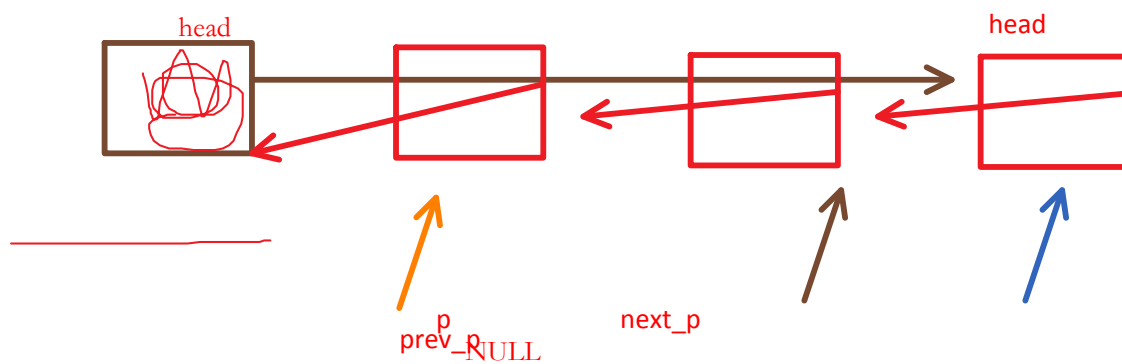


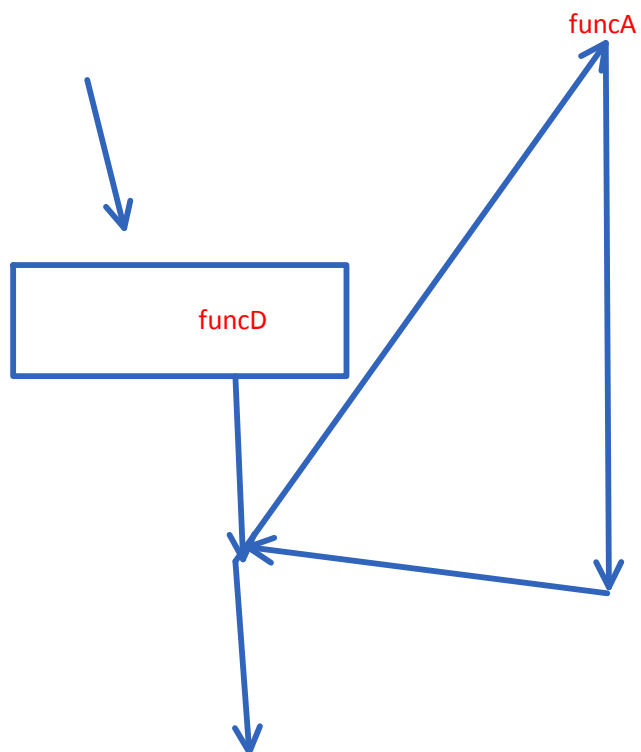
1 逆序一个单向链表

2015年11月23日 9:20

从右往左遍历 显然需要n个指针
不是好方法



有头，



在调用funcD的内部

间接的调用了外部的funcA函数

这种调用函数的方式
就是回调。

3 函数指针的练习

2015年11月23日 10:51

0xffffffff



void main()

知道这个函数体的入口是0地址 但是函数名 他找不到了。

(void (*)(void))0 //将0地址转换成一个 函数指针 void (*) (void)

```
fp
fp();
(*fp());
(void (*)(void))0 ==== fp  (*fp)(), fp()
```

(* (void (*)(void))0)() //将0地址转换成一个函数指针， 再调用

(int*)(0)

```
fp
fp();
(*fp)();
```

0x00000000



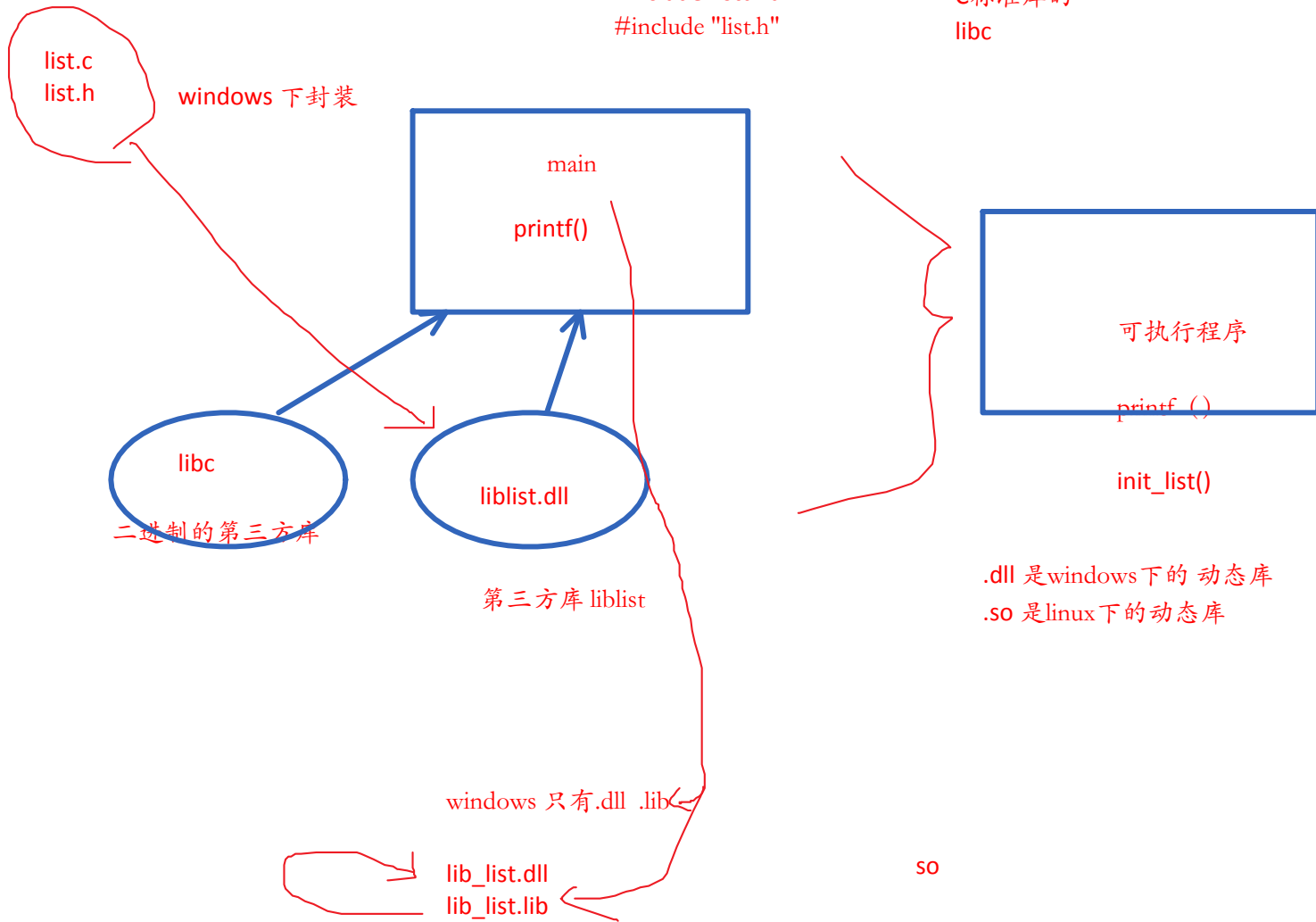
```
函数体 void func(void)
{
    函数的实现
}
```

4 windows动态库的封装

2015年11月23日 16:03

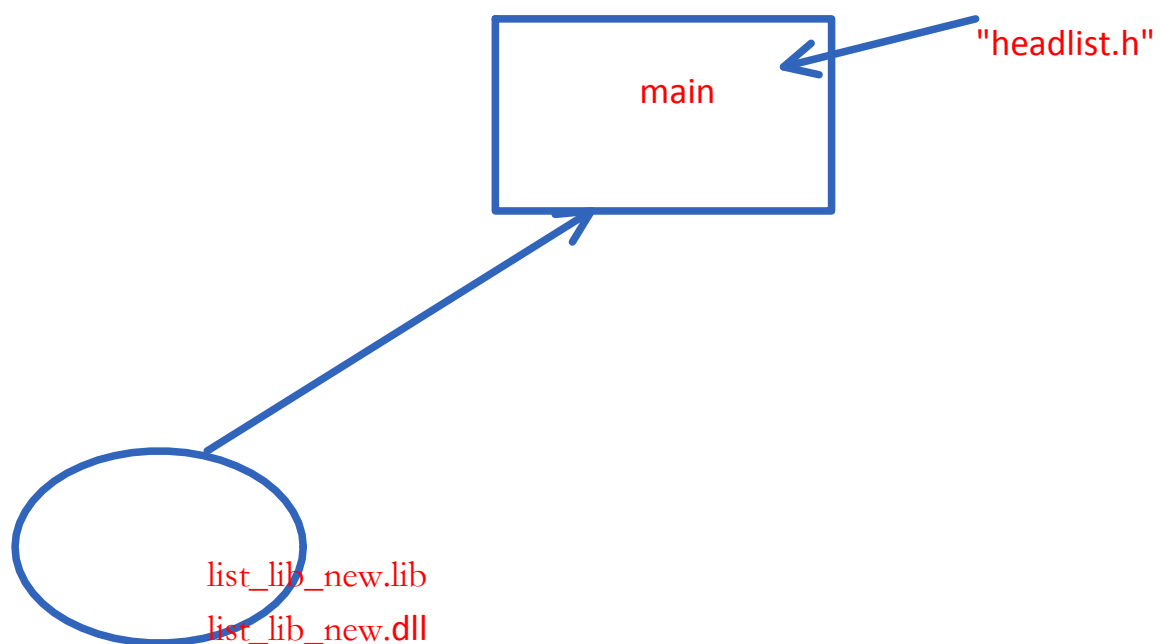
```
#include <stdio.h>
#include <stdlib.h>
#include "list.h"
```

C标准库的
libc



5 系统调用动态库

2015年11月23日 16:44



valgrind

内存检测工具

2015年11月23日 17:18

malloc c-标准库

去将malloc函数封装以下 每开辟一个内存我

每调用malloc一次 都会给这个内部的链表加一个节点，（纪律开辟内存低地址 空间长度）



每次调用free 就会从对象的地址的内存空间从链表拆除。

开辟内存低地址