

# 计算机系统 Hello World!

湖南大学

《计算机系统》课程教学组



# 计算机语言



人与计算机的交流，或者说人给计算机布置任务，是通过计算机语言来实现的。



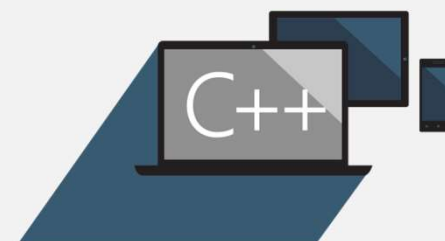
基本分类：

- 高级语言
- 汇编语言
- 机器语言















编写程序需要遵循一定的字符及语法规则

```
root@pwn-pc /test
1 /*
2  模拟登录程序
3 */
4 #include <stdio.h>
5 #include <string.h>
6 int main()
7 {
8     int numbers;
9     int flag = 1;
10    char name[10] = { "admin" }, name1[10];
11    char passwd[10] = { "123456" }, passwd1[10];
12    for(numbers; numbers; numbers++)
13    {
14        printf("请输入用户名: (1-10个字符)\n");
15        fgets(name1, 10, stdin);
16        if(strcmp(name, name1))
17        {
18            printf("用户名错误, 请重新输入: \n");
19            flag--;
20            printf("您还有%d次机会\n", flag);
21            continue;
22        }
23        printf("请输入密码: (1-10个数字及字母)\n");
24        fgets(passwd1, 10, stdin);
25        if(strcmp(passwd, passwd1))
26        {
```



# 计算机语言

Feb 2022	Feb 2021	Change	Programming Language	Ratings	Change
1	3	▲	 Python	15.33%	+4.47%
2	1	▼	 C	14.08%	-2.26%
3	2	▼	 Java	12.13%	+0.84%
4	4		 C++	8.01%	+1.13%
5	5		 C#	5.37%	+0.93%
6	6		 Visual Basic	5.23%	+0.90%
7	7		 JavaScript	1.83%	-0.45%
8	8		 PHP	1.79%	+0.04%
9	10	▲	 Assembly language	1.60%	-0.06%
10	9	▼	 SQL	1.55%	-0.18%
11	13	▲	 Go	1.23%	-0.05%
12	15	▲	 Swift	1.18%	+0.04%

## 计算机语言

# C语言

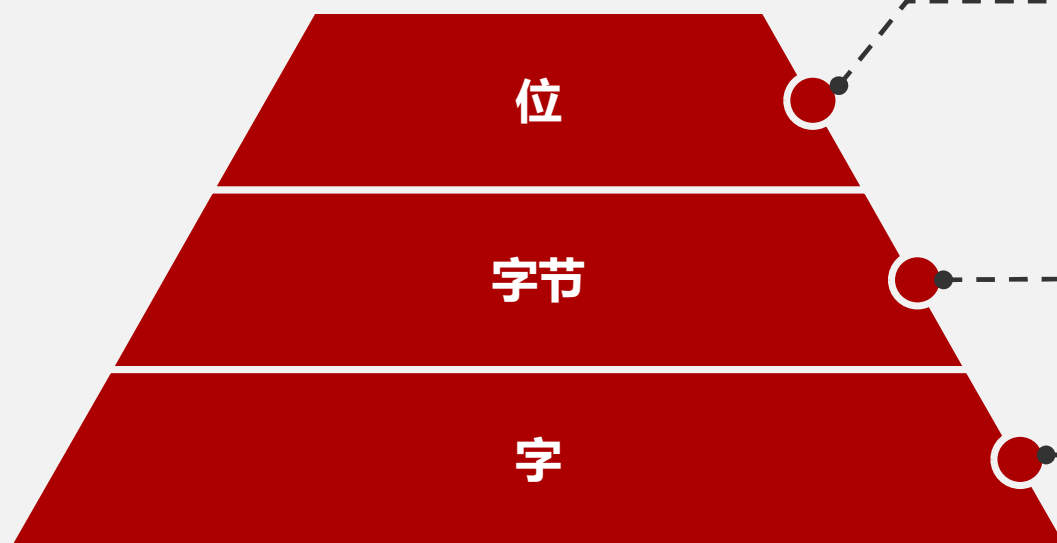
1960年代，AT&T公司**贝尔实验室**的研究员**Ken Thompson**为了玩一个自己编的游戏Space Travel，背着老板，为一台PDP-7开发操作系统。后来，这个操作系统被命名为——**UNIX**。



1971年，同样酷爱Space Travel的 **Dennis M. Ritchie** 为了能早点儿玩上游戏，加入了Thompson的开发项目，合作开发UNIX。他的主要工作是改造B语言，使其更成熟。1972年，美国贝尔实验室的 D.M. Ritchie 在B语言的基础上最终设计出了一种新的语言，他取了BCPL的第二个字母作为这种语言的名字，这就是**C语言**。



## 基础知识



"位(bit)"是电子计算机中**最小的数据单位**。每一位的状态只能是0或1。

8个二进制位构成1个"字节(Byte)",它是存储空间的基本计量单位。**1个字节可以储存1个英文字母或者半个汉字**,换句话说,1个汉字占据2个字节的存储空间。

"字"由若干个字节构成,字的位数叫做**字长**,不同档次的机器有不同的字长。例如一台8位机,它的1个字就等于1个字节,字长为8位。如果是一台16位机,那么,它的1个字就由2个字节构成,字长为16位。在32位操作系统当中,一个字是4个字节,**字是计算机进行数据处理和运算的单位**。

基础知识

内存

0000	00000000
0001	00000000
0010	00000000
0011	00000000
0100	00000000
0101	00000000
0110	00000000
0111	00000000
1000	00000000
1001	00000000
1010	00000000
1011	00000000
1100	00000000
1101	00000000
1110	00000000
1111	00000000

地址为4位，内存容量为16个字节

寄存器

\$1	00000000
\$2	00000000
\$3	00000000
\$4	00000000

某机器有4个8位寄存器

%1	00000000 00000000
%2	00000000 00000000
.....	00000000 00000000
%6	00000000 00000000

某机器有6个16位寄存器

AX	00000000 00000000 00000000 00000000
.....	00000000 00000000 00000000 00000000
ZX	00000000 00000000 00000000 00000000

某机器有26个32位寄存器

## 基础知识

**ASCII**: American Standard Code for Information Interchange, **美国信息交换标准代码**, 用指定的7 位或8 位二进制数组合来表示128 或256 种可能的字符。标准ASCII 码也叫基础ASCII码, 使用7 位二进制数 (剩下的1位二进制为0) 来表示所有的大写和小写字母, 数字0 到9、标点符号, 以及在美式英语中使用的特殊控制字符。

演示代码 **ascii.c**

ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo
0 0 NUL	16 10 DLE	32 20 (espacio)	48 30 0
1 1 SOH	17 11 DC1	33 21 !	49 31 1
2 2 STX	18 12 DC2	34 22 "	50 32 2
3 3 ETX	19 13 DC3	35 23 #	51 33 3
4 4 EOT	20 14 DC4	36 24 \$	52 34 4
5 5 ENQ	21 15 NAK	37 25 %	53 35 5
6 6 ACK	22 16 SYN	38 26 &	54 36 6
7 7 BEL	23 17 ETB	39 27 '	55 37 7
8 8 BS	24 18 CAN	40 28 (	56 38 8
9 9 TAB	25 19 EM	41 29 )	57 39 9
10 A LF	26 1A SUB	42 2A *	58 3A :
11 B VT	27 1B ESC	43 2B +	59 3B ;
12 C FF	28 1C FS	44 2C ,	60 3C <
13 D CR	29 1D GS	45 2D -	61 3D =
14 E SO	30 1E RS	46 2E .	62 3E >
15 F SI	31 1F US	47 2F /	63 3F ?

ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo
64 40 @	80 50 P	96 60 `	112 70 p
65 41 A	81 51 Q	97 61 a	113 71 q
66 42 B	82 52 R	98 62 b	114 72 r
67 43 C	83 53 S	99 63 c	115 73 s
68 44 D	84 54 T	100 64 d	116 74 t
69 45 E	85 55 U	101 65 e	117 75 u
70 46 F	86 56 V	102 66 f	118 76 v
71 47 G	87 57 W	103 67 g	119 77 w
72 48 H	88 58 X	104 68 h	120 78 x
73 49 I	89 59 Y	105 69 i	121 79 y
74 4A J	90 5A Z	106 6A j	122 7A z
75 4B K	91 5B [	107 6B k	123 7B {
76 4C L	92 5C \	108 6C l	124 7C
77 4D M	93 5D ]	109 6D m	125 7D }
78 4E N	94 5E ^	110 6E n	126 7E ~
79 4F O	95 5F _	111 6F o	127 7F □

# Hello World!

```
#include <stdio.h>

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



# Hello World!

```
#include <stdio.h>

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

#	i	n	c	l	u	d	e	<sp>	<	s	t	d	i	o	.
35	105	110	99	108	117	100	101	32	60	115	116	100	105	111	46
h	>	\n	\n	i	n	t	<sp>	m	a	i	n	(	)	\n	{
104	62	10	10	105	110	116	32	109	97	105	110	40	41	10	123
\n	<sp>	<sp>	<sp>	<sp>	p	r	i	n	t	f	(	"	h	e	l
10	32	32	32	32	112	114	105	110	116	102	40	34	104	101	108
l	o	,	<sp>	w	o	r	l	d	\	n	"	)	;	\n	}
108	111	44	32	119	111	114	108	100	92	110	34	41	59	10	125

The ASCII text representation of hello.c.

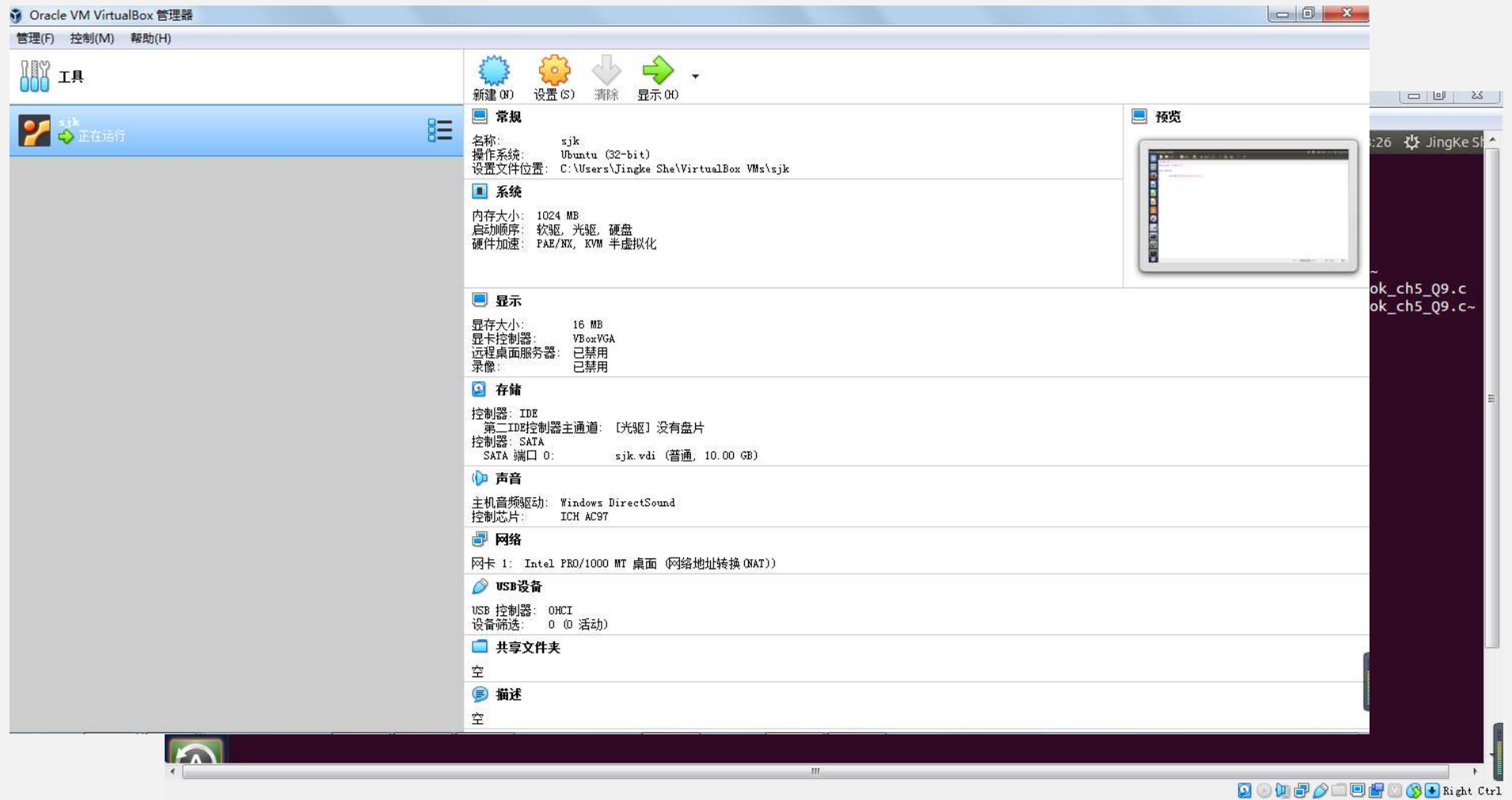
**GCC**

# GNU Compiler Collection



- **我们使用gcc作为本课程工作语言C的编译器套件**
- **完全的自由软件**
- **Linux环境下使用gcc**
  - 推荐使用虚拟机+Ubuntu+gcc

# 虚拟机



# 编译过程

hello.c



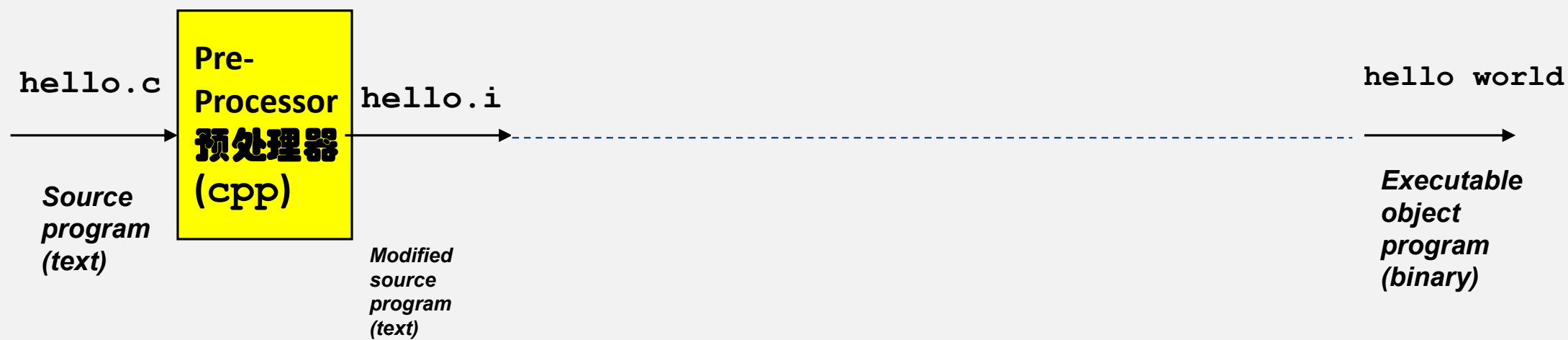
*Source  
program  
(text)*

hello world



*Executable  
object  
program  
(binary)*

# 编译过程

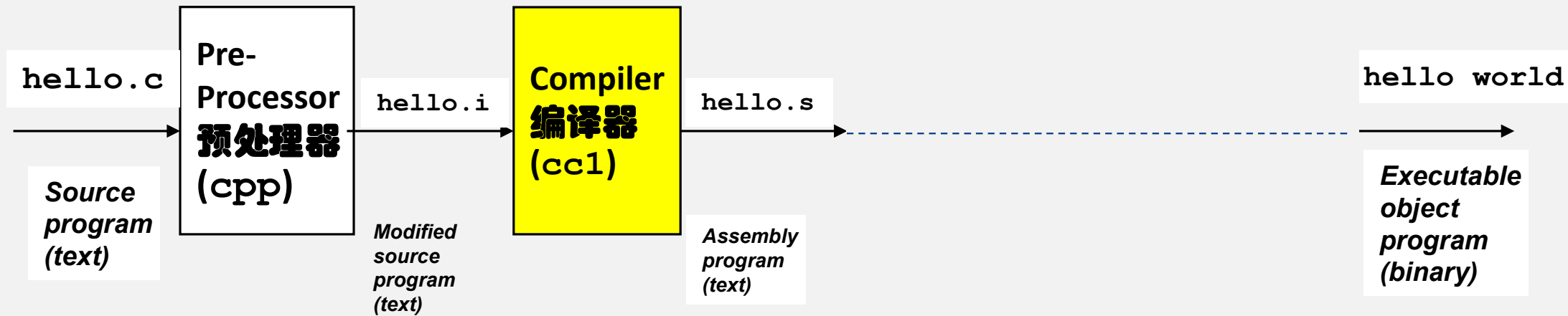


## 预处理

- 编译器将C源代码中的包含的头文件如stdio.h编译进来
- 预处理器 (cpp) 根据以字符#开头的命令 (directives) , 修改原始C程序
  - ◆ 如hello.c中 `#include <stdio.h>` 指令告诉预处理器读系统头文件stdio.h的内容, 并将其直接插入到程序文本中去
  - ◆ 结果就得到另外一个C程序, 通常是 以.i作为文件扩展名的。

**gcc -E hello.c -o hello.i**

# 编译过程



这个阶段中，gcc首先检查代码

1. 规范性；
2. 是否有语法错误；
3. 确定代码的实际要做的工作

gcc将代码翻译成汇编语言。

**gcc -S hello.i -o hello.s**

## 汇编语言

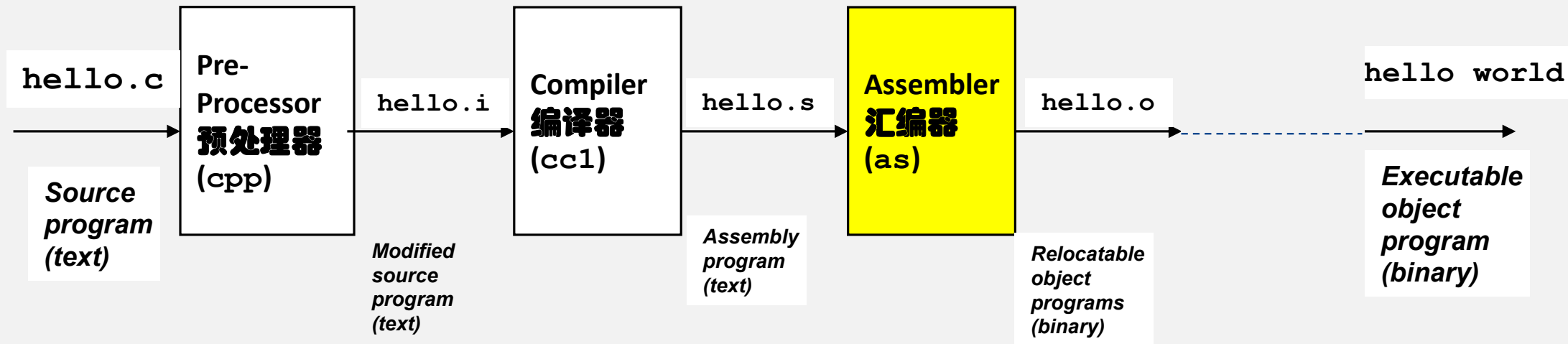
- 汇编更接近机器语言，能够直接对硬件进行操作
- 让编程者知道计算机在软件层面的最底层操作
- 逆向工程

## 汇编示例

```
1  main:
2  subl  $8, %esp
3  movl  $/LC0, %edi
4  call  puts
5  movl  $0, %eax
6  addl  $8, %esp
7  ret
```



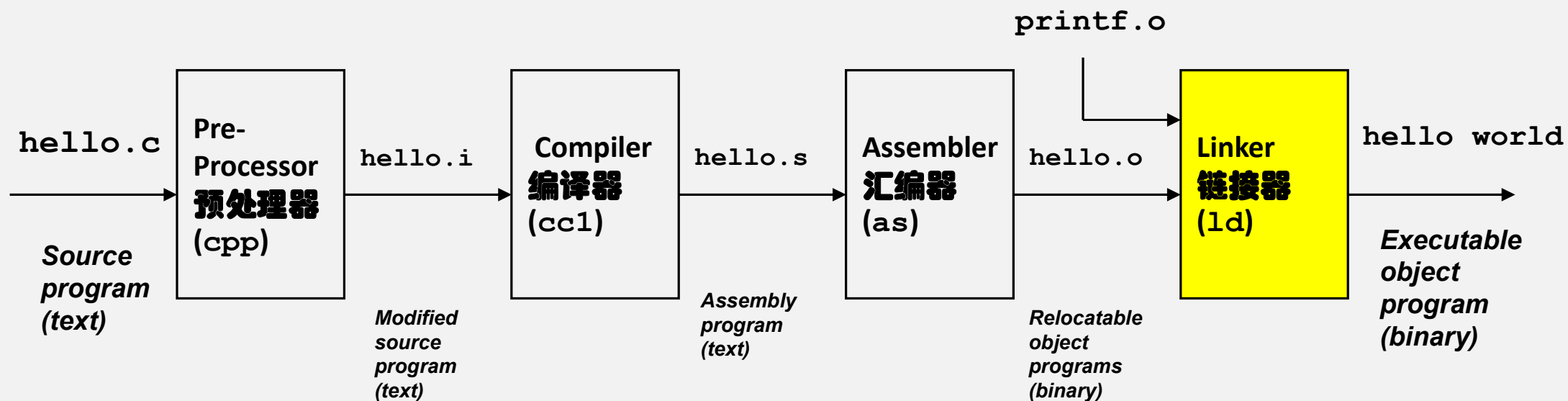
## 编译过程



汇编阶段将编译阶段生成的“`.s`”文件转成二进制目标代码。

**`gcc -c hello.s -o hello.o`**

## 链接过程



在成功编译之后，即进入了链接阶段

**gcc hello.o -o hello.out**

## 链接过程

**1** 汇编程序中可能还有许多没有解决的问题

**2** 怎么解决?

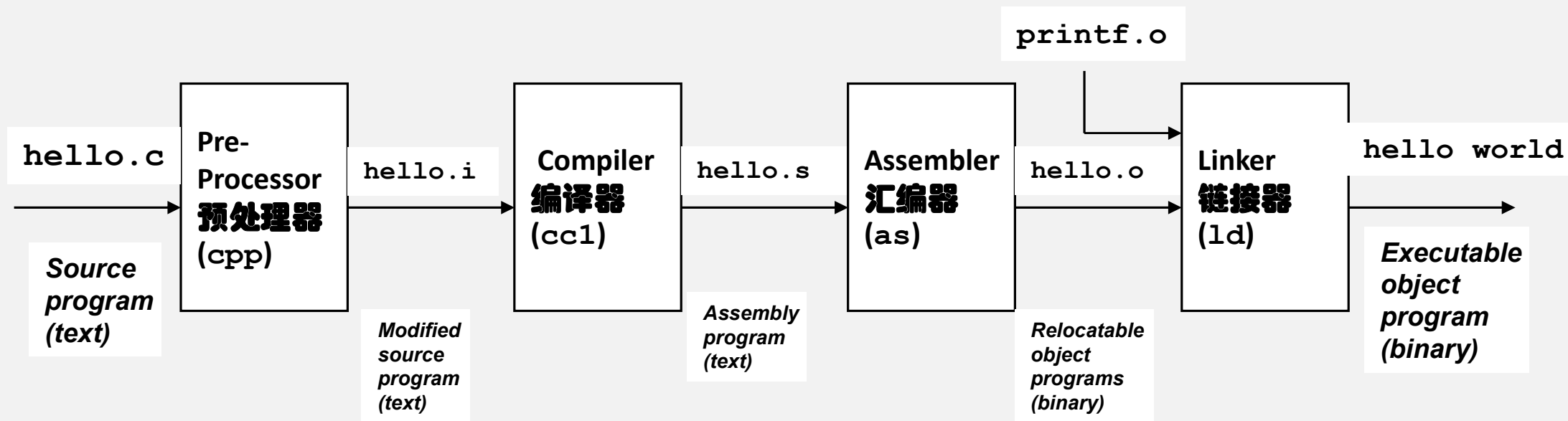
**3** 链接的含义

某个源文件中的函数可能引用了另一个源文件中定义的某个符号如变量或者函数调用  
在程序中可能调用了某个库文件中的函数, etc

通过“链接”将程序运行所需要的所有资源全部结合起来

链接程序的主要工作就是将有关的目标文件彼此相连接, 即将在一个文件中引用的符号同该符号在另外一个文件中的定义连接起来, 使得所有的这些目标文件成为一个能够load执行的统一整体

## 完整过程



- 了解编译系统的工作过程可以帮助我们：
- 优化程序性能；
- 理解链接时出现的错误；
- 避免安全漏洞

## 单选题 1分

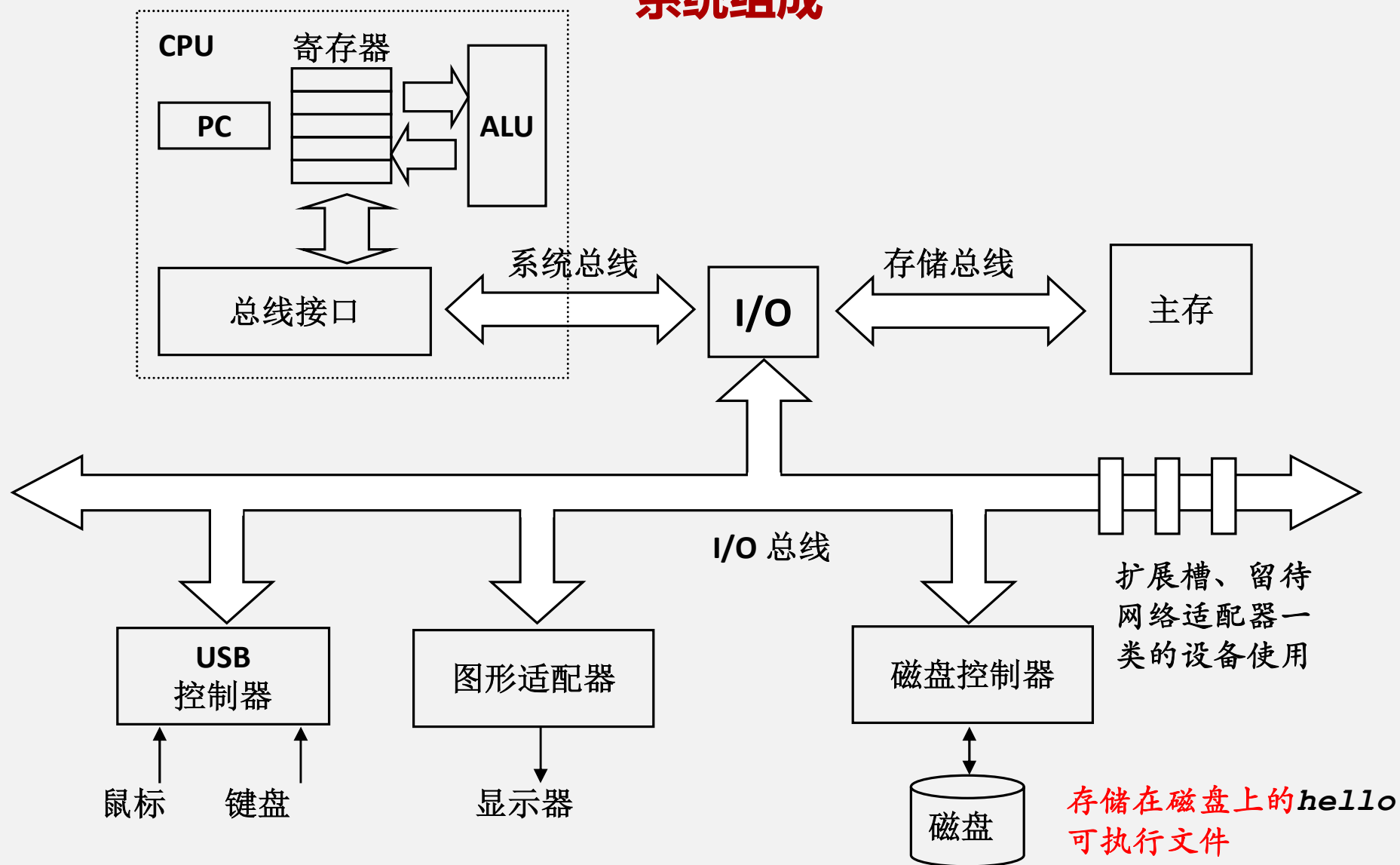


从一个C语言源程序到可执行文件的正确路径是：

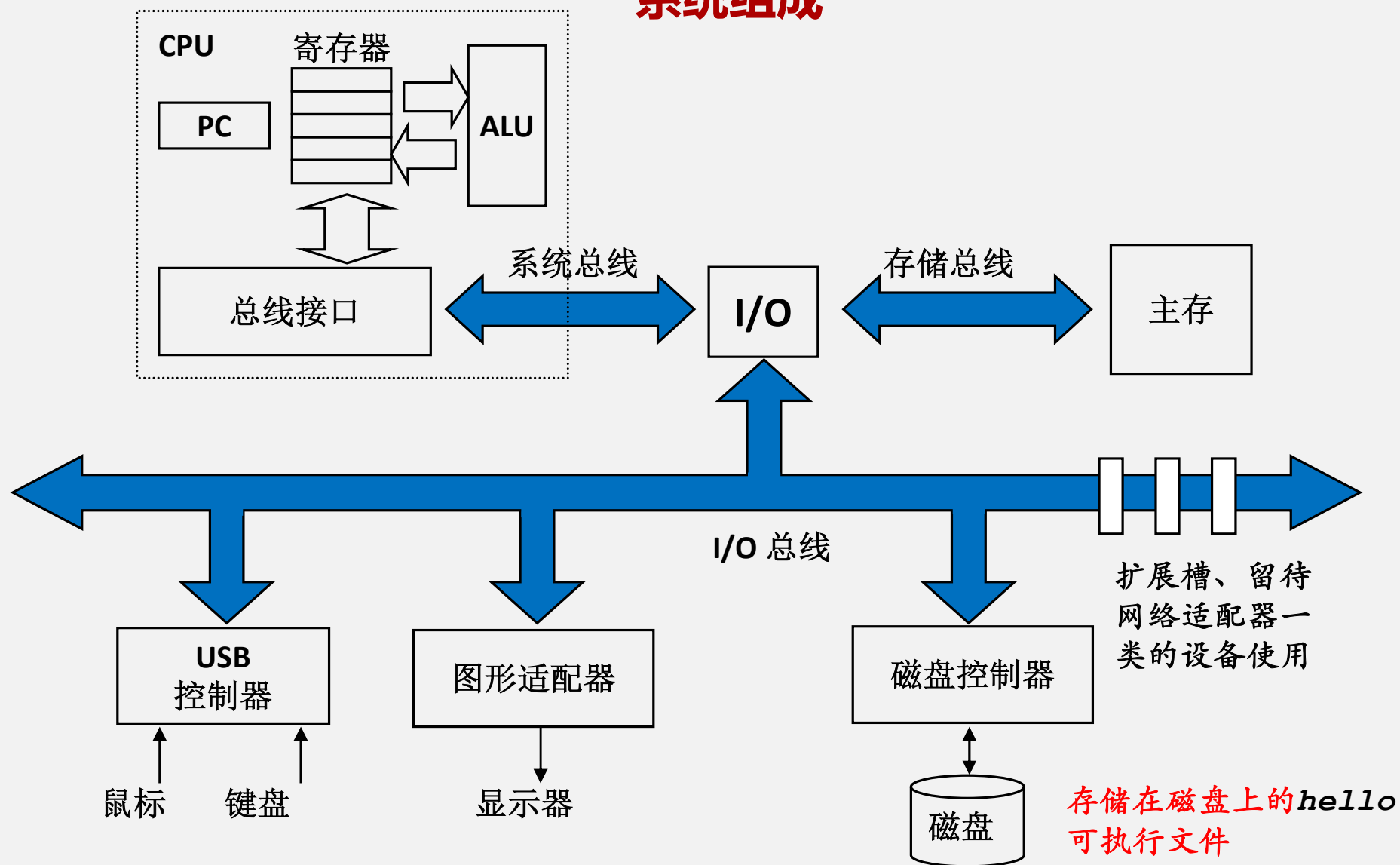
- ☐ A .c → .s → .i → .o → .out
- ☐ B .c → .i → .o → .s → .out
- ☐ C .c → .s → .o → .i → .out
- ☒ D .c → .i → .s → .o → .out

提交

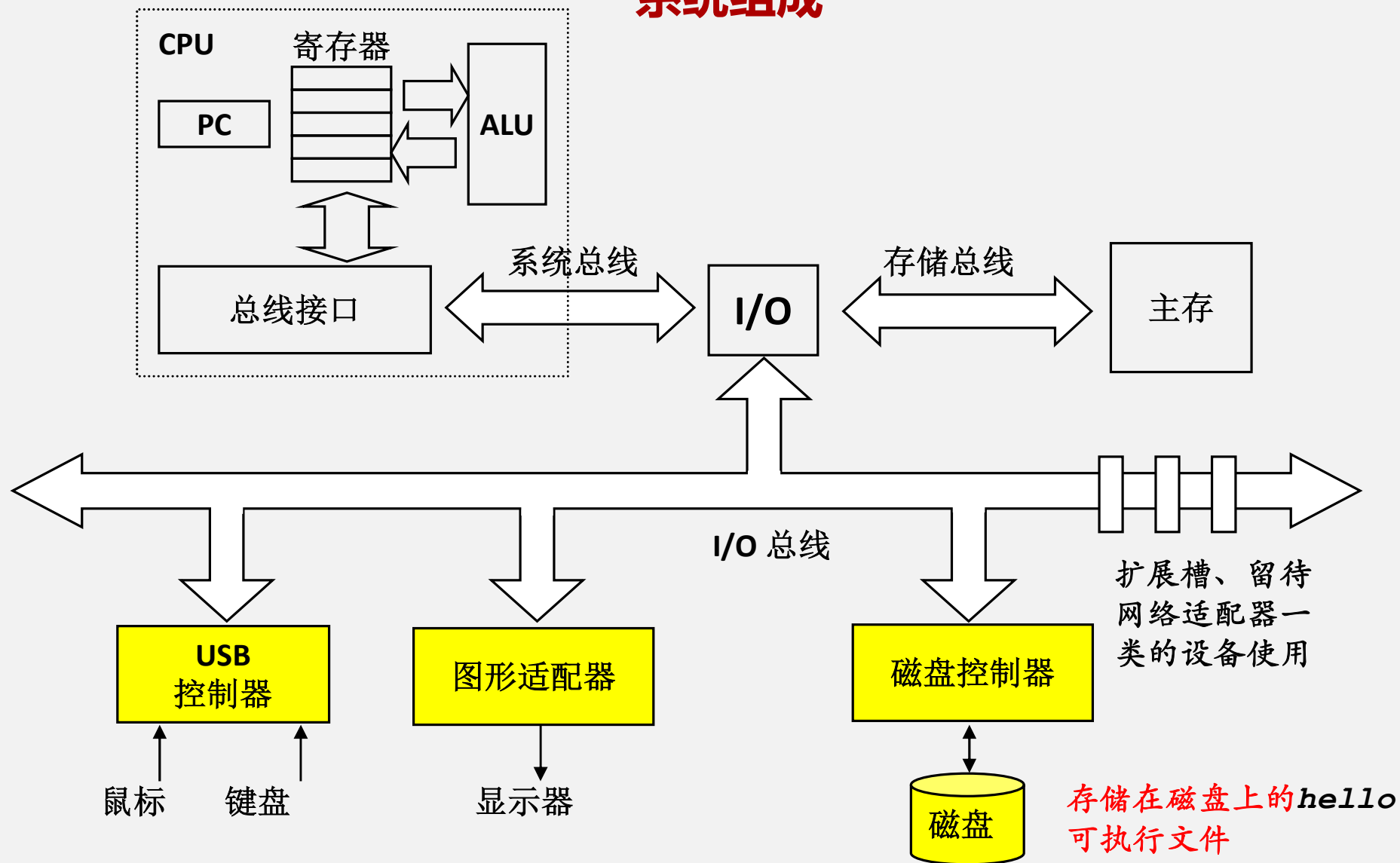
## 系统组成



## 系统组成

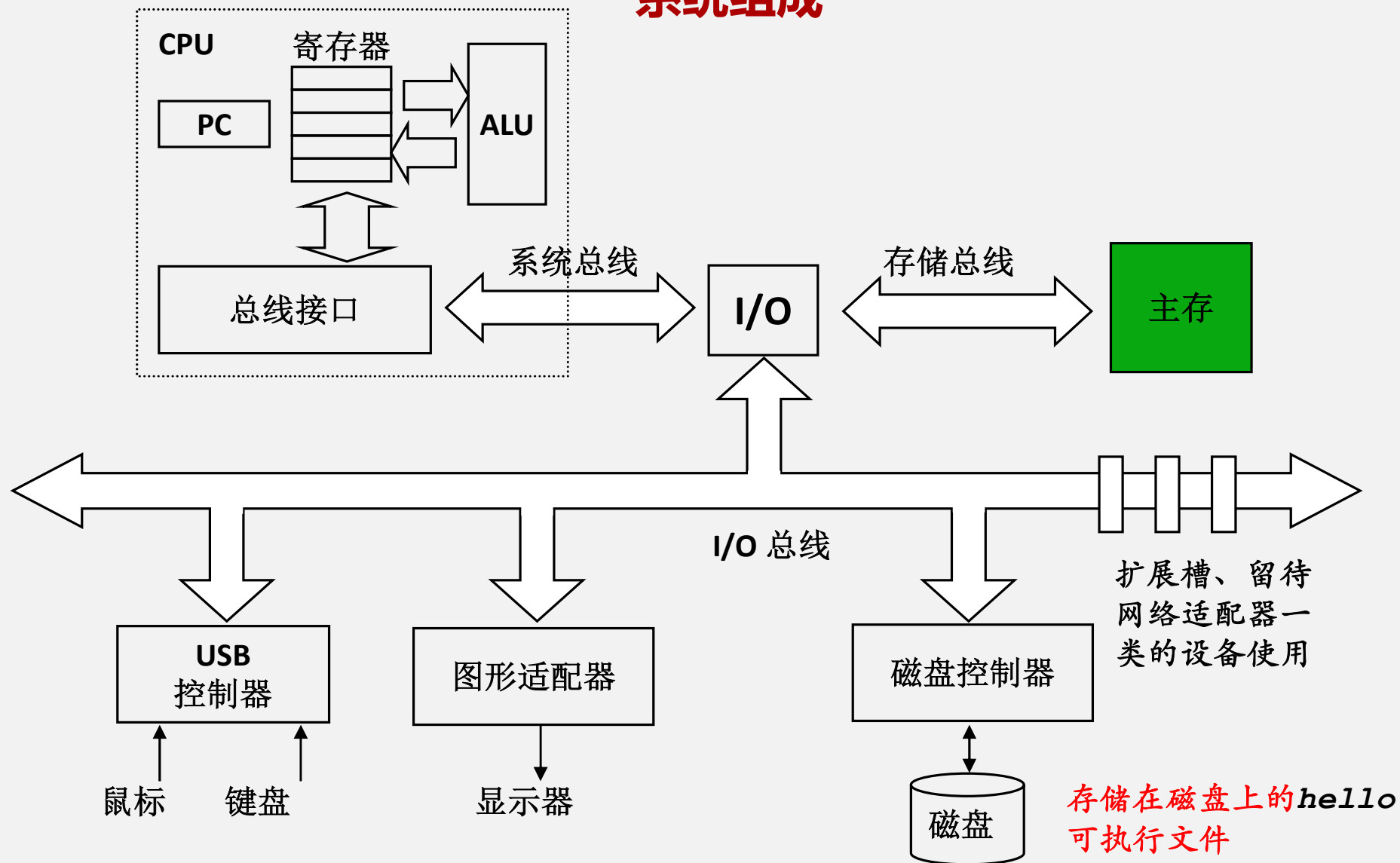


## 系统组成

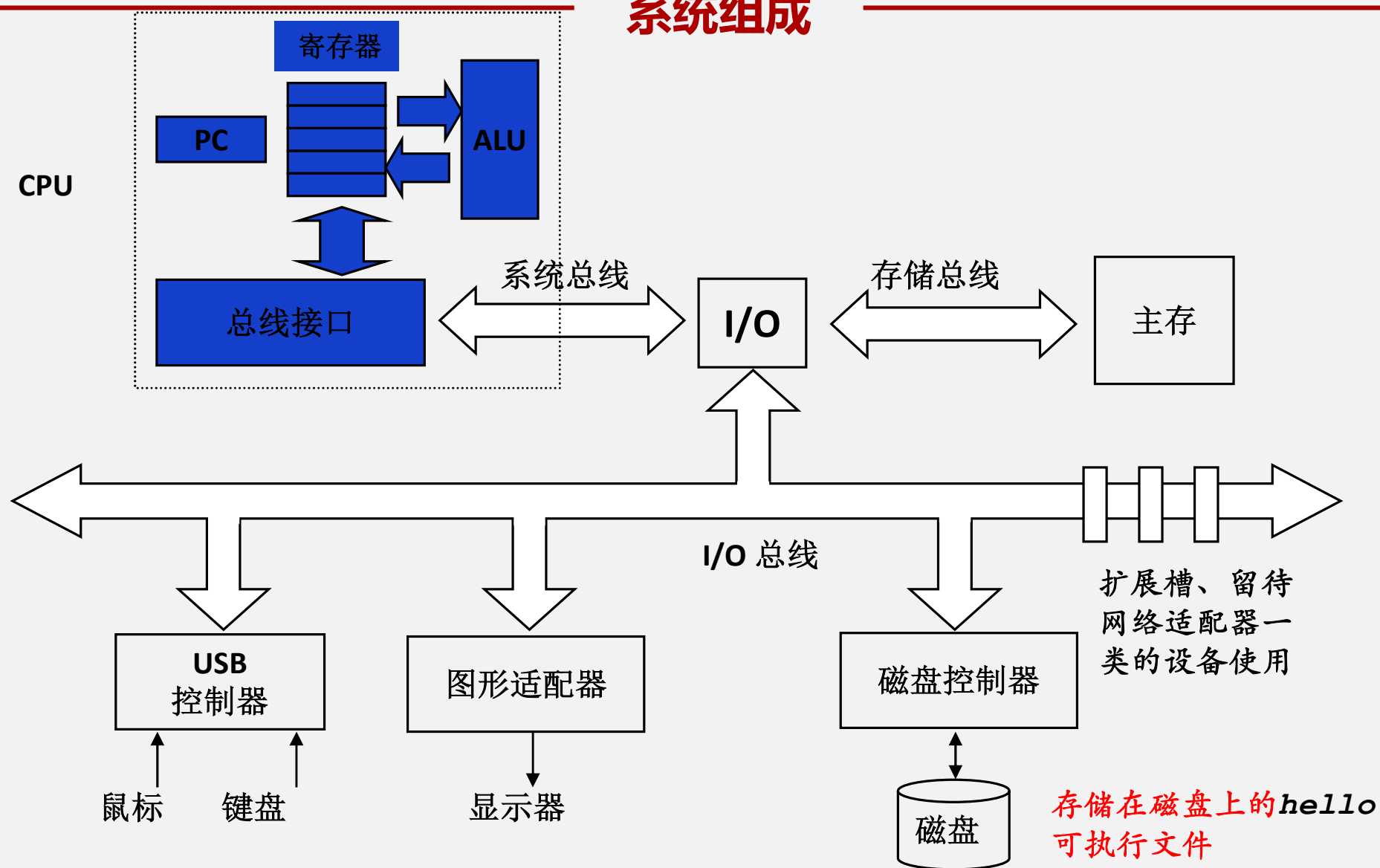




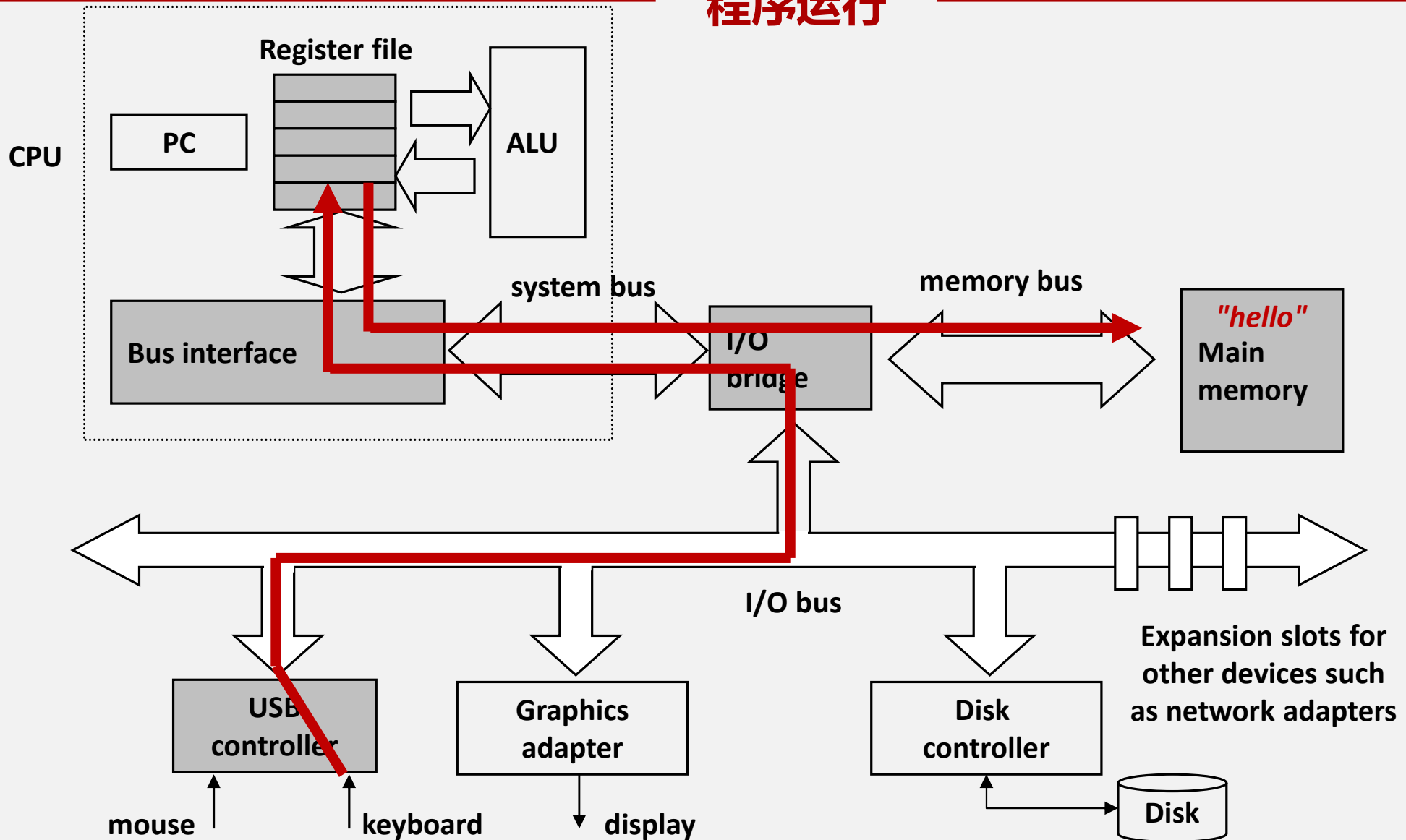
## 系统组成



## 系统组成



## 程序运行





# 下一节：汇编初步

湖南大学

《计算机系统》课程教学组

