while(1) 和for(;;)有什麼區別?

strongerHuang C語言與C++編程 2021-12-17 09:36

> 作者| strongerHuang
> 微信公眾號| 嵌入式專欄

有讀者問題了類似這樣的問題：**while(1) 和for(;;)**它們不都是無限循環嗎，作用應該一樣啊，它們到底有什麼區別？

要回答這個問題，其實你各自編寫一段while(1) 和for(;;)的代碼，編譯對比一下代碼大小和彙編文件，你就大概知道了。

while(1)和for(;;)語法表達

這裡先說一下while(1)和for(;;)語法表達式。

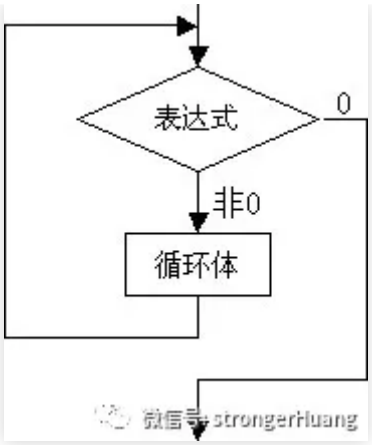
1.while語法表達

```
while( 表达式 )
{
  语句
}
```

其中：

- 表達式：是循環條件
- 語句：為循環體。

while語句的語義是：計算表達式的值，當值為真(非0)時， 執行循環體語句。其執行過程可用下圖表示：



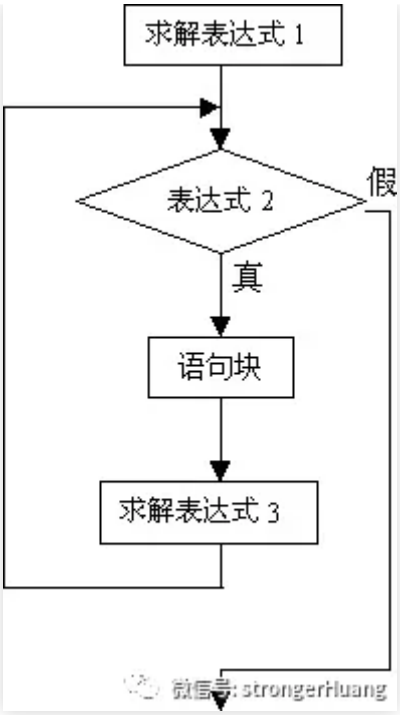
2.for語法表達

```
for(表达式1; 表达式2; 表达式3)
{
   语句
}
```

它的執行過程如下：

> 1.先求解表達式1
>
> 2.求解表達式2
> 若其值為真（非0），則執行for語句中指定的內嵌語句，然後執行下面第3）步；
> 若其值為假（0），則結束循環，轉到第5）步。
>
> 3.求解表達式3
>
> 4.轉回上面第2）步繼續執行。
>
> 5.循環結束，執行for語句下面的一個語句。

執行過程可用下圖表示：



while(1)和for(;;)異同點

這裡先說一下結論，然後再驗證驗證結論。

1.相同點

作用和效果都一樣：都是實現無限循環的功能。

2.不同點

while(1)：其中括號裡面是一個條件，程序會判斷真假。而括號裡面的“1”永遠是一個“真值”。

其中，每一次循環，編譯器都要判斷常量1是不是等於零。

for(;;)：這兩個;;空語句，編譯器一般會優化掉的，直接進入死循環。

根據上面的描述，你可能會覺得：while(1) 比for(;;) 要做更多事，彙編代碼更多，代碼量也更大。

但事實是這樣嗎？下面驗證一下。

驗證while(1)和for(;;)差異

我們編寫分別兩個文件for.c和while.c，然後分別生成彙編代碼，看下情況。

1.源代碼

while.c:

```
// filename: while.c
int main(int argc, char const *argv[])
{
    while(1)
    {}

    return 0;
}
```

for.c:

```
// filename: for.c
int main(int argc, char const *argv[])
{
    for(;;)
    {}

    return 0;
}
```

2.生成彙編

我們這裡使用gcc編譯器生成彙編，執行命令如下：

```
gcc -S -o while.s while.c
gcc -S -o for.s for.c
```

while彙編代碼：

```
; filename: whiles
  .file  "while.c"
  .text
  .globl  main
  .type  main, @function
main:
.LFB0:
  .cfi_startproc
  pushq  %rbp
  .cfi_def_cfa_offset 16
  .cfi_offset 6, -16
  movq  %rsp, %rbp
  .cfi_def_cfa_register 6
  movl  %edi, -4(%rbp)
  movq  %rsi, -16(%rbp)
.L2:
  jmp  .L2
  .cfi_endproc
.LFE0:
  .size  main, .-main
  .ident  "GCC: (GNU) 9.3.0"
  .section  .note.GNU-stack,"",@progbits
```

for彙編代碼：

```
; filename: for.s
  .file  "for.c"
  .text
  .globl  main
  .type  main, @function
main:
.LFB0:
  .cfi_startproc
  pushq  %rbp
```

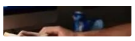
```
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq  %rsp, %rbp
    .cfi_def_cfa_register 6
    movl  %edi, -4(%rbp)
    movq  %rsi, -16(%rbp)
.L2:
    jmp  .L2
    .cfi_endproc
.LFE0:
    .size  main, .-main
    .ident  "GCC: (GNU) 9.3.0"
    .section  .note.GNU-stack,"",@progbits
```

你會發現，除了文件名不同，其餘都相同。

當然，這裡額外說一下，不同代碼、不同編譯器，以及不同優化等級，可能最終結果有所差異。

--- EOF ---

C語言與C++編程



生活| 口罩有致癌物？戴前抖一抖？真相來了

法製文萃報

外賣員在顧客麻辣燙中小便，細思恐極

喬志峰評論