

《计算机系统基础》习题

第四章 程序的链接

2019年11月6日

第四章课后习题布置

- 第二版课本p.g. 216 开始
- 第 4、5、7、10、11题
- 习题截止时间：2019年11月15日24时
- PA 2-2 截止时间：2019年11月16日24时
- PA 2-3 截止时间：2019年11月19日24时

第四章内容梳理

编译、汇编和静态链接

hello_world.c

```
#include<stdio.h>

int main() {
    printf("Hello World!\n");
    return 0;
}
```

```
$ gcc -o hello_world hello_world.c
$ ./hello_world
Hello World!
```

编译
运行

编译、汇编和静态链接

```
$ gcc -o hello_world hello_world.c
```

一条命令执行了四个步骤

hello_world.c

gcc -E

Preprocess:

展开#include, 宏定义

hello_world.i

gcc -S

Compile:

将高级语言程序转变成汇编语言程序

hello_world.S

gcc -c

Assemble:

将汇编语言程序转变成机器语言程序

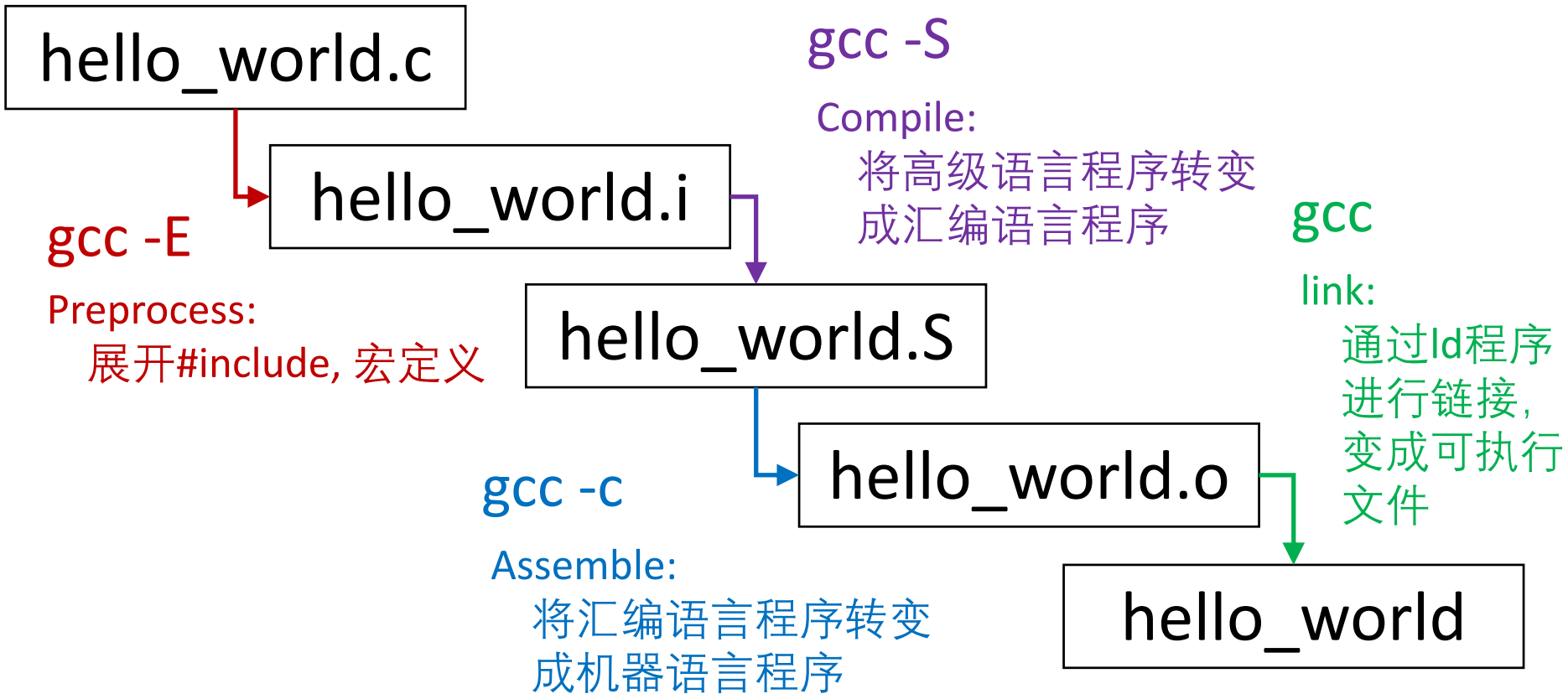
hello_world.o

gcc

link:

通过ld程序进行链接, 变成可执行文件

hello_world



编译、汇编和静态链接

```
$ gcc -o hello_world hello_world.c
```

一条命令执行了四个步骤

hello_world.c

gcc -E

Preprocess:

展开#include, 宏定义

hello_world.i

gcc -S

Compile:

将高级语言程序转变成汇编语言程序

hello_world.S

gcc -c

Assemble:

将汇编语言程序转变成机器语言程序

hello_world.o

gcc

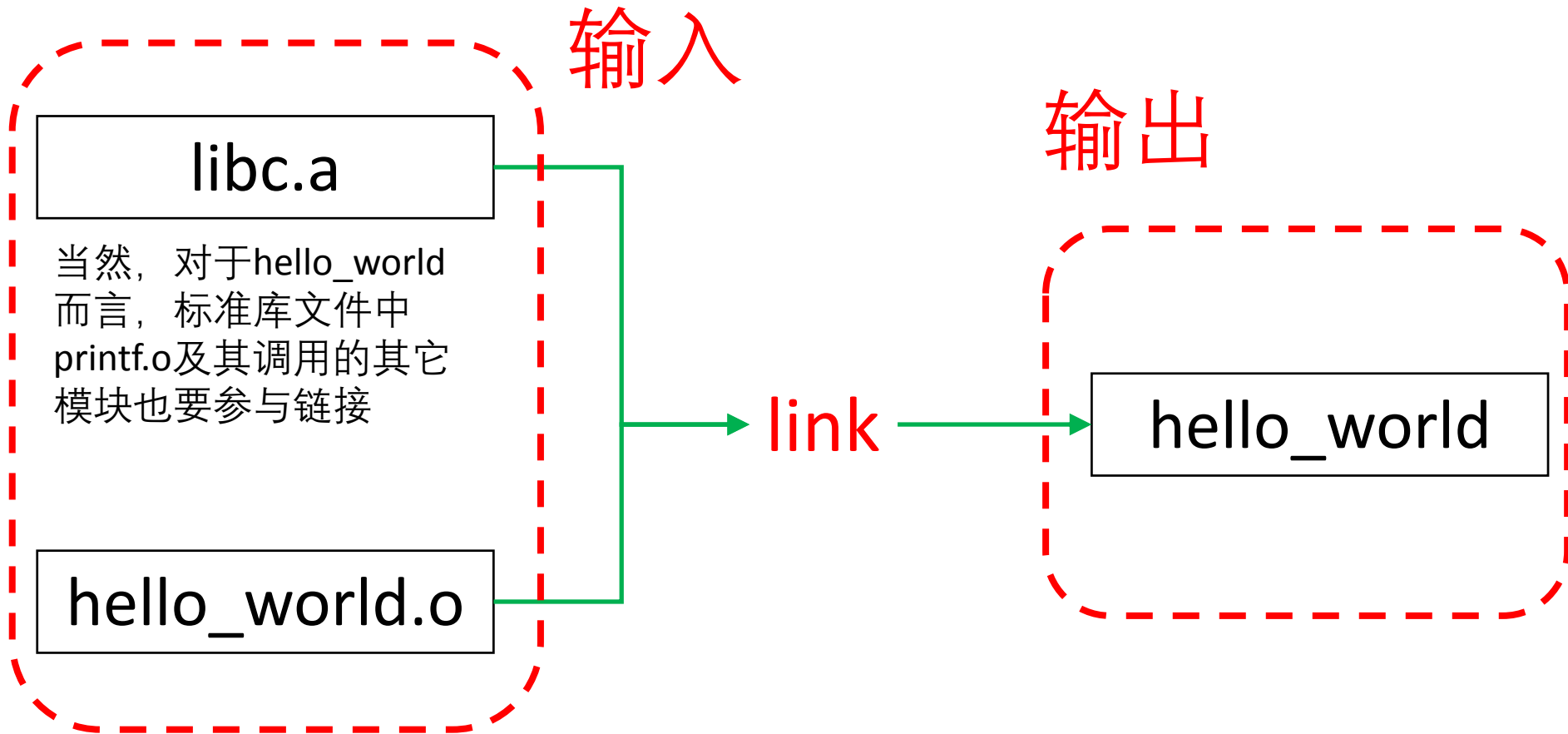
link:

通过ld程序进行链接, 变成可执行文件

hello_world

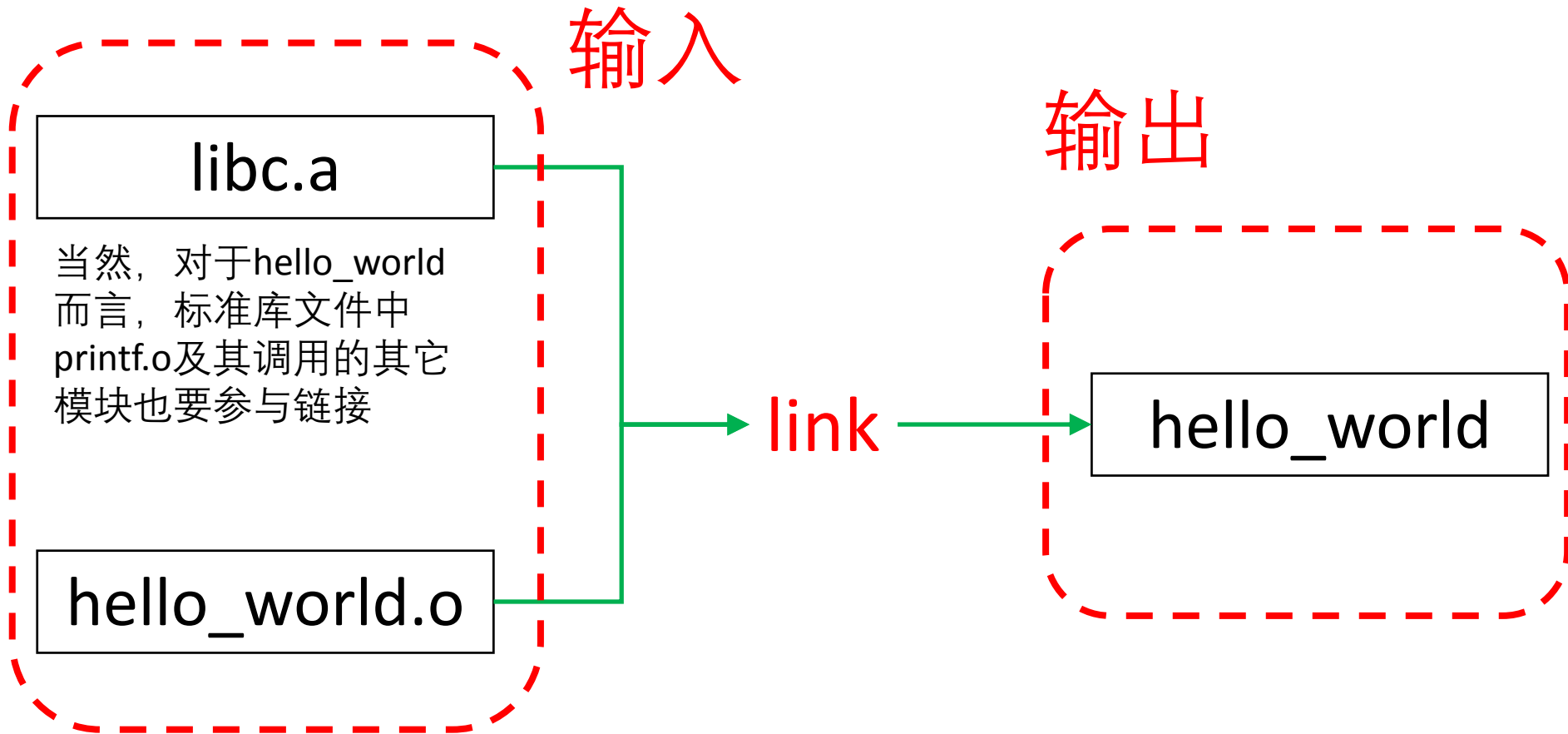
link: 第四章的主题

第四章 程序的链接



第四章 程序的链接

第四章的问题



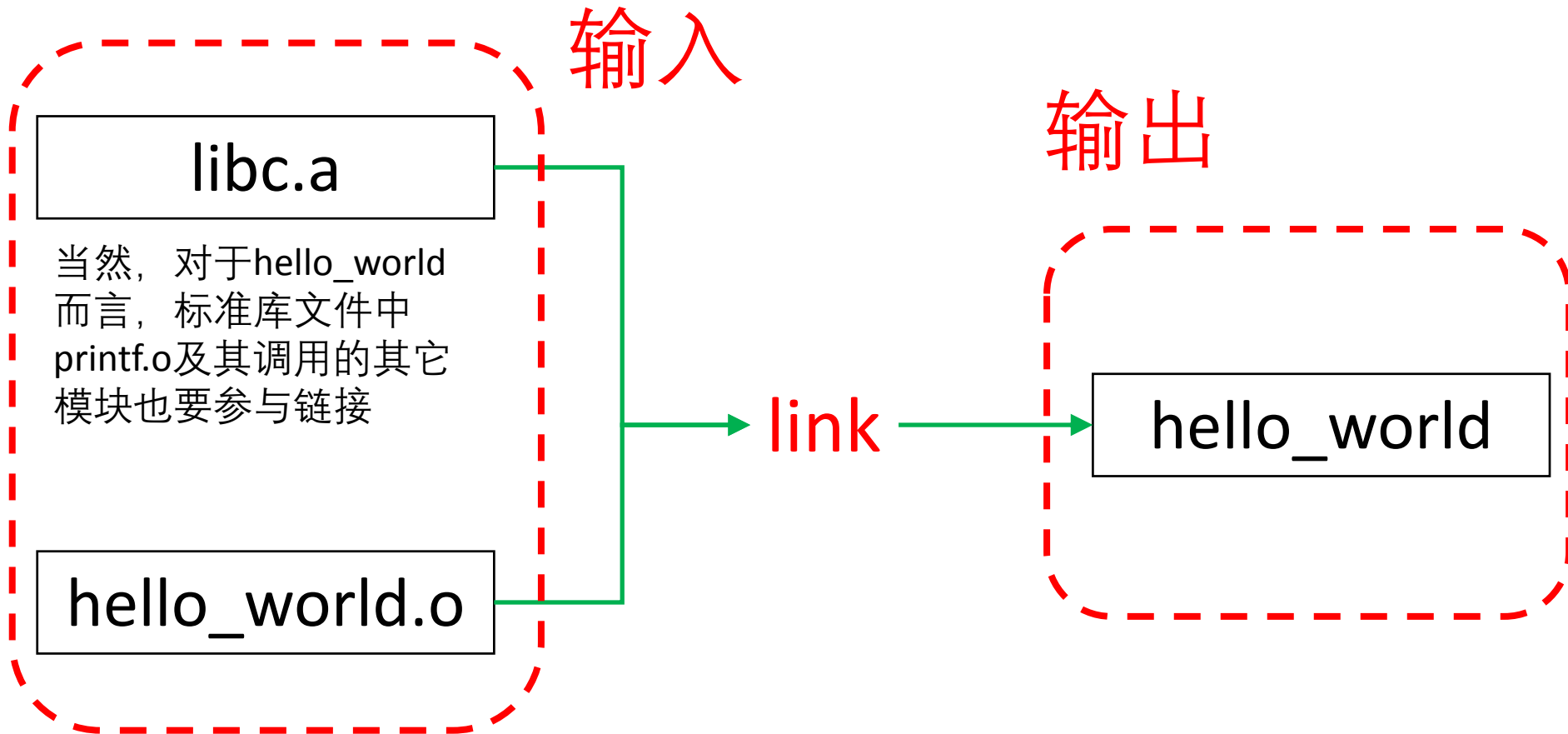
问题1：输入输出什么格式？

问题2：输入输出对应的核心操作？

问题3：link的两种方式及核心问题？

第四章 程序的链接

第四章的问题



问题1：输入输出什么格式？

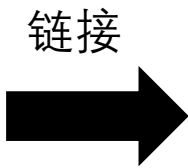
问题2：输入输出对应的核心操作？

问题3：link的两种方式及核心问题？

ELF 文件格式

输入：ELF可重定位目标文件

ELF头 (ELF Header)
.init节
.text节 (程序代码)
.rodata节 (只读数据)
.data节 (已初始化全局变量数据)
.bss节 (未初始化全局变量数据, 其实啥也没存)
.symtab节 (符号表)
.xxx节
.yyy节
.strtab节 (字符串表)
节头表 (Section Headers)

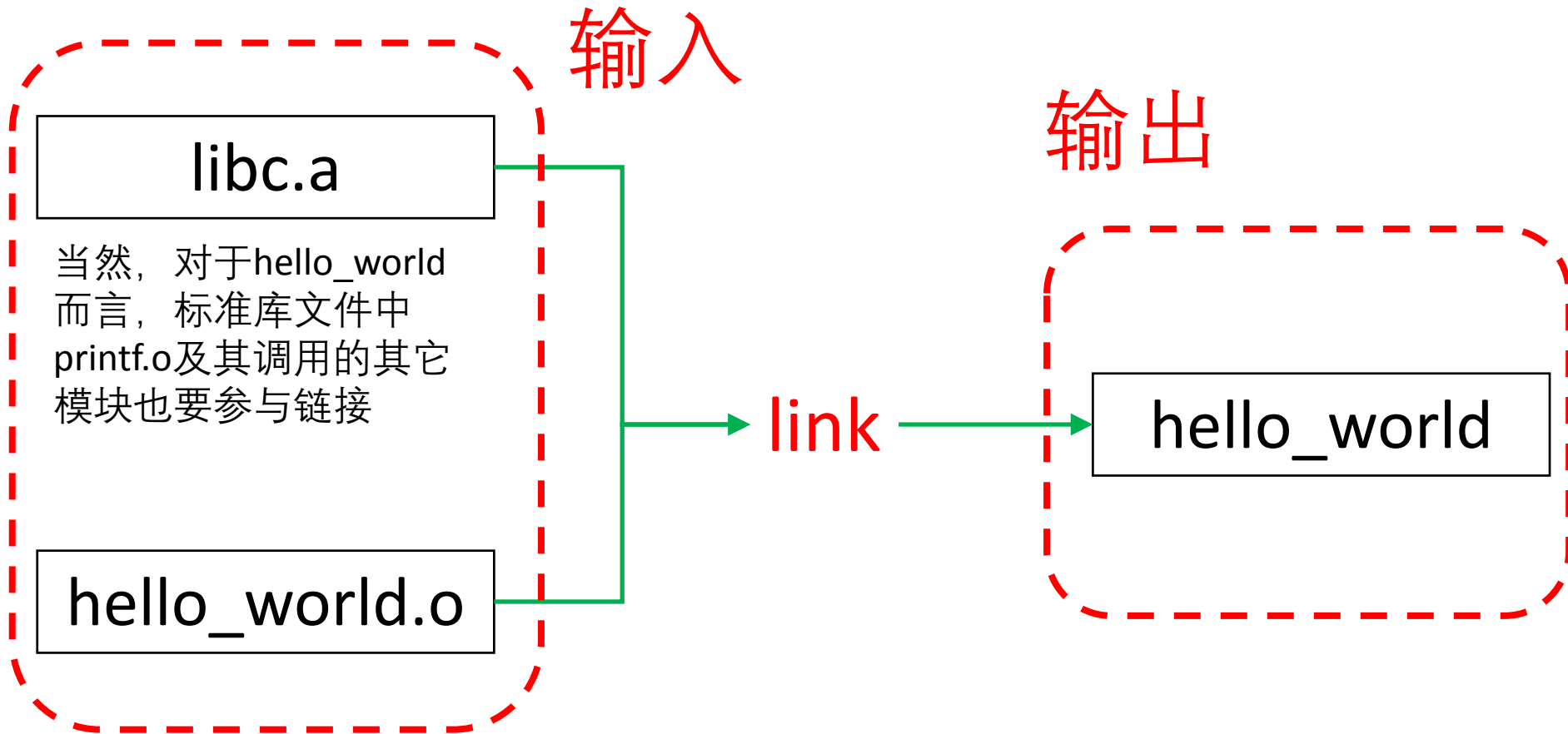


输出：ELF可执行目标文件

ELF头 (ELF Header)
程序头表 (Program Headers)
.init节
.text节 (程序代码)
.rodata节 (只读数据)
.data节 (已初始化全局变量数据)
.bss节 (未初始化全局变量数据, 其实啥也没存)
.symtab节 (符号表)
.xxx节
.yyy节
.strtab节 (字符串表)
节头表 (Section Headers)

第四章 程序的链接

第四章的问题



问题1：输入输出什么格式？

问题2：输入输出对应的核心操作？

问题3：link的两种方式及核心问题？

符号表和符号解析

输入：ELF可重定位目标文件

ELF头 (ELF Header)
.init节
.text节 (程序代码)
.rodata节 (只读数据)
.data节 (已初始化全局变量数据)
.bss节 (未初始化全局变量数据, 其实啥也没存)
.symtab节 (符号表)
.xxx节
.yyy节
.strtab节 (字符串表)
节头表 (Section Headers)

对它的核心操作

符号解析

符号表和符号解析

- 什么叫符号 (Symbol) ?

```
void swap(void);  
  
int buf[2] = {1, 2};  
  
int main() {  
    int a = 0;  
    swap();  
    return 0;  
}
```

哪些是符号？

符号表和符号解析

- 什么叫符号 (Symbol) ?

```
void swap(void);  
int buf[2] = {1, 2};  
int main() {  
    int a = 0;  
    swap();  
    return 0;  
}
```

哪些是符号？

外延：函数名，全局变量名

内涵：？

符号表和符号解析

- 什么叫符号 (Symbol) ?

```
void swap(void);  
int buf[2] = {1, 2};  
int main() {  
    int a = 0;  
    swap();  
    return 0;  
}
```

哪些是符号？

外延：函数名，全局变量名

内涵：**目标的（首）地址**

符号表和符号解析

- 符号的类型
 - 全局符号
 - 外部符号
 - 本地符号
- 全局符号的强弱特性和多重定义（p.g. 176三条规则）
 - 强符号
 - 弱符号

```
multiple definition of `instr_execute_2op'  
.....: first defined here  
collect2: error: ld returned 1 exit status
```

如何重现这个错误？

在编译过程中哪一步报的错？

如何得知nemu中有多少个instr_execute_2op？

在空白中指明下列三种情况之一：

- REF(x.i) --> DEF(x.k)：将模块i中对符号x的引用关联到模块k中x的定义
- ERROR：链接错误
- UNKNOWN：链接器任意选择一个符号定义

A.	<pre>/* Module 1 */ int main() { }</pre>	<pre>/* Module 2 */ static int main=1; int p2() { }</pre>
----	--	---

(a) REF(main.1) --> DEF(_____.____)

(b) REF(main.2) --> DEF(_____.____)

B.	<pre>/* Module 1 */ int x; void main() { }</pre>	<pre>/* Module 2 */ double x; int p2() { }</pre>
----	--	--

(a) REF(x.1) --> DEF(_____.____)

(b) REF(x.2) --> DEF(_____.____)

C.	<pre>/* Module 1 */ int x=1; void main() { }</pre>	<pre>/* Module 2 */ double x=1.0; int p2() { }</pre>
----	--	--

(a) REF(x.1) --> DEF(_____.____)

(b) REF(x.2) --> DEF(_____.____)

在空白中指明下列三种情况之一：

- REF(x.i) --> DEF(x.k)：将模块i中对符号x的引用关联到模块k中x的定义
- ERROR：链接错误
- UNKNOWN：链接器任意选择一个符号定义

A.	<pre>/* Module 1 */ int main() { }</pre>	<pre>/* Module 2 */ static int main=1; int p2() { }</pre>
----	--	---

(a) REF(main.1) --> DEF(main.1)

(b) REF(main.2) --> DEF(main.2)

B.	<pre>/* Module 1 */ int x; void main() { }</pre>	<pre>/* Module 2 */ double x; int p2() { }</pre>
----	--	--

(a) REF(x.1) --> DEF(UNKNOWN)

(b) REF(x.2) --> DEF(UNKNOWN)

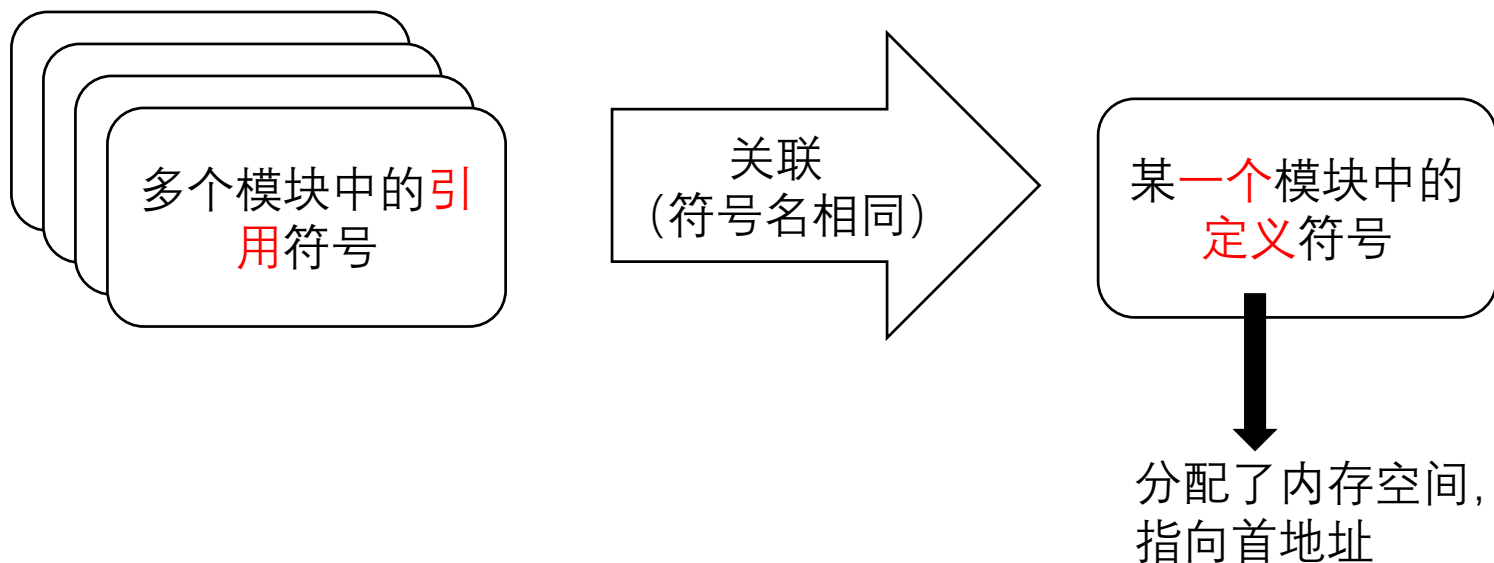
C.	<pre>/* Module 1 */ int x=1; void main() { }</pre>	<pre>/* Module 2 */ double x=1.0; int p2() { }</pre>
----	--	--

(a) REF(x.1) --> DEF(ERROR)

(b) REF(x.2) --> DEF(ERROR)

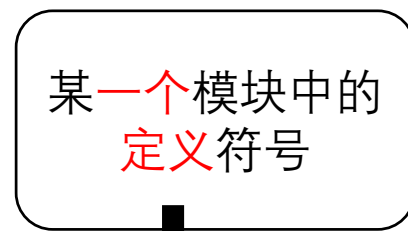
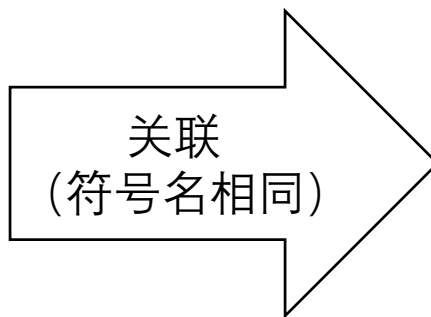
符号表和符号解析

- 符号解析



符号表和符号解析

- 符号解析



分配了内存空间，
指向首地址

ELF可重定位目标文件

ELF头 (ELF Header)
...
.symtab节 (符号表)
.xxx节
.yyy节
.strtab节 (字符串表)
节头表 (Section Headers)

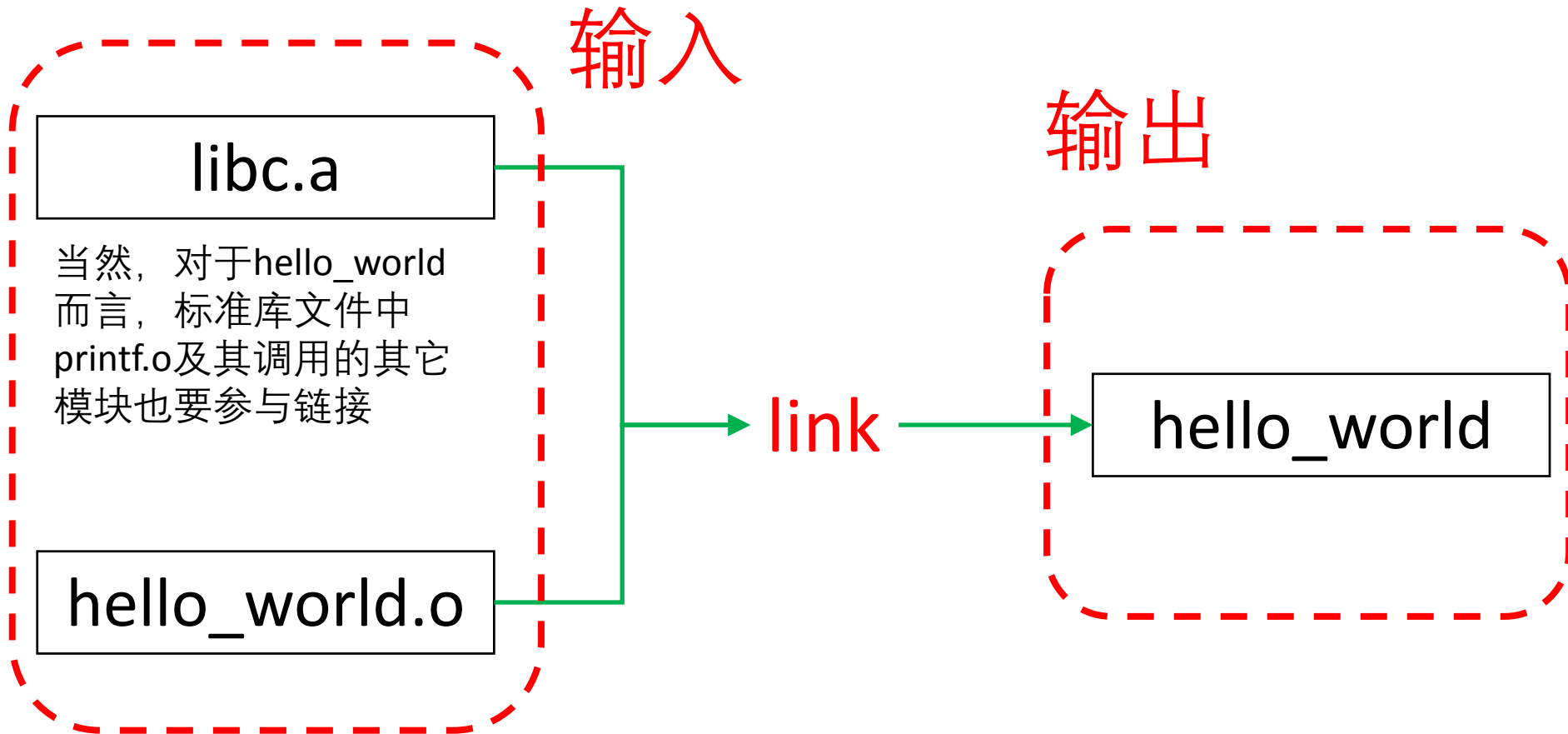
如何获取符号信息?

对.o文件，配合.strtab节查看.symtab节

对应命令行：readelf -s xxx
对应代码实现：PA 2-3.2

第四章 程序的链接

第四章的问题



问题1：输入输出什么格式？

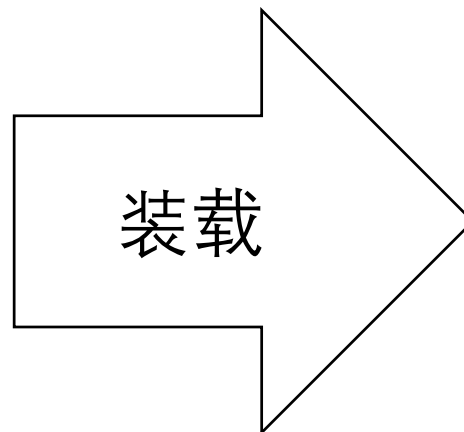
问题2：输入输出对应的核心操作？

问题3：link的两种方式及核心问题？

可执行文件的加载

输出：ELF可执行目标文件

ELF头 (ELF Header)
程序头表 (Program Headers)
.init节
.text节 (程序代码)
.rodata节 (只读数据)
.data节 (已初始化全局变量数据)
.bss节 (未初始化全局变量数据, 其实啥也没存)
.symtab节 (符号表)
.xxx节
.yyy节
.strtab节 (字符串表)
节头表 (Section Headers)

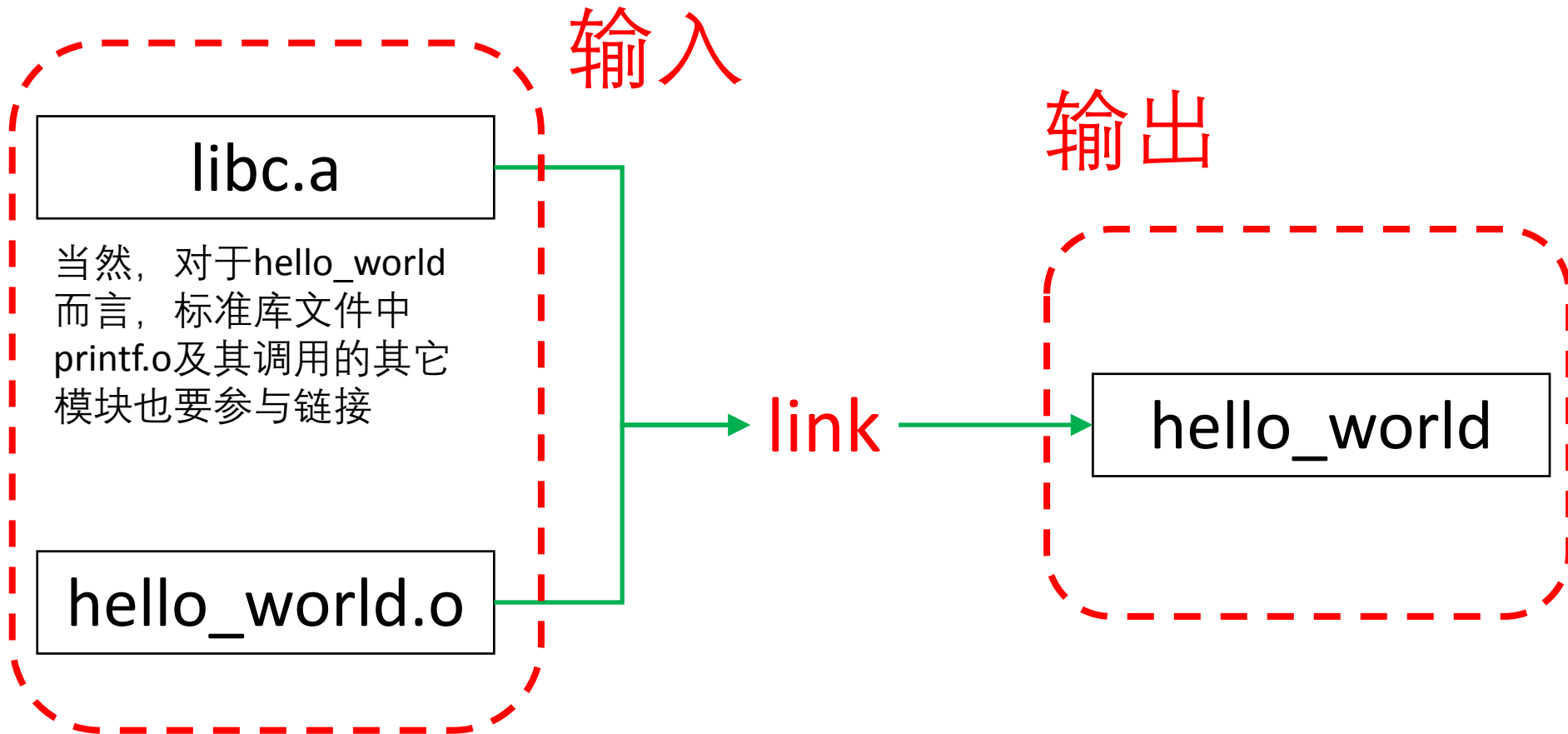


对应 PA 2-2

内存

第四章 程序的链接

第四章的问题



问题1：输入输出什么格式？

问题2：输入输出对应的核心操作？

问题3：link的两种方式及核心问题？

静态、动态链接

- 静态链接
 - 课本 4.3.3 节
- 动态链接
 - 课本 4.6 节
- 重定位
 - 课本 4.4 节