**專家心得報告之三**

106級甲班 F74024070 蔡侑軒

今日講師：Logan (簡子翔) 老師

上課時間：105年5月6日下午兩點到五點

簡介：

* Master of Science in Computer Science / National Taiwan Unversity / 2012 — 2013
* Bachelor of Science in Computer Science / National Taiwan Unversity / 2008 — 2012

經歷：

* MediaTek / Software Engineer / 2013 — present
* Google / Software Engineer Internship / 2011 — 2013
* LLVM & Clang
* Android Open Source Project
  + android NDK開發者

※PPT網址: <http://slide.logan.tw/compiler-intro/#/>

課程內容：

* Compiler => 程式and最佳化
* Lexer
  + 把詞跟詞分開
  + 要區分關鍵字跟識別字
  + 貪多識別方法：字看最多，不會隨意斷句
* Parser(LL | LR)
  + 文法的分析，建構出程式結構
  + LL parser在業界比較常用，因為產生錯誤訊息方便，貼近人的思維
* syntax-directed translation
  + 看到甚麼敘述就產生甚麼code

※單純的能翻譯與最佳化有很大的一段差距!!

* compiler requirement
  + 結果要正確
  + 設備使用率高
* three related fieds
  + Programming language theory
  + Computer Architecture
  + Compiler
* programming language theory
  + type theory
    - variable scoping
    - language semantics(語意)
    - 額外資料<dependen time>
  + 人要好讀懂code
  + 抽象化的程式語言!?
* 問題一
  + (a) **covariant type**, (b) **mutable records**, and (c) **type consistency**. 只能擇二
    - pointer 指向const時 有可能會使得const的值被更改
* 問題二
  + dynamic scoping：程式的結果決定於它被call的位子(非常不好)
* Computer architecture
  + C語言只保證程式結果與從上往下讀的結果一樣 並非真的從上往下
    - 最佳化在平行化處裡時會出問題
  + Memory hierarchy: load & store 特別慢問題
* Compiler optimization
  + Scalar optimization
  + Vectorize optimization
  + Interprocedural optimization
* just-in-tiime-compiler 即時翻譯語言
* binaty translators 把A語言翻譯成B語言 (手機android => x86)
* 效率測試
* 自動GB
* 業界的compiler VS 學校的compiler
  + 業界compiler幾百萬行起跳!
    - 程式語言的行為分析
    - 發展最佳化(要證明)
    - 要有中間碼
    - 要夠高階(要看得遠)但又要夠低接(複雜)
    - 收前端不同的程式語言 產生後面不同的機器碼 (分多段!!)
* GGC
  + Generic：程式語言的結合體
  + Gimple：類似assembly
  + Rtl：超低階，register......
  + Basic block
    - 一定要從頭進去，一定是最後一個(尾)出來
    - 構成control flow graph
  + control flow graph
    - definition：where value is assigned to variable
    - use：where the variable is used
    - reaching definition：where the variable value is  from
    - dominance relation：一定會經過誰！到某一點之前一定會經過誰 dominance tree
    - dominance frontier: A往下走到第一個不被A dominate的B
      * B是兩條路的交界
* static single-assignment form
  + static：A static analysis to the program (not the execution.)
  + single-assignment：每一個variable只被assign一次
  + SSA是目前最常用的compiler寫法
  + 好處：每一個variable只被assign一次(好找!)
  + Referential transparency：r => r.0
  + phi function：回傳值決定於從哪裡進入phi function
    - phi function會放在dominance frontier
* Optimization
  + constant propagation / constant folding：把 (SSA超好做)
  + global value numbering：用hash table做，要traverse dominator tree ，如果動作有做過=>就直接給值；沒有就把動作跟值存入hash table
  + dead code elimination：永遠到不了的code
  + loop optimization
    - loop invariant：不變的拿出來
    - induction variable
    - loop strength reduction：減少loop的計算
    - loop unrolling：擴大loop 減少load store的stall (software pipelining.)
    - instruction selection：就是複合的指令
      * 盡可能挑出最好的指令!
* Register allocation
  + allocation：把虛擬的registers縮成有限的k個registers
  + assignment：把值給register registe，任何時候都要足夠
* Instruction scheduling
  + Data dependence
  + Functional units
  + Issue width
  + Datapath forwarding
  + Instruction cycle time

課程心得：

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？