

- Blog
- Paste
- Ubuntu
- Wiki
- Linux
- Forum

搜索

進入

搜索

- 頁面
- 討論
- 編輯
- 歷史
- 简体
- 繁體

- 導航
  - 首頁
  - 社群入口
  - 現時事件
  - 最近更改
  - 隨機頁面
  - 幫助
- 工具箱
  - 鏈入頁面
  - 鏈出更改
  - 所有特殊頁面
- 個人工具
  - 登入

# Mix C Cpp

出自**Ubuntu**中文

## 目錄

- 1 C與C++混合編程
- 2 C++調用C函數
- 3 C調用C++函數
- 4 其他參考

## C與C++混合編程

C++ 是在 C 語言的基礎上發展起來的。在某種程度上，我們可將 C++ 看做 C 的一種擴展。在本質上，二者的數據類型和函數調用慣例都是一致的，因此 C 與 C++ 混合編譯也是很自然的事情。

二者的區別僅在於編譯后函數的名字不同。C 簡單地使用函數名而不考慮參數的個數或類型，而 C++ 編譯后的函數名則總是將參數類型列表作為其一部分。儘管如此，C++ 提供了特殊的機制來聲明 C 函數，這意味着一個 C++ 程序可以直接聲明和調用 C 函數。

## C++調用C函數

下面是 C++ 程序調用 C 函數 `csayhello()` 的一個例子。由於該函數在 C++ 程序內聲明時使用了 `extern "C"`，故調用可以直接進行：

```
/* cpp2c.cpp */
#include <iostream>
extern "C" void csayhello(char *str);
int main(int argc, char *argv[])
{
    csayhello("Hello from cpp to c");
    return(0);
}
```

C 函數不需任何特殊處理，其代碼如下：

```
/* csayhello.c */
#include <stdio.h>
void csayhello(char *str)
{
    printf("%s\n", str);
}
```

下面三條命令編譯以上兩個文件並將二者鏈接為一個可執行文件。由於 `gcc` 和 `g++` 的靈活性使得存在很多方法來完成該任務，但這三條命令或許是最常用的：

```
$ g++ -c cpp2c.cpp -o cpp2c.o
$ gcc -c csayhello.c -o csayhello.o
$ gcc cpp2c.o csayhello.o -lstdc++ -o cpp2c
```

注意到，在最後鏈接的時候指定 C++ 標準庫是必須的，這是因為我們用的是 `gcc` 而不是 `g++` 調用的鏈接器。如果使用的是 `g++` 的話，C++ 標準庫默認會被鏈接。

最普遍的做法是，將函數聲明放到頭文件中，然後將所有內容包含在 `extern "C"` 聲明塊內。文件內容像下面所示：

```
extern "C" {
    int mlimitav(int lowend, int highend);
    void updatedesc(char *newdesc);
    double getpct(char *name);
};
```

## C調用C++函數

要使 C 程序能夠調用 C++ 中函數的話，C++ 提供一個符合 C 調用慣例的函數是必須的。下面的例子演示了在 C++ 內創建 C 函數的語法：

```
/* cppsayhello.cpp */
#include <iostream>
extern "C" void cppsayhello(char *str);
void cppsayhello(char *str)
{
    std::cout << str << "\n";
}
```

儘管函數 `cppsayhello()` 通過 `extern "C"` 聲明為 C 函數，事實上它是 C++ 源代碼的一部分，這意味着函數體內是真正的 C++ 代碼。在函數內你可以自由地創建和析構對象。如果你要在 `cppsayhello()` 內調用 C 函數的話，將其聲明為 `extern "C"` 是必須的。否則，編譯器會將作為一個 C++ 函數並相應地更改函數名。

下面是調用 C++ 函數 `cppsayhello()` 的 C 程序：

```
/* c2cpp.c */
int main(int argc, char *argv[])
{
    cppsayhello("Hello from C to C++");
    return(0);
}
```

下面的命令編譯並鏈接生成 `c2cpp`：

```
$ g++ -c cppsayhello.cpp -o cppsayhello.o
$ gcc -c c2cpp.c -o c2cpp.o
$ gcc cppsayhello.o c2cpp.o -lstdc++ -o c2cpp
```

## 其他參考

- GCC新手入門
- C/C++ IDE簡介
- 用GDB調試程序
- Gtk與Qt編譯環境安裝與配置
- 跟我一起寫Makefile
- C編譯初步
- C++編譯初步
- Fortran編譯初步
- C和C++混合編譯初步
- C和Fortran混合編譯初步

取自"[http://wiki.ubuntu.org.cn/index.php?title=Mix\\_C\\_Cpp&variant=zh-hant](http://wiki.ubuntu.org.cn/index.php?title=Mix_C_Cpp&variant=zh-hant)"

本頁面已經被瀏覽5,235次。

- 此頁由Dbzhang800於2008年4月16日（星期三）19:18的最後更改。
  - 關於Ubuntu中文
  - 免責聲明