1、数据类型

（2）浮点类型

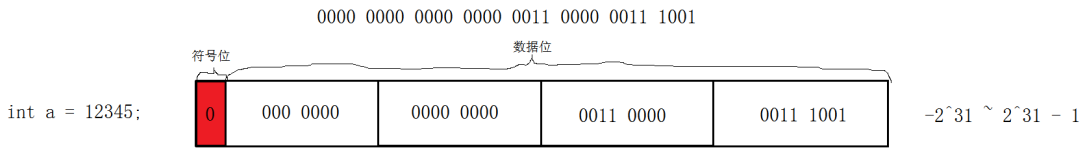
数据类型	名称	在内存上所占空间大小(单位：字节)	有效小数位
float	单精度浮点型	**4**	**4-5/5-6**
double	双精度浮点型	**8**	**14-15/15-16**

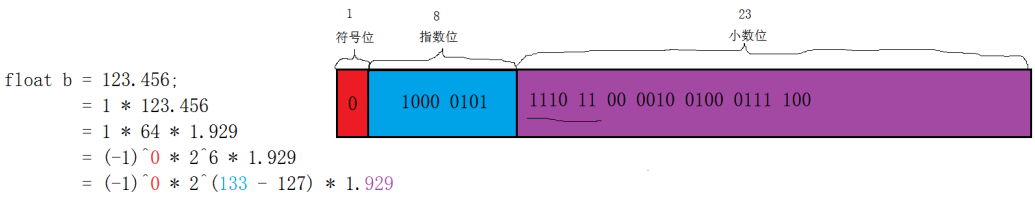
注：① **float**与**int**同为**4**字节，但由于存储方式不同，**float**值域远大于**int**值域

②浮点型数遵循**IEEE754**原则存储，因此不能准确表示数据，存在模糊数位，所以为范围值

③**float**型存储时，符号位**1bit**，指数位**8bit**，其余23位为小数位

double型存储时，符号位**1bit**，指数位**16bit**，其余47位为小数位





0.929 * 2 = 1.858　1　0.432 * 2 = 0.864　0
0.858 * 2 = 1.716　1　0.864 * 2 = 1.728　1
0.716 * 2 = 1.432　1　0.728 * 2 = 1.456　1。。。。

(-1) ^m*2^(n-127)*1.s

m表示符号位，0是正数，1是复数；n为指数位，以二进制表示；s需要转换成二进制形式表示，但需要有效数位个数

笔试题：请比较整型、浮点型、指针与“零值”的大小

整型：　　浮点型：　指针：

int a;　　**float b; int *p;**

if(a == 0)　　if(b> -0.000001 && b < 0.000001) if(p==NULL)

"0值"更换成其他a：float if(b> a-0.000001 && b < a+0.000001)

（3）逻辑类型

数据类型	名称	在内存上所占空间大小(单位：字节)	头文件
bool	波尔类型	1	#include <stdbool.h>
_Bool			

注：①逻辑类型只有两个值：**true**和**false**，其中**true**表示真，**false**表示假

② 逻辑类型的值一般使用**0**和**1**表示，其中**1**为非**0**值

（4）void *******

void不是数据类型，无法确定变量存储的内存大小；**void**可以代替数据类型完成一定功能

2、常量

不可改变的量

5　a yes　34.56

（1）整型常量

3456　03456　0x3456

10010111001　01010101010　0x101001010

注：①八进制常量以**0**作为标记开头，十六进制以**0x**作为开头

② 二进制不能用于显示

（2）浮点型常量

34.5678　1.23e-4

注：**1.23e-4 = 1.23*10^(-4)**

（3）字符常量 参考**ASCII**表

‘a’　‘A’　‘4’　‘\n’　‘$’　‘\77’　‘\x58’　‘\\’（显示 ****）

注：①所有字符都必须使用 ‘ ’ 括起来

② ‘ ’ 中包含 **** 为转义字符（使用 **** 将原字符含义进行改变）

③ 在一定范围字符可以和整型交换使用

④ 转义字符可以转义数字，若转义**8**进制数字，最多只能是**3**位**8**进制数；若转义**16**进制数，最多只能是**2**位**16**进制数，并且需要携带**x**标记

⑤ 在” “ 中显示%，需要使用 %%

```
printf("%c\n",'a'); //a
printf("%c\n",'4');//4
printf("%c\n",'\n');//换行
printf("%c\n",'$');//$
printf("%c\n",'\110');//1*8^2+1*8+0=72 H
printf("%c\n",65);//a
printf("%d\n",'a');//97
printf("%c\n",'\x58');//5*16+8=88 /X
printf("%c\n",'\');// \
//printf("%c\n",'%');// %
printf("%%\n");// %
```

ASCII表

（ American Standard Code for Information Interchange 美国标准信息交换代码 ）

高四位 / 低四位		ASCII控制字符												ASCII打印字符																			
		0000						0001						0010		0011		0100		0101		0100		0111									
		0						1						2		3		4		5		6		7									
		十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl							
0000	0	0		^@	NUL	\0	空字符	16	▶	^P	DLE		数据链路转义	32		48	0	64	@	80	P	96	`	112	p								
0001	1	1	☺	^A	SOH		标题开始	17	◀	^Q	DC1		设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q								
0010	2	2	☻	^B	STX		正文开始	18	↕	^R	DC2		设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r								
0011	3	3	♥	^C	ETX		正文结束	19	‼	^S	DC3		设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s								
0100	4	4	♦	^D	EOT		传输结束	20	¶	^T	DC4		设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t								
0101	5	5	♣	^E	ENQ		查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u								
0110	6	6	♠	^F	ACK		肯定应答	22	▬	^V	SYN		同步空闲	38	&	54	6	70	F	86	V	102	f	118	v								
0111	7	7	•	^G	BEL	\a	响铃	23	↨	^W	ETB		传输块结束	39	'	55	7	71	G	87	W	103	g	119	w								
1000	8	8	◘	^H	BS	\b	退格	24	↑	^X	CAN		取消	40	(	56	8	72	H	88	X	104	h	120	x								
1001	9	9	○	^I	HT	\t	横向制表	25	↓	^Y	EM		介质结束	41	)	57	9	73	I	89	Y	105	i	121	y								
1010	A	10	◙	^J	LF	\n	换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z								
1011	B	11	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{								
1100	C	12	♀	^L	FF	\f	换页	28	∟	^\	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|							
1101	D	13	♪	^M	CR	\r	回车	29	↔	^]	GS		组分隔符	45	-	61	=	77	M	93	]	109	m	125	}								
1110	E	14	♫	^N	SO		移出	30	▲	^^	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~								
1111	F	15	☼	^O	SI		移入	31	▼	^-	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	⌂	^Backspace 代码：DEL							

注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入。

（4）字符串常量

“hello”

注：①所有字符串都必须使用” “ 括起来

② 所有字符串中都会以 **‘\0’** 作为字符串标记，’ **\0’** 不会显示到字符串上，但是需要占据保存字符串的空间**1**字节

‘a’ 与” a” ？不一样， ‘a’ 是字符，只有一个字节， “a” 是字符串，两个字节

（5）标识常量 -- 宏

格式：

#define 宏名 被替换内容

注：①宏定义一般写在头文件后，函数之外

② 宏名一般为大写，一旦宏定义完成，宏名即可完全替换被替换内容，符合标识符命名规则

③ 被替换内容可以是常量、变量、表达式、数组、指针、函数、结构体、数据类型、字符串等

④ 宏定义属于单纯替换，替换后与被替换内容完全相同

宏函数：

行为和外观都比较接近函数的宏定义

格式：

#define 宏函数名(形参列表) { 函数体; } //尽量写在这一行中,换行的话加'\',与注释相反

注：①宏函数名符合标识符命名规则

② 形参列表只有参数名，多个参数之间使用” ,” 分隔

③ 宏函数使用：宏函数名(实际参数); 实参个数必须与形参个数相同

④ 宏函数若需要多行显示，则需要在换行处出现折行符” \”，空行也需要折行符，折行符后不能出现其他符号或数据或空格

例子：两数求和

```
#include<stdio.h>
#define ADD(a, b) \
{\
int sum=a+b;\
printf("sum=%d\n",sum);\
}
int main()
{
int a=10,b=2;
ADD(a,b);
return 0;
```

}

3、变量

可以改变的量

（1）格式：

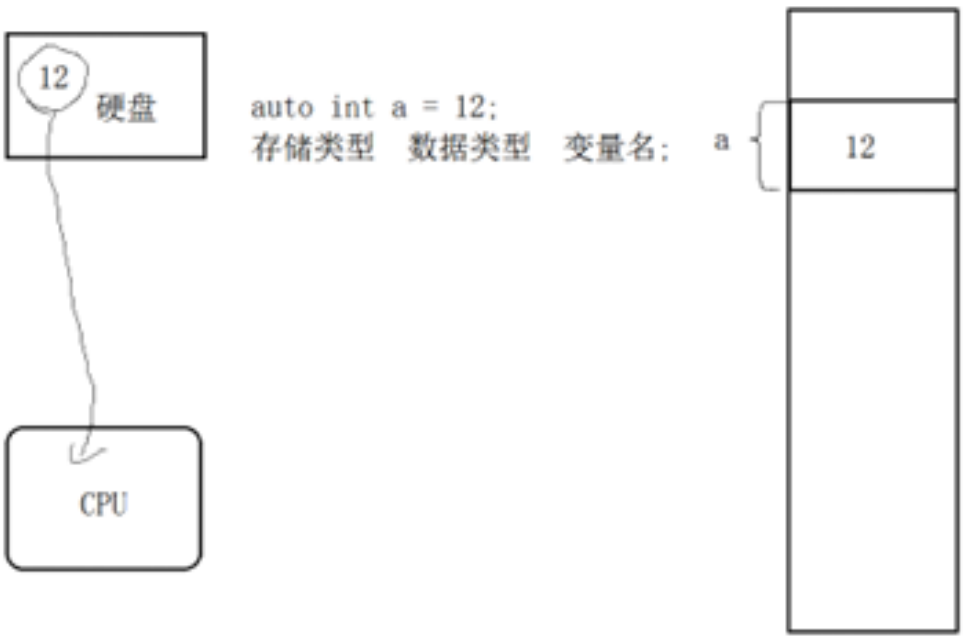
存储类型 数据类型 变量名;

注：①存储类型 表示数据在内存上的存储位置

② 数据类型 表示变量空间所保存的数据的类型以及变量空间的大小

③ 变量名 表示变量空间的名称，符合标识符命名规则

例：**auto int val;**



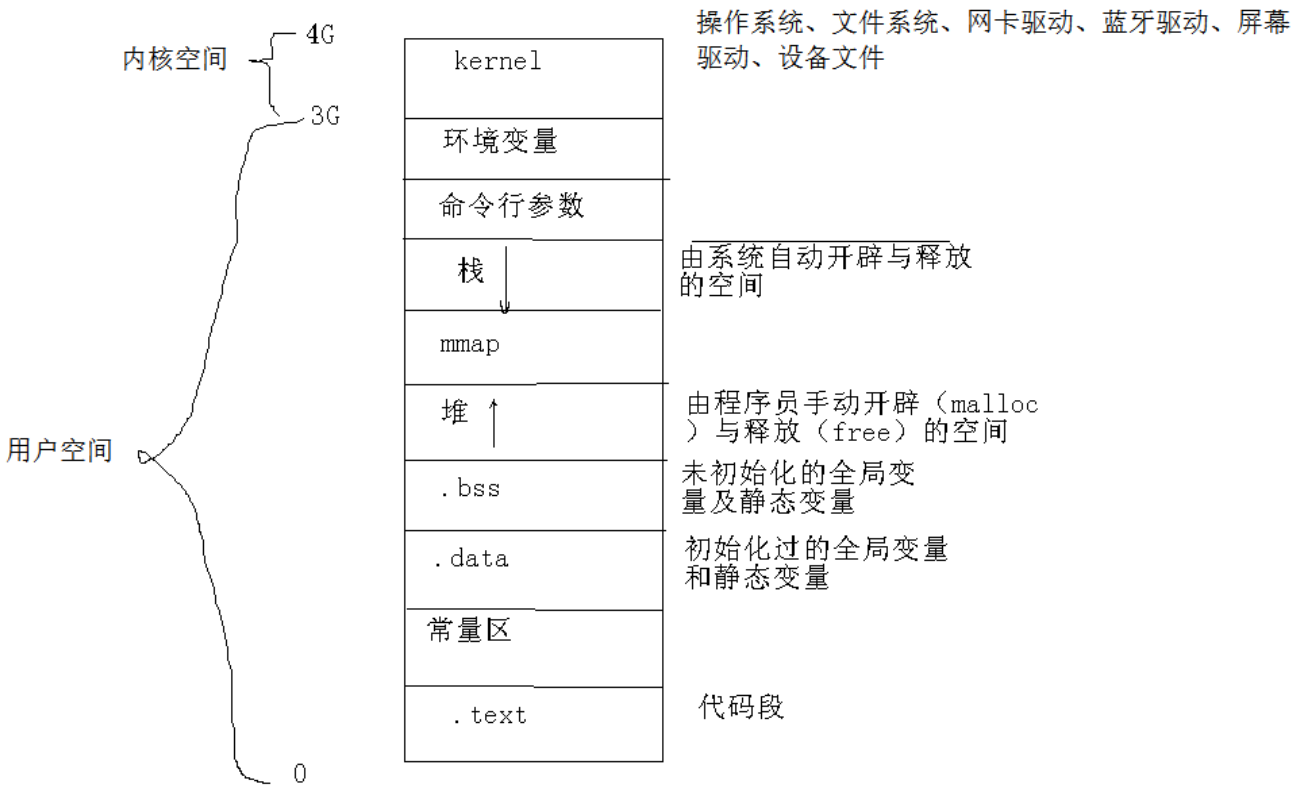
（2）初始化

定义变量的同时并给变量赋予初始值的过程被称为初始化，若未初始化，则由系统赋予随机值

（3）变量使用

①直接使用 变量名，可以获取或修改变量的值

②使用变量的地址 **&**变量名



（4）存储类型

<1> auto

① auto定义的变量被称为自动型变量

② auto定义的变量保存在内存的栈区，若未初始化由系统赋予随机值

③ auto定义的变量空间会随着变量的定义由系统主动申请开辟，随着变量在局部范围使用结束由系统回收释放

④ auto定义的变量，其存储类型可以省略

<2> static

<3> extern

extern修饰的变量被称为外部变量，可以调用其他文件中的全局变量或函数使用

<4> register

register修饰的变量被称为寄存器变量，若寄存器已满，则会自动退化为自动型变量

笔试题：

一，堆区与栈区的区别

1、栈区和堆区的区别：

1)申请方式： 栈区内存由系统**自动**分配，函数结束时释放；堆区内存由**程序员**自己申请，并指明大小，用户忘释放时，会造成内存泄露，不过进程结束时会由系统回收。
2)申请后系统的响应： 只要栈的剩余空间大于所申请的空间，系统将为程序提供内存，否则将报异常提示**栈溢出**；堆区，**空闲链表**，分配与回收机制，会产生碎片问题（外部碎片）-->（固定分区存在内部碎片（分配大于实际），可变分区存在外部碎片（太碎无法分配））。
3)申请大小的限制：栈是1或者2M，可以自己改，但是最大不超过8M；堆，看主机是多少位的，如果是32位，就是4G
4)申请效率：栈由系统自动分配，速度较快，程序员无法控制；堆是由new分配的内存，一般速度较慢，而且容易导致内存碎片，但是用起来方便!
5)存储内容：栈，函数调用（返回值，各个参数，局部变量（**静态变量不入栈**））；堆，一般在堆的头部用一个字节存放堆的大小，堆中的具体内容由程序员安排。
6)存取效率的比较：栈比堆快，Eg :char c[] = /"1234567890/";char *p =/"1234567890/";读取c[1]和p[1],c[1]读取时直接吧字符串中的元素读到寄存器cl中，而p[1]先把指针值读到edx中，再根据edx读取字符，多一次操作。
7)管理方式不同：栈，数据结构中的栈；堆，链表
8)生长方向：栈，高到低；堆，低到高
2、预习输入输出、强制类型转换、运算符