1全局变量=0；

在int和unsigned int 混合运算过程中，int 被转化成unsigned int ,-1---->2^32-1此时大于4；

2宏定义

3strlen,sizeof

4静态存储期块作用域静态变量，程序离开函数这些变量不消失，多次函数调用将会记录它的值，程序似乎跳过这条声明，静态变量和外部变量在被载入内存时就已经执行完毕

5移位

6\*+的优先级

7浮点数判断大小，精度丢失问题，强制转化是数值转化，

//两次相反的强转恢复原来的值？？？

//1.25在IEEE标准下因为除得尽，可以根据余127码和余1023码互相转化没问题！

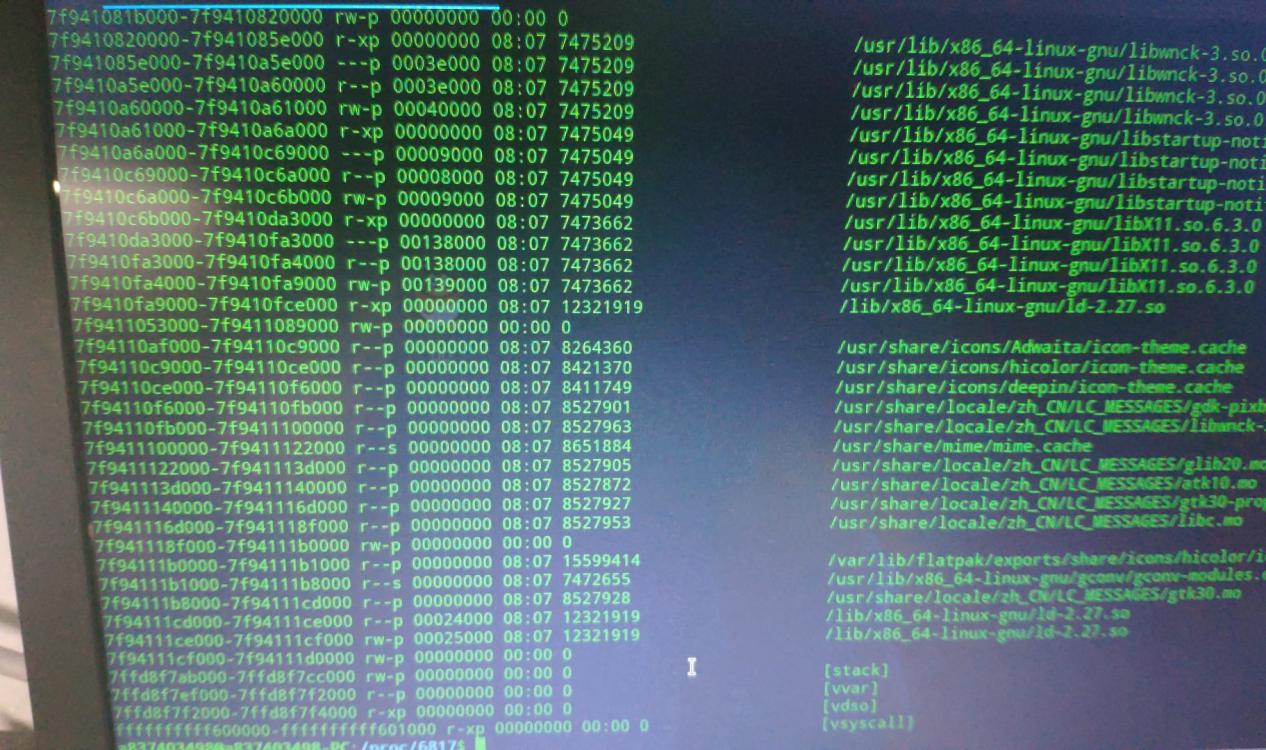
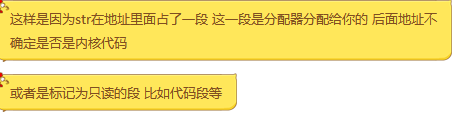
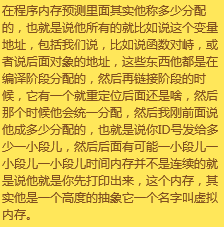
3.14二进制除不尽，数值储存出现偏差（余127码和余1023码下存的值不同）

8 (char\*)强制转化，0x6f796958--->32位二进制，char8位8位读，读到Xiyo,在数值上从右往左读（58->X）(69->i)(79->y)(6f->o),在读数组a[6]右边先入栈，左边后入，先入先出，即先读到栈顶的a[0]再依次读。Printf(“%s”,...)的返回值是字符串中的字符数；

9数组的初始化

10（char\*）的强制转化，拆出（数值）后8位（1字节）而不是地址，

11

P7G8783HOH~I{X(SAQ3TPFY

字符串常量存储在常量池，其中的东西不可以被修改，用指针指向的就是此处的字符串常量，然而用数组的话修改的是栈中字符串的副本。P326,个别编译器处理起来会出问题。因为用指针标记了字符串常量之后，会影响所有使用该字符串的代码。比如你想打印字符串常量printf“asd”，由于你修改了s--->r打出来是ard，这很畸形对不？所以不允许。用数组的话修改的是栈（动态内存）中字符串的副本，对静态内存中的字符串常量就没影响喽。

Const 并不是要原数组是常量，而是在函数处理数组时把数组视为常量。Const仅仅保证不可以显式改变变量的值。（嵌套指针，多重解引用可能会。。。P305,p306）

12gcc -E hello.c -o hello.i .c-->.i 预编译

gcc- S hello.i-o hello.s .i-->.s 编译

gcc -c hello.c -o hello.o .s-->.o 汇编形成目标文件

gcc hello.o -o hello .o--->链接 可执行文件

./hello 执行

编译器的工作是将源代码(通常使用高级语言编写)翻译成目标代码

13冒泡优化（1.前后各冒泡）（2.标记是否已排好了）（3.标记上次排的最后一次的位置）