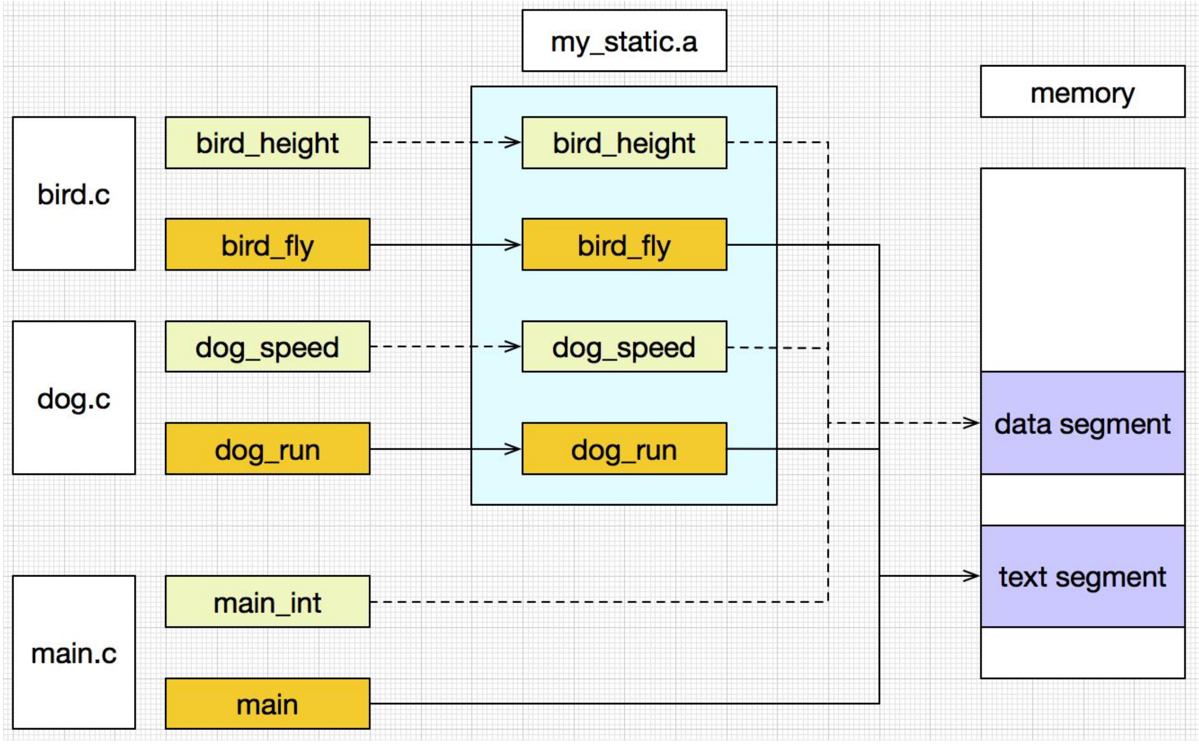
查看进程的内存布局：

```
[root@local static]# ps aux | grep ./main
root      77207  0.0  0.0   4216   352 pts/3    S+   21:23   0:00 ./main
root      77287  0.0  0.0 112812   992 pts/4    S+   21:23   0:00 grep --color=auto ./main
[root@local static]# cat /proc/77207/maps
00400000-00401000             r-xp               00000000               08:03               17815742
/root/code/x86-asm/common2/elf2/static/main
00600000-00601000             r--p               00000000               08:03               17815742
/root/code/x86-asm/common2/elf2/static/main
00601000-00602000             rw-p               00001000               08:03               17815742
/root/code/x86-asm/common2/elf2/static/main
7f8071c7a000-7f8071e3e000 r-xp 00000000 08:03 15928                      /usr/lib64/libc-2.17.so
7f8071e3e000-7f807203d000 ---p 001c4000 08:03 15928                      /usr/lib64/libc-2.17.so
7f807203d000-7f8072041000 r--p 001c3000 08:03 15928                      /usr/lib64/libc-2.17.so
7f8072041000-7f8072043000 rw-p 001c7000 08:03 15928                      /usr/lib64/libc-2.17.so
```

```
7f8072043000-7f8072048000 rw-p 00000000 00:00 0
7f8072048000-7f807206a000 r-xp 00000000 08:03 611075                     /usr/lib64/ld-2.17.so
7f807225e000-7f8072261000 rw-p 00000000 00:00 0
7f8072267000-7f8072269000 rw-p 00000000 00:00 0
7f8072269000-7f807226a000 r--p 00021000 08:03 611075                     /usr/lib64/ld-2.17.so
7f807226a000-7f807226b000 rw-p 00022000 08:03 611075                     /usr/lib64/ld-2.17.so
7f807226b000-7f807226c000 rw-p 00000000 00:00 0
7fffa043d000-7fffa045e000 rw-p 00000000 00:00 0                          [stack]
7fffa0474000-7fffa0476000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
```

查看符号的内存布局：
符号 bird_height、dog_speed、main_int，地址前缀为 0x6010，在数据区 `00601000-00602000 rw-p` 。
符号 bird_fly、dog_run、main，地址前缀为 0x4006，在代码区 `00400000-00401000 r-xp` 。

符号的流转示意图：



查看静态库的组成：
查看文件 my_static.a.elf.txt、my_static.a.dump.txt。
静态库包含多个目标文件。my_static.a.elf.txt 包含 `File: my_static.a(bird.o)` 、`File: my_static.a(dog.o)`。

查看符号表

File: my_static.a(bird.o)	Symbol table '.symtab' contains 12 entries:
	   Num: Value Size Type Bind Vis Ndx Name
	     9: 0000000000000000 4 OBJECT GLOBAL DEFAULT 3 bird_height
	    10: 0000000000000000 29 FUNC GLOBAL DEFAULT 1 bird_fly

File: my_static.a(dog.o)	Symbol table '.symtab' contains 12 entries:
	   Num: Value Size Type Bind Vis Ndx Name
	     9: 0000000000000000 4 OBJECT GLOBAL DEFAULT 3 dog_speed
	    10: 0000000000000000 29 FUNC GLOBAL DEFAULT 1 dog_run

查看重定位表

File: my_static.a(bird.o)	Relocation section '.rela.text’ at offset 0x238 contains 3 entries: Offset Info Type Sym. Value Sym. Name + Addend 000000000006 000900000002 R_X86_64_PC32 0000000000000000 bird_height - 4
File: my_static.a(dog.o)	Relocation section '.rela.text’ at offset 0x228 contains 3 entries: Offset Info Type Sym. Value Sym. Name + Addend 000000000006 000900000002 R_X86_64_PC32 0000000000000000 dog_speed - 4