

## 程式減肥三步走

對於設計嵌入式 Linux 系統的研發人員來說，有一個問題是必須要考慮到的，那就是記憶體的空间。

我們知道嵌入式 Linux 系統所用的記憶體不是軟碟、硬碟、ZIP 盤、CD-ROM、DVD 這些？所周知的大容量常規記憶體，它使用的是例如 Rom, CompactFlash, M-Systems 的 DiskOnChip, SONY 的 MemoryStick, IBM 的 MicroDrive 等體積極小，與主板上的 BIOS 大小相近，存儲容量很小的記憶體。所以怎樣盡可能的節省空間就顯的很重要。

嵌入式系統的記憶體中放置的無非是內核，文件系統，軟體，以及自己開發的程式。本文就從程式入手，以一個非常簡單的 C 程式來作？例子，通過三步來讓它減肥。

Hello.c:

```
#include <stdio.h>
int main ()
{
    printf ("hello,world");
    return 0;
}
```

我們先用正常的編譯方法來編譯，看看生成的程式的大小是多少

```
#gcc -o hello hello.c
#ls -l hello
-rwxr-xr-x  1 root  root      11542 Nov 13 20:07 hello
```

從結果可以看到正常編譯後的程式大小是 11542Byte

現在開始我們的三步減肥，看看到底效果如何。

### 步驟一：用 gcc 的代碼優化參數

代碼優化指的是編譯器通過分析源代碼，找出其中尚未達到最優的部分，然後對其重新進行組合，目的是改善程式的執行性能。GCC 提供的代碼優化功能非常強大，它通過編譯選項 -O<sub>n</sub> 來控制優化代碼的生成，其中 n 是一個代表優化級別的整數。對於不同版本的 GCC 來講，n 的取值範圍及其對應的優化效果可能並不完全相同，比較典型的範圍是從 0 變化到 2 或 3。

編譯時使用選項 -O 可以告訴 GCC 同時減小代碼的長度和執行時間，其效果等價於 -O1。在這一級別上能夠進行的優化類型雖然取決於目標處理器，但一般都會包括線程跳轉 (Thread Jump) 和延遲退棧 (Deferred Stack Pops) 兩種優化。選項 -O2 告訴 GCC 除了完成所有 -O1 級別的優化之外，同時還要進行一些額外的調整工作，如處理器指令調度等。選項 -O3 則除了完成所有 -O2 級別的優化之外，還包括迴圈展開和

其他一些與處理器特性相關的優化工作。通常來說，數位越大優化的等級越高，同時也意味著程式的運行速度越快。許多 Linux 程式師都喜歡使用 -O2 選項，因為它在優化長度、編譯時間和代碼大小之間，取得了一個比較理想的平衡點。

```
#gcc -O2 -o hello hello.c
```

```
#ls -l hello
```

```
-rwxr-xr-x 1 root root 11534 Nov 13 20:09 hello
```

優化過的程式的大小是 11534Byte, 比正常編譯的結果 11542Byte 似乎沒有小多少，不過不用著急，這才是第一步。我們接著往下進行。

## 步驟二：用 strip 命令

我們知道二進位的程式中包含了大量的符號資訊(symbol table)，有一部分是用來給 gdb 除錯提供必要幫助的。可以通過 readelf -S 查看到這些符號資訊。

```
#readelf -S hello
```

```
Section Headers:
```

[Nr]	Name	Type
[ 0]		NULL
[ 1]	.interp	PROGBITS
[ 2]	.note.ABI-tag	NOTE
[ 3]	.hash	HASH
[ 4]	.dynsym	DYNSYM
[ 5]	.dynstr	STRTAB
[ 6]	.gnu.version	VERSYM
[ 7]	.gnu.version_r	VERNEED
[ 8]	.rel.dyn	REL
[ 9]	.rel.plt	REL
[10]	.init	PROGBITS
[11]	.plt	PROGBITS
[12]	.text	PROGBITS
[13]	.fini	PROGBITS
[14]	.rodata	PROGBITS
[15]	.eh_frame	PROGBITS
[16]	.data	PROGBITS
[17]	.dynamic	DYNAMIC
[18]	.ctors	PROGBITS
[19]	.dtors	PROGBITS
[20]	.jcr	PROGBITS
[21]	.got	PROGBITS

```
[22] .bss                NOBITS
[23] .comment            PROGBITS
[24] .debug_aranges      PROGBITS
[25] .debug_pubnames     PROGBITS
[26] .debug_info         PROGBITS
[27] .debug_abbrev       PROGBITS
[28] .debug_line         PROGBITS
[29] .debug_frame        PROGBITS
[30] .debug_str          PROGBITS
[31] .shstrtab          STRTAB
[32] .symtab             SYMTAB
[33] .strtab            STRTAB
```

類似於 .debug\_xxxx 的就是用來 gdb 除錯的。去掉它們不但不會影響程式的執行還可以減小程式的 size。這裏我們通過 strip 命令拿掉它們。

```
#strip hello
```

```
#ls -l hello
```

```
-rwxr-xr-x  1 root    root          2776 Nov 13 20:11 hello
```

程式立刻變成 2776Byte 了，效果不錯吧。讓我們再接再厲，進行最後一步。

### 步驟三：用 objcopy 命令

上一步的 strip 命令只能拿掉一般 symbol table，有些資訊還是沒拿掉，而這些資訊對於程式的最終執行是沒有什麼影響的。如：.comment; .note.ABI-tag; .gnu.version 就是完全可以去掉的。所以說程式還有簡化的餘地，我們可以使用 objcopy 命令把它們抽取掉。

```
#objcopy -R .comment -R .note.ABI-tag -R .gnu.version hello hello1
```

```
#ls -l hello1
```

```
-rwxr-xr-x  1 root    root          2316 Nov 13 20:23 hello1
```

到這一步，程式的減肥就完成了，我們可以看到程式由正常編譯的 11542Byte 一下子漸少到 2316Byte，效果非常明顯。

### 小結

程式容量的減小無疑對嵌入式 Linux 系統的設計有著重要的意義，它？我們節省了大量空間，使得我們可以利用這部分空間來完善我們的系統，比如加大內核等等，畢竟這才是我們最終的目的。

## 作者簡介

姓名：雷凱

工作單位：升技主板(蘇州)研發中心

聯繫地址：蘇州市新區馬運路羅禮科技有限公司研發中心 郵編 215000

E-mail: tigerleihm@yahoo.com.cn

“ 本文作者是雷凱 升技主板(蘇州)研發中心工程師。他目前在中國蘇州 升技主板(蘇州)研發中心工作。可以通過 tigerleihm@yahoo.com.cn 與他聯繫。 ”