自由研究專題 I

經濟二 407510046 黃郁雯

問題一

這支程式要如何轉成組合語言？轉成組合語言會是長什麼樣子？

在編譯時加入`-S`參數(`gcc -S test.c`)就可以轉成組合語言。

以下是這支的程式轉成組合語言的結果:

```
    .file   "test.c"
    .text
    .section    .rodata
.LC0:
    .string "%d"
.LC1:
    .string "The a variable us %d.\n"
    .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
    subq    $16, %rsp
    movq    %fs:40, %rax
    movq    %rax, -8(%rbp)
    xorl    %eax, %eax
    movl    $5, -16(%rbp)
    movl    $6, -12(%rbp)
    leaq    -16(%rbp), %rax
    movq    %rax, %rsi
    leaq    .LC0(%rip), %rdi
    movl    $0, %eax
    call    __isoc99_scanf@PLT
    movl    -16(%rbp), %eax
    movl    %eax,%esi
    leaq    .LC1(%rip), %rdi
    movl    $0, %eax
    call    printf@PLT
    movl    $0, %eax
    movq    -8(%rbp), %rdx
    xorq    %fs:40, %rdx
    je  .L3
    call    __stack_chk_fail@PLT
.L3:
    leave
```

```
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
.LFE0:
    .size   main, .-main
    .ident  "GCC: (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0"
    .section    .note.GNU-stack,"",@progbits
```

問題二

在 gcc -O0下，每一個statement個別會對應到哪一個區塊的組合語言？你能知道每一條組合語言的意義嗎？

`int a=5;`	`int b=6;`	`scanf("%d", &a);`	`printf("The a variable is %d.\n", a);`	`return 0;`
`subq $16, %rsp`	`movl $5, -16(%rbp)`	`leaq .LC0(%rip), %rdi`	`leaq .LC1(%rip), %rdi`	`je .L3`
`movq %fs:40, %rax`	`movl $6, -12(%rbp)`	`movl $0, %eax`	`movl $0, %eax`	`call stackchk_fail@PLT`
`movq %rax, -8(%rbp)`	`leaq -16(%rbp), %rax`	`call isoc99scanf@PLT`	`call printf@PLT`	
`xorl %eax, %eax`	`movq %rax, %rsi`	`movl -16(%rbp), %eax`	`movl $0, %eax`	
		`movl %eax,%esi`	`movq -8(%rbp), %rdx`	
			`xorq %fs:40, %rdx`	

`subq $16, %rsp`	`pushq %rbp`	`xorl %esx, %esx`	`movq %rsp, %rbp`	`movl %6, -12(%rbp)`	`xorq %rdx,%rdx`	`je`
保留 16 byte 給框架區	儲存上一層函數的框架暫存器	把 esx 設為 0	儲存返回點	6=-12(%rbp)	對%rdx置0	等於則跳轉

問題三

你認為每一台電腦所編譯出來的組合語言會一樣嗎？什麼情況下一樣/不一樣？

不一定，因為組合語言主要是要看中央處理器(CPU)的指令集(x86,ARM,...)而不同，所以可能會一樣也可能會不一樣。

問題四

如果使用 gcc -O0、 gcc -O1、 gcc -O2以及 gcc -O3所編譯出來的程式在組合語言的呈現上會有什麼樣的差異？為何會有這些差異？以這隻程式來說，效能會有差異嗎？

gcc-O0 不要優化。

gcc-O1優化。優化編譯需要花費更多的時間，並且需要更多的記憶體去優化較大的函式。主要對代碼的分支，常量以及表達式等進行優化

gcc-O2進一步優化。GCC執行幾乎支持所有的優化，而這些優化不涉及空間速度。指定時，編譯器不執行循環展開或函數內聯gcc-02。相比於gcc-O0，此選項會增加編譯時間和所生成代碼的性能，但會在編譯期間佔用更多的內存和編譯時間。

gcc-O3優化更多。 gcc-O3 打開由指定的所有優化 gcc-02 並打開內聯函數。例如使用偽寄存器網絡，普通函數的內聯，以及針對循環的更多優化。

根據的情況程式速度會有所區別，但是越好的的優化中間所使用的方法會比較複雜，不小心就會有bug，並且編譯時也會花比較久的時間。

不會。

問題五

請你調查看看：如果拿執行檔，你能使用一些工具還原成組合語言嗎？或者直接還原原本的C程式碼？

Showman、Salamander 、Reshape、net relector、ilspy、dot peek、JustDecompile

比較

除了 gcc以外，還有 clang、 MSVC等等的編譯器，如果你心有餘力可以做其他編譯器的輸出比較。

gcc的特性：

1. 可處理C、C++、Fortran、Pascal、Objective-C、Java、Ada，以及Go等語言。
2. gcc較為流行且廣泛使用。

clang特性：

1. 速度快，為gcc的2.5倍速。
2. 內存占用小。
3. 診斷信息可讀性強。

MSVC的特性：

1. 可處理C/C++、C#、Visual Basic、F#、Python等程式語言。
2. 整合微軟Windows視窗作業系統應用程式介面（Windows API）、三維動畫DirectX API、Microsoft .NET框架。

資料來源

https://gcc.gnu.org/onlinedocs/gcc-2.95.3/gcc_2.html#

https://blog.csdn.net/qq_31108501/article/details/51842166

https://www.itread01.com/content/1548093978.html

https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/547323/

http://www.ruanyifeng.com/blog/2018/01/assembly-language-primer.html

https://medium.com/%E7%A8%8B%E5%BC%8F%E4%BA%BA%E6%9C%88%E5%88%8A/c%E8%AA%9E%E8%A8%80%E5%B0%8D%E6%87%89%E7%9A%84%E7%B5%84%E5%90%88%E8%AA%9E%E8%A8%80%E8%A7%A3%E6%9E%90-bf942e6562b4

https://medium.com/@jefflin1982/showman-%E5%8F%8D%E7%B5%84%E8%AD%AFsharedlib%E7%9A%84opensource%E5%B7%A5%E5%85%B7-dba4e32c8fed

https://fzheng.me/2016/03/15/clang-gcc/

https://dotblogs.com.tw/kinanson/2017/04/14/105631

https://www.itread01.com/content/1497195488.html