

####C Trick

#####BASIC

....

scanf("%[^\\n]", str); 回车才返回

printf中 %zu 可输出size_t类型且不报警告

...

>底层是指针实现

a[5] == 5[a];

"ABCD"[2] == 2["ABCD"] == 'C';

相当于*(a + 5) == *(5 + a)

...

```
#include <stdio.h>
int main(void){
    int x = 5;
    printf("%d and ", sizeof(x++)); // note 1
    printf("%d\\n", x); // note 2
    return 0;
}
```

result: 4 and 5, 因为 sizeof 是编译时行为，运行时不会执行

...

#####MACRO

>##pragma once**是编译器相关的，有的编译器支持，有的编译器不支持，具体情况请查看编译器API文档，不过现在大部分编译器都有这个杂注了。
#ifndef, #define, #endif是C/C++语言中的宏定义，通过宏定义避免文件多次编译。所以在所有支持C++语言的编译器上都是有效的，如果写的程序要跨平台，最好使用这种方式。

>下划线是变量

```
_(__, __);
__[__];
____=&__;
```

#####GNU C EXTEND

> gcc的C扩展支持嵌套函数（函数里可以再定义函数），clang的C好像还有闭包..

```
...
```

```
#include <stdio.h>
```

```
static const unsigned char a[] = {
```

```
#include "values.txt"
};
```

```
int main(int argc, char *argv[])
{
    int i;

    for (i = 0; i < sizeof(a); ++i)
        printf("a[%d] = %d\n", i, a[i]);
    return 0;
}
```

```
values.txt
1,2,3,4,5,
6,7,8,9,0,
```

```
...
```

```
---
```

```
...
```

来自Linux Kernel的例子，一般用得比较少

```
static struct usb_driver usb_storage_driver = {
    .owner = THIS_MODULE,
    .name = "usb-storage",
    .probe = storage_probe,
    .disconnect = storage_disconnect,
    .id_table = storage_usb_ids,    };
...
----
```

>匿名数据结构：匿名数组、匿名结构体等。下面我们来实现Lisp的List

```
...
```

```
int main() {
    struct mylist {int a; struct mylist* next;};
    #define cons(x, y) (struct mylist[]){x, y}
    struct mylist *list = cons(1, cons(2, cons(3, NULL)));
    struct mylist *p = list;
    while(p != 0) {
        printf("%d\n", p->a);
        p = p->next;
    }
}
```

```
...
```

```
----
```

```
#####TRICK
```

>C++1x 可以用 auto 来省略类型声明, 而 C 的 auto 关键字意思完全不同但明显是个废话(auto 的意思是非 static, 和不写一样), 还好有 C 有 __typeof__

>异或实现双链表 https://en.wikipedia.org/wiki/XOR_linked_list

#####STUDY SOURCE

...

duff's device

https://en.wikipedia.org/wiki/Duff%27s_device

trigraph

http://blog.sina.com.cn/s/blog_4b687eaco1008ice.html

...