

## 真 OO无双 之 真乱舞书

寫程式是很快樂的一件事 Since Sep.15,2006

(原創) 簡單的Linked List實現 (C/C++) (C) (Data Structure)

### Abstraction

使用C語言簡單的實現linked list，並用C++的std::vector實作出相同的功能作比較。

### Introduction

學習資料結構，第一個要學的就是linked list，本文示範最簡單的linked list實現，包含建立與顯示，可把它當成linked list的標準範本，畢竟步驟都差不多。

一個基本的問題：為什麼需要linked list？若要將大量資料存到記憶體，你會想到什麼？第一個想到的就是array，但C語言是個靜態語言，array必須事先宣告大小，這樣compiler才能進行最佳化，若到時候沒用這麼多記憶體，就白白浪費記憶體了。或許你會說array可以配合malloc()變成動態array，但前提是你必須告訴malloc()要建立多大的array，若連要建立多大的陣列也不確定，而是在run-time看有多少資料就建立多大，這時連malloc()的動態array也不能用了，此時就得靠linked list。

linked list是資料結構的基礎，基本上就是靠struct如火車車廂那樣連在一起，run-time有需要時，在動態掛一個struct上去。

### C語言

```
1 /*
2 (C) OOMusou 2008 http://oomusou.cnblogs.com
3
4 Filename   : DS_linked_list_simple.c
5 Compiler   : Visual C++ 8.0
6 Description : Demo how to use malloc for linked list
7 Release    : 03/22/2008 1.0
8 */
9 #include <stdio.h>
10 #include <stdlib.h>
11 #include <string.h>
12
13 #define SLEN 255
14
15 struct list {
16     int no;
17     char name[SLEN];
18     struct list *next;
19 };
20
21 int main() {
22     int no;
23     char s[255];
24
25     struct list *head = NULL;
26     struct list *current = NULL;
27     struct list *prev = NULL;
28
29     while(1) {
30         printf("No. = ");
31         scanf("%d", &no);
32
33         if (no == 0)
34             break;
35
36         printf("Name = ");
37         scanf("%s", s);
38
39         current = (struct list *)malloc(sizeof(struct list));
40         if (current == NULL)
41             exit(EXIT_FAILURE);
42
43         current->next = NULL;
44
45         current->no = no;
46         strncpy(current->name, s, SLEN - 1);
47         current->name[SLEN - 1] = '\0';
48
49         if (head == NULL)
50             head = current;
```

导航

博客园

首页

新随笔

联系

订阅 XML

管理

公告



我是一個小小的數位IC工程師，從事SOC IP開發，業餘則喜歡研究FPGA、Embedded System、OS、MFC、NET與OOP相關技術。我並不是靠寫Blog或寫書維生，只是記下來怕自己忘記，所以不加任何廣告。所有文章與程式碼歡迎轉載使用。

昵称: 真 OO无双

园龄: 7年2个月

粉丝: 616

关注: 0

+加关注

随笔分类(2095)

μC/OS-II(9)

.NET(112)

ARM(1)

ASP.NET(9)

boost(7)

C(81)

C#(46)

C/C++(287)

C++/CLI(33)

C2H(3)

CentOS(12)

CSS(7)

Data Structure(1)

Database(12)

DE2(56)

DE2-70(42)

Debussy(6)

Design Pattern(22)

DirectX(6)

DSP Builder(4)

English(4)

Excel(1)

FireFox(1)

g++(3)

gcc(4)

GDI+(15)

HTML(3)

IE(3)

Image Processing(35)

iOS(2)

iPad(2)

iPhone(7)

ISE(1)

iTunes(1)

Japanese(3)

Java(15)

JavaScript(9)

Linux(42)

```

51  else
52      prev->next = current;
53
54  prev = current;
55  }
56
57  // display linked list
58  current = head;
59  while(current != NULL) {
60      printf("No. = %d, Name = %s\n", current->no, current->name);
61      current = current->next;
62  }
63
64  // free linked list
65  current = head;
66  while(current != NULL) {
67      prev = current;
68      current = current->next;
69      free(prev);
70  }
71 }

```

執行結果

```

No. = 1
Name = clare
No. = 2
Name = jessie
No. = 0
No. = 1, Name = clare
No. = 2, Name = jessie

```

15行

```

struct list {
    int no;
    char name[SLEN];
    struct list *next;
};

```

linked list的基礎就是struct，所以先建立一個自訂的struct型別，因為linked list是靠struct串聯起來，所以最後要多一個struct pointer指向下一個struct。

25行

```

struct list *head = NULL;
struct list *current = NULL;
struct list *prev = NULL;

```

建立linked list最基本需要三個指標，head指向linked list的第一個struct，current指向目前剛建立的struct，prev則指向前一個struct，目的在指向下一個struct，對於未使用的pointer，一律指定為NULL。這是一個好的coding style，可以藉由判斷是否為NULL判斷此pointer是否被使用。

39行

```

current = (struct list *)malloc(sizeof(struct list));
if (current == NULL)
    exit(EXIT_FAILURE);

current->next = NULL;

```

每當有新資料，需要建立一個新的struct時，就用malloc()要一塊記憶體，由於malloc()傳回的是void \*，所以要手動轉型成struct list \*。但malloc()並不是一定會成功，若記憶體不足時，仍然會失敗，所以必須判斷是否傳回NULL。

由於一個新的node，一定是linked list最後一個node，所以將current->next接null。

45行

```

current->no = no;
strncpy(current->name, s, SLEN - 1);
current->name[SLEN - 1] = '\0';

```

正式將輸入的資料填進struct，至於為什麼要用strncpy()而不用strcpy()呢?雖然strcpy()也可以，但strncpy()比較安全，若輸入的字串大小超過struct所定義的字串大小，則會只接受struct

Matlab(4)  
MegaCore (10)  
Misc(20)  
ModelSim(13)  
NB(34)  
NC-Verilog(4)  
NEEK  
News(77)  
Nios II(76)  
nLint(1)  
N-Tier(7)  
OO(47)  
OpenMP(2)  
OS(59)  
P7010(4)  
PetShop(6)  
Qsys(6)  
Quartus II(67)  
RedHat(3)  
RVDS(1)  
SignalTap II(6)  
SOC(257)  
SOPC Builder(39)  
SQL Server(5)  
STL(41)  
Synplify(1)  
SystemC(8)  
Tcl(1)  
template(27)  
ThinkPad(30)  
TMP(2)  
TRDB-D5M(8)  
TRDB-DC2(5)  
TRDB-LCM(4)  
TRDB-LTM(7)  
T-SQL(4)  
Ubuntu(8)  
UltraEdit(4)  
UML(5)  
VC++(10)  
VCS(1)  
Verdi(3)  
Verilog(74)  
Verilog PLI(7)  
VHDL(4)  
VirtualBox(4)  
Visual BASIC(1)  
Visual FoxPro(8)  
Visual Studio(9)  
VMWare(4)  
Web(26)  
Win32 API  
Windows(21)  
Windows CE(2)  
Windows Form(2)  
Windows Live Writer(5)  
Word  
X300(5)  
X61(5)  
XML(1)  
µClinux(3)  
翻譯(3)  
美食(3)  
日記(68)  
推薦軟體(5)  
中森明菜(11)

所接受的字串大小，而不會因為找不到'\0'而造成程式錯誤。

49行

```
if (head == NULL)
    head = current;
else
    prev->next = current;

prev = current;
```

判斷若是第一個node，則將目前的node當成head，若不是第一個node，則將前一個node指向日前的node，完成linked list的連接。最後將目前的node當成前一個node，以備指向下一個node。

58行

```
// display linked list
current = head;
while(current != NULL) {
    printf("No. = %d, Name = %s\n", current->no, current->name);
    current = current->next;
}
```

要重新顯示linked list，所以將指標再度指向第一個node，每當顯示一個node後，就指向下一個node，直到指到NULL為止。

64行

```
// free linked list
current = head;
while(current != NULL) {
    prev = current;
    current = current->next;
    free(prev);
}
```

由於malloc()是將記憶體放在heap，而不是放在stack，所以並不會隨著function的結束而釋放，必須要手動使用free()釋放記憶體，否則會造成memory leak。

再來看C++，由於STL已內建一些容器，所以不需再重新實作linked list，有兩個選擇：

std::vector或者std::list。std::vector的優點是non-sequential access超快，新增資料於後端超快，但insert和erase任意資料則超慢；std::list則剛好相反，insert和erase速度超快，但non-sequential access超慢，由於本例只有新增與non-sequential access，所以適合std::vector。不過由於STL使用泛型技術，若將來需求改變，想改用std::list也沒關係，只要將容器改掉即可，剩下的都不用改，因為STL的演算法並不挑容器，這正是泛型偉大之處。

**C++**

```
1 /*
2 (C) OOMusou 2008 http://oomusou.cnblogs.com
3
4 Filename   : DS_linked_list_simple_vector_class.cpp
5 Compiler   : Visual C++ 8.0
6 Description : Demo how to use vector instead of linked list
7 Release    : 03/22/2008 1.0
8 */
9 #include <iostream>
10 #include <string>
11 #include <vector>
12
13 using namespace std;
14
15 class List {
16 public:
17     int no;
18     string name;
19 };
20
21 int main() {
22     vector<List> vec;
23
24     while(1) {
25         List list;
26         cout << "No. = ";
27         cin >> list.no;
28
29         if (list.no == 0)
```

```
30     break;
31
32     cout << "Name = ";
33     cin >> list.name;
34
35     vec.push_back(list);
36 }
37
38 vector<List>::iterator iter = vec.begin();
39 for(; iter != vec.end(); ++iter)
40     cout << "No. = " << iter->no << ", Name = " << iter->name << endl;
41 }
```

執行結果

```
No. = 1
Name = clare
No. = 2
Name = jessie
No. = 0
No. = 1, Name = clare
No. = 2, Name = jessie
```

15行由struct改用了class，不過若繼續使用struct亦可，至於其他的程式都很直觀，就不再多做解釋。

### Conclusion

本文主要是討論使用C語言透過malloc()實現資料結構的linked list，以彌補靜態語言的不足，同時亦討論C++使用STL的替代方案與便利性，C與C++各擅勝場，你可自行決定使用C或C++。

### See Also

(原創) C/C++哪些地方會用到pointer呢? (C/C++) (C)

(原創) 如何將輸入的字串存到記憶體後，再一起印出來? (C/C++) (C)

分类: [Data Structure](#), [C/C++](#), [C](#)

绿色通道: [好文要顶](#) [关注我](#) [收藏该文](#) [与我联系](#) 



真 OO无双  
关注 - 0  
粉丝 - 616

+加关注

0 0

(请您对文章做出评价)

« 上一篇: (轉貼) 包道格: 離美1年未用 綠卡就失效 (News)

» 下一篇: (原創) 二維陣列與字串陣列有什麼差異? (C/C++) (C)

posted on 2008-03-22 15:26 真 OO无双 阅读(58848) 评论(5) 编辑 收藏

评论

### #1楼 2008-04-22 11:43 Rogers[未注册用户]

版主大人，您好：  
感謝您發表的文章，  
讓小弟我獲益良多...  
小弟我是C語言的初學者(自學者)，  
您這一篇介紹linked list的code，  
似乎並沒有使用暫停如system("pause");的函數??

是不是這樣，  
所以我compile Run之後無法看到顯示的結果？  
如：

```
No. = 1
Name = clare
No. = 2
Name = jessie
No. = 0
No. = 1, Name = clare
No. = 2, Name = jessie
```

### #2楼 2008-04-22 11:53 Rogers[未注册用户]

承上文，(使用C語言)  
所以我自行將system("pause");加入到第61行 或是 第69行，  
顯示出來的結果皆不盡理想...  
會隨著使用者輸入no.數目的多寡一併列印出同數目的：請按任意鍵繼續

(小弟我使用Dev-C++)

謝謝 感恩

#3楼[楼主] 2008-04-22 13:50 真 OO无双

@Rogers

若用VC6, VC7, VC8, 則使用ctrl + F5執行即可  
不需加上system("pause"),自己會暫停

我沒使用Dev C++  
所以不清楚Dev C++, 不好意思

支持(0) 反对(0)

#4楼 2008-04-26 07:54 littlepotato[未注册用户]

其實可以用 c99 或gnu extension: 在更早以前就有可變長度的 trick, 方法是用 struct:

```
struct aa {  
int size;  
char data[0];  
};  
  
func(len)  
{  
struct aa *ptr;  
ptr = malloc(sizeof(struct aa) + sizeof(char)*len);  
...  
}
```

gcc -0 aa -std=c99 aa.c

--

用 gnu extension 比較方便。

#5楼 2009-06-26 13:31 小魔女[未注册用户]

感謝你的文章，  
簡單易懂好修改。  
謝謝。

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

[博客园首页](#) [博问](#) [新闻](#) [闪存](#) [程序员招聘](#) [知识库](#)



最新IT新闻:

- 三星和HTC设备因作弊被剔除出Futuremark排名
  - 腾讯回应360公开信: 做企业不是比谁的嗓门大
  - 淘宝网推“扫码购”服务 “双12”将送超亿红包
  - Canalys: 2014年平板电脑将超计算机出货量50%
  - 只有四分之一的雅虎雇员使用自家的Yahoo Mail
- » 更多新闻...

最新知识库文章:

- 对于现代开发来说, Javascript就是一种垃圾语言
  - 教你一眼认出英语单词的意思
  - 数学之美
  - 理解本真的REST架构风格
  - 工程师的生活
- » 更多知识库文章...

---

Powered by:

[博客园](#)

Copyright © 真 OO无双