**指针强化**

1. 指针的步长
   1. 具体看我们定义的类型是什么
   2. 如果是 int 则步长是 4 个字节
   3. Char 则是 1 个字节
   4. 代码

void pointerTest(){

char buff[1024] = {0};

int a = 100;

memcpy(buff+1, &a, sizeof(int)); // 把a的地址拷贝到stuff的第二个位置上

// 那么现在我们想取出来，我们知道int 是4个字节

char \*p = buff;

// 我们是从第二位开始

++p;

printf("a = %d\n", \*(int\*)(p)); 我们拿到首地址，但是我们还是需要知道我们需要的数字的类型是什么，才能知道取地址的几位，对应的值，我们结构体上有一个offsetof(结构体, 成员) 就能知道偏移是多少的

}

1. 字符串的copy（有n中方法）我们只举例一种就好了
   1. 代码

void stringCopy(char \*from, char \*to){

while(\*from != '\0'){

\*to++ = \*from++;

}

\*to ='\0';

}

void test01(){

char buff[1024] = {0};

char origin[1024] = "i am weierLin";

stringCopy(origin, buff);

printf("%s\n", buff);

}

1. 字符串反转
   1. 代码

void stringReverse(char \*str){

if(str == NULL){

return;

}

int len = strlen(str);

int i = 0;

int j = len-1;

while(i++ < j--){

char ch = \*(str+i);

\*(str+i) = \*(str+j);

\*(str+j) = ch;

}

}

void test01(){

char origin[1024] = "i am weierLin";

stringReverse(origin);

printf("%s\n", origin);

}