

月曆下載

文章彙整

Search

4月
18
2008

一個長整數各自表述 (in 64-bit system)

作業系統, 程式語言

Add comments

Size of long integer may be different between 64-bit systems (一個長整數各自表述)

不知道是不是我太落伍了...

我一直以為 C/C++ 下面 short, long, long long 三種資料型態都固定是 2, 4, 8 個 bytes 大小。只有 int 這個資料型態會因為 16-bit/32-bit 系統的不同而變成 2 或 4 bytes 的大小，所以理所當然 int 在 64-bit 的電腦也應該會變成 8 bytes (64-bit) 的大小囉？！

在整理前一篇文章《[Bypass the 2GB file size limit on 32-bit Linux](#)》的時候，讓我驚覺在 64-bit 的系統下，long 的長度也是各自表述的！

首先，int 的大小即使到了 64-bit 的機器上，大部分的系統仍然使用 4 bytes 的大小而已，這主要是為了避免程式從 32-bit 系統轉換到 64-bit 系統需要修改太多地方

再來，請參考 Wikipedia: [64-bit data models](#) 的說明

絕大多數的 UNIX 系統在 64-bit 下面採用 LP64 這種 data model，這時候 long 就不再是固定為 4 bytes 大小，而是變成 8 bytes 的大小了！

然而，Win64 卻不是使用 LP64，而是採用 LLP64 這個 data model，這時候 long 的大小仍然還是 4 bytes

Data Model	short	int	long	long long	pointer
LP64	16	32	64	64	64
LLP64	16	32	32	64	64

Many 64-bit compilers today use the LP64 model (including Solaris, AIX, HP, Linux, Mac OS X, and IBM z/OS native compilers). Microsoft's VC++ compiler uses the LLP64 model.

兩種 data model 的最大差異點就是 long 這個資料型態的大小，LP64 是 64-bit，而 LLP64 則是 32-bit

LLP64 data model 基本上可以說跟 32-bit 的系統一樣，唯一差別只有位址(pointer)改成了 64-bit 而已。資料物件(class, structure) 等如果沒有包含 pointer 的成員的話，整個物件的大小是與 32-bit 系統一樣的！

而 LP64 則是除了位址(pointer)改成 64-bit 之外，long 的大小也變成了 64-bit 大小。所以在 UNIX 下面，要把 32-bit 程式 porting 到 64-bit 可能要比 Windows 多花費多一點功夫。

所以呢，我們觀察到兩個問題影響著程式的相容性

1. 在 UNIX 下面，long 的大小在 32-bit 與 64-bit 的系統下是不一樣的
2. 同樣是 64-bit 系統，UNIX 與 Windows 對於 long 的大小看法是不一致的

為了使程式在 32-bit 與 64-bit 之間以及 UNIX 與 Windows 之間的相容性提昇，改用固定長度的資料型態是寫程式的一個好習慣

在 UNIX 下面，我們可以改用 [stdint.h](#) 這個 header file 中對於資料型態的定義：

```
int8_t      8-bit signed interger
int16_t     16-bit signed interger
int32_t     32-bit signed interger
```

熱門文章與頁面：

- [自製台灣2021年月曆 \(含台灣節日及農曆節氣\)](#) [Word/PDF]
- [月曆下載](#)
- [自製台灣2022年月曆 \(含台灣節日及農曆節氣\)](#) [Word/PDF]
- [自製台灣2020年月曆 \(含台灣節日及農曆節氣\)](#) [Word/PDF]
- [一個長整數各自表述 \(in 64-bit system\)](#)

分類

- 心情小語 (2)
- 台灣 (6)
- 行事曆 (10)
- 作業系統 (15)
- 投資理財 (4)
- 亞洲 (2)
- 美國 (3)
- 消費資訊 (5)
- 旅遊休閒 (11)
- 軟體 (13)
- 程式語言 (8)
- 硬體 (5)
- 資訊安全 (5)
- 電腦資訊 (1)
- 網路科技 (25)
- 趣味軼事 (9)

其他操作

- [登入](#)
- [訂閱網站內容的資訊提供](#)
- [訂閱留言的資訊提供](#)
- [WordPress.org 台灣繁體中文](#)

```
int64_t    64-bit signed integer
uint8_t    8-bit unsigned integer
uint16_t   16-bit unsigned integer
uint32_t   32-bit unsigned integer
uint64_t   64-bit unsigned integer
```

在 Windows 下面，則改用下面這些整數固定大小的資料型態

```
INT8       8-bit signed integer
INT16      16-bit signed integer
INT32      32-bit signed integer
INT64      64-bit signed integer
UINT8      8-bit unsigned integer
UINT16     16-bit unsigned integer
UINT32     32-bit unsigned integer
UINT64     64-bit unsigned integer
```

絕對不要再使用 `int` 和 `long` 了！

尤其是寫網路程式時，很可能 `client` 是 Windows 而 `server` 是 UNIX，然後又有 32-bit 及 64-bit 系統混在裡面，一不小心就發生不相容的問題了...

當然，在 64-bit 的系統下寫程式，要考慮的絕對不只上面這些基本的資料型態。除了 `pointer` 的大小變成 64-bit 外，許多系統內建函式會用到的 `size_t` 及 `off_t` 的大小也變成 64-bit 了.... 寫程式時若有用到這些資料型態，需特別注意，尤其是 `casting` 時，千萬不要用 32-bit 的整數去裝這些資料，免得造成不可預期的結果！

最後提供一個小程式讓你得知你的系統主要資料型態的大小

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 int main()
4 {
5     printf("sizeof(short)    = %d\n", sizeof(short));
6     printf("sizeof(int)      = %d\n", sizeof(int));
7     printf("sizeof(long)     = %d\n", sizeof(long));
8     printf("sizeof(long long) = %d\n", sizeof(long long));
9
10    printf("sizeof(size_t)   = %d\n", sizeof(size_t));
11    printf("sizeof(off_t)    = %d\n", sizeof(off_t));
12    printf("sizeof(void *)    = %d\n", sizeof(void *));
13 }
14 </sys></stdio.h>
```

參考資料:

1. Wikipedia: [64-bit data models](#)
2. [64-Bit Programming Models: Why LP64?](#)
3. [Introduction to Win32/Win64](#)
4. [Porting 32-bit Applications to the Itanium® Architecture](#)
5. [Preparing Code for the IA-64 Architecture \(PDF\)](#)

讚 分享 10 人說這個讚。成為朋友中第一個說讚的人。

相關

[Bypass the 2GB file size limit on 32-bit Linux](#)
2008-04-18
在「作業系統」中

[URL 含中文路徑名稱的終極解法](#)
2005-09-08
在「網路科技」中

[讓 FreeBSD 下的 PHP DBA extension 能支援 'dbm' handler](#)
2005-03-05
在「程式語言」中

Posted by dada at 22:06

7 Responses to “一個長整數各自表述 (in 64-bit system)”

1. 大樂透 says: