**專家心得報告之二**

106級甲班 F74024070 蔡侑軒

今日講師：吳中如 老師

主題：學術Compiler VS 業界Compiler

上課時間：105年5月13日下午兩點到五點

簡介：

* 2015年清大資工博士班畢業
* 前魔獸專業玩家

經歷：

* 碩博士班邊讀書邊到公司上班
* Compiler / Toolcain Tem Leader
  + Andes Corp.
  + 改GCC (open source)
* Skymizer Inc. Tech Lead (現職)
  + Skymizer Inc.
* GCC NDS32 CPU Port Maintainer
  + GCC community

課程內容：

* GCC 是 compiler driver
  + 程式 => cpp (把.h檔轉成.c檔  純c code 變 .i .ii 檔)
  + cc1(產生assembly code  .s檔)
  + as(把.s檔 轉成 機械碼 .o檔)
  + collect2(link.....)
* 實作compile程式不透過GCC!
  + 注意產生出來的.out .o ....檔案是不是當下生產出來的，還是之前所留下的
  + 用 -v 可以看GCC 的完整訊息
  + GCC新版後 cpp 和 cc1已結合，若要看cpp可下 –E
  + 真正的核心compiler：cc1
  + -S 可以產生 .s 檔
  + 最佳化參數:
    - -Olevel，( 0 1 2 3 s g fast )(低到高)(套餐)
      * -O0 by default (不做任何事)
      * -O1 多做一點
      * -O2 dead code，inline
      * -O3 做很多，但是因為有loop unrollling 等等，code size 有可能會很大 不是必然最好
      * -Os optimization for size 以size為考量
      * -g 好debug?
      * -Ofast  加上一些數學函數 不考慮float運算的一些特殊例子 +-0 +-無限大....)
    - -f option 可以做單獨的某種optimization　(單點)
      * 通則：-foption　-fno-option　-foption = n
      * -fbuildin -fno-buildin
      * -ftree-vectorize：運算結果如果是常數便直接用常數代替-fno-tree-vectorize

※printf 複雜，在沒有%d 等等東西時，compiler會用puts(較簡單)來取代！

※compiler 的 optimization 彼此之間沒有 "正交性"(會互相影響)，所以結果有時候不一定會如預期！

* + - -W　warning message => 盡量要解掉！
      * 在C語言的標準規格，sign overflow is undefined behavior (無法預測結果!!)
      * 在function沒有太多行的時候，inline optimization 會把function的內容直接放到caller(main)裡面
    - -D　define　-DLOOP = 2　-DLOOP=0　-DLOOP　(value = 1 by default) 　macro!!
      * [#ifdef](https://hackpad.com/ep/search/?q=%23ifdef&via=cqhTTZORzrE) LOOP，只要有define LOOP 就會執行
      * [#if](https://hackpad.com/ep/search/?q=%23if&via=cqhTTZORzrE) LOOP，要define LOOP 且 value不為0 才執行
* Language Standard
  + 業界很重要!!!!!因為要面對很多人
  + C89(NCC)：語言標準第一版!!
  + C90：ISO 出的，幾乎抄C89!?
    - C89/90 規定variable declaration 一定要在block的最開頭!! => 所以不能用for(int i ...)這種
      * 到C99才行，gcc inti.c -std=c99
    - 在code 內加上"inline"：建議compiler把此function 用inline (C90)
    - "inline" inline definition：當有不同檔案之間的function call時，可以用來避免不會重複定義某function，又可以實行inline!! (C99之後)

※在業界不好用printf!!會有很多很多問題

* 阻擋最佳化
  + printf只能在O0做debug
  + 因為printf是function code
* 有最佳化要用GDB來debug
* Regression Test
  + 當成是加上新東西後，除了新的test code以外，還要測試舊的test code
  + Benchmarks：業界愛用test code

※GCC 4.9 快過時

※GCC 5 6 有在維護

課程心得：

這次的演講聽得非常開心！首先是對吳中如博士的台風。他的台風我非常的喜歡！既沒有到jserv那般的浮誇，也沒有到Luba那樣閉月羞花，覺得他正經與玩笑的部分拿捏得可是恰到好處！唯一讓人不開心的可能是他沒有誠實地說出自己的經歷吧！要不是教授適時出面爆料出實際情況，可能現在大家都還被騙得團團轉，覺得打魔獸打打真的就能成為超級名人呢！

另一個我很喜歡的是他演講的方式！與前面兩位jserv和Luba有著極大的不同，博士是用許許多多的實作來代替PPT與許多的唇舌！雖然這樣的方式使他在一開始來到這個陌生(詭譎)的成大製作系時，遇到了許多不可預期的困難，但從台下的我看起來，他從容不迫、循序漸進、有條有理的披荊斬棘，直到環境架完為止，這個過程我看的津津有味！

再來是他演講的內容，前面Jserv提到了許許多多的Compiler

經過前三次的專家演講，各個都讓我刷新五感，不由得讓我期待起接下來的每一場演講…的我懷抱著這種心情來上這次的課，但這次我真的有點小失望，也許是前面兩位的演講太精彩吧？所以對於這次的我比較雞蛋裡挑骨頭一點……

首先是台風。前面兩位講者，都展現出不錯的台風，Luba可能看起來還稍稍有點緊張，Jserv則是非常自在有趣的講述他備的內容，與他們相比，這次的講者講話語氣就如同水平線一樣平，讓我在冷氣悠悠的教室裡，不禁哈欠直打，周公敲門，這是前面三次演講都不曾給我的感覺……而其中令我印象最深的，便是在演講的最後。講者說的甚是有理，「開天窗」（晚到、晚退）的確是演講或是授課的大忌，不論是對台下聽眾的心理感受還是對台上講者的演講專業。而在那時，隔壁的教室提早下課，學生們蜂擁而出，伴隨著腳步與嘻鬧聲傳至整間教室，回到許久，瞬間場面變的一陣尷尬：台下聽眾一方面聽到嬉鬧聲便開始心猿意馬，一方面因為吵雜聲而聽不見講者的聲音；而講者可能是深怕「開天窗」，便不顧一切的快速帶過最後的幾頁PPT……即便講者真的沒「開天窗」好了，但最後的幾段台下的人根本無心也無法聽得進去，面對這樣的情況，講者的選擇竟是視而不見，令我傻眼……反觀之前有一次我們上Compiler，教授難得晚下課，這時趕著上班的我原本想要默默地先收東西，卻不小心發出巨大聲響，令場面一陣尷尬（這是我不對……），但面對地與這次類似狀況的教授，選擇了正面解決這樣的窘境：「我看到同學有人在收書包了…我們在一下就好…」之類的，化解了尷尬，也得以使得課程延續！於此，身為上大學後也算是身經數戰的報告者，台風包含的不僅是要把你所準備的報告內容完好呈現，另一個重大部分也是面對任何情況（如這次情況，或是回答聽眾問題）都能迎刃而解的危機處理能力啊……

但回頭想另一件事，忘記在哪時聽到還看到Jserv說，不知何時，老師的職責，不再只是傳授知識這麼簡單，還要考慮學生的視覺、聽覺感受，覺得老師的目的似乎偏離了原先的初衷……等等，這令我深思，再看這次的情況，也許不能都focus在台風上，但演講的內容也讓我很不開心……或許是該說講者真的很崇拜Jserv吧！？他的演講內容，跟Jsev的演講內容有八成像，甚至還出現和Jserv當初拿來開頭的「生活中的compiler」，只是它放在最後當結尾而已……且聽過就算了，Jserv與他講的內容相比只能說是有過之而無不及，許多當初Jserv提到的重點，他都帶過甚至是沒提……例如原先optimization的困難點，他講了幾個，但關於最重要的，pointer aliasing，他可說是隻字未提，而其他的部分，我覺得Jserv講的都比他來的清楚好懂得多……

於是，我跟我朋友甚至在聊，為什麼Jserv不像上次Luba那樣來台下聽演講呢？會不會是看不起他甚麼的……但後來不僅聽到教授說他晚上跟Jserv有約，而且晚上看到Jserv Po文FB，上頭寫著與講者相談甚歡，甚至Jserv還自認講者的能力在自己之上甚麼的……對我而言算是一大打擊！也許只能說，好的學者不一定就會是好的講者…這樣吧？