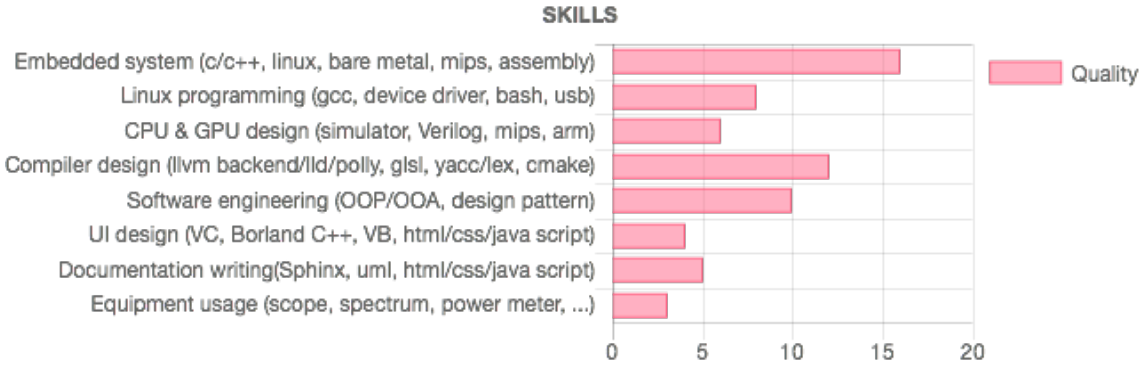
陳鍾樞

我是位有經驗的軟體工程師，過去在工作中使用不同軟體語言、工具，來開發軟硬體產品，諸如：數位電視機上盒、處理器，編譯器，模擬軟體 simulator，...，。 對寫程式總是懷抱熱情，瞭解工具與軟體工程方法，喜愛思考程式與除錯，並對撰寫文件以產生較有可讀性及可維護性的軟體感到快樂。

詳細履歷

資格

二十年c/c++嵌入系統軟體開發經驗，五年編譯器相關工具開發經驗，碩士時研究平行處理。



我的開放原始碼專案

很高興我的作品已被LLVM接受，出現在 http://llvm.org/docs/tutorial/#external-tutorials

如何建立LLVM後端編譯器 http://jonathan2251.github.io/lbd/index.html

如何建立LLVM後端系統工具 http://jonathan2251.github.io/lbt/index.html

學歷

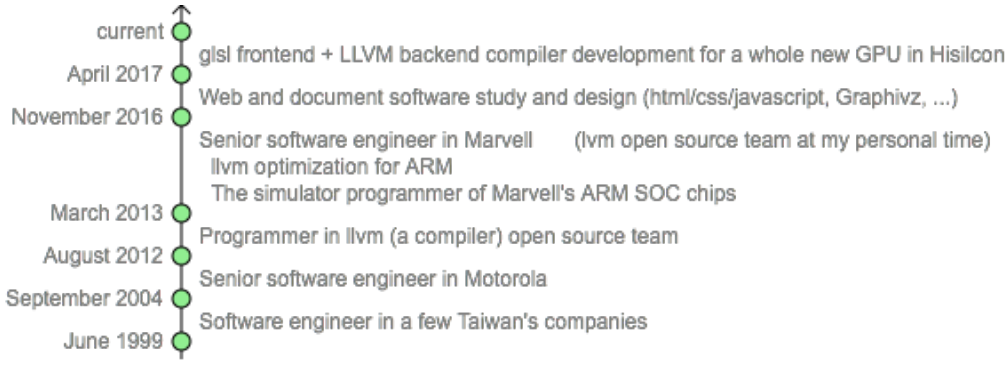
1997-1999 碩士班，六月 1999，國立台灣師範大學，台北，主修：資訊科學。

1991-1994 學士班，六月 1994，國立台灣科技大學，台北，主修：工業工程。

證照

1995年 高考資訊技師及格。

經驗



碩士論文THESIS OF MASTER DEGREE

The Researches of Column Sort and Related Problems

博士班研究計畫

The Researches of Sorting Network and Related Algorithm

其餘作品

修影像處理課程與撰寫:Jpeg decoder程式

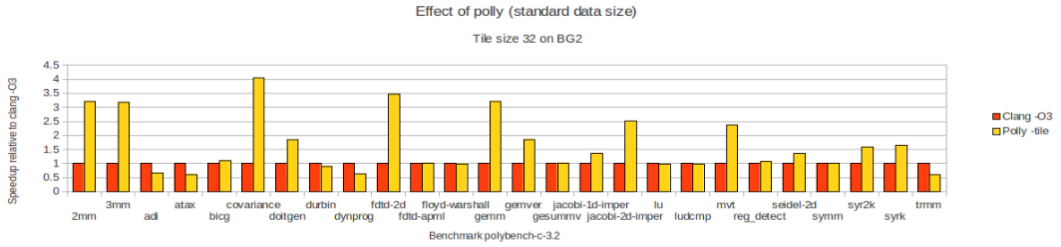
網頁與javascript: html簡歷 與 我個人網頁

Graphivz: 如此詳細履歷裡的一些圖學的圖。原始碼： mywork_1.gv and study_and_apply_ch1.gv

工作貢獻

Marvell

- 設計半自動的軟體系統，自動執行用gcc編譯器編譯benchmark程式，並產生excel比較圖表。
  - 目的是確保新改的gcc沒有邊際效應(side effect) (解釋如此份流程圖)。
  - 設計bash scrip比較任兩版的gcc編譯器的benchmark效能。
- 為提升Marvell公司gcc與llvm編譯器軟體工具效能，介紹Polly軟體系統。Polly是針對loop最佳化的開放原始碼專案。
  - 此系統的好處是針對某些數值應用程式，像是矩陣相乘、各類矩陣運算的程式，他能提升五倍的速度，遺憾的是Marvell的處理器並沒有數值運算領域的運用，但也許未來的64位元的處理器會有機會。

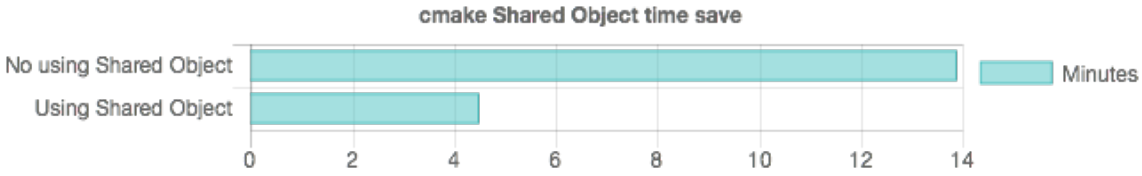


- 提出解決方案利用編譯器產生程式碼，此想法獲得Csim小組採用。 推薦函

  ARM spec → script → generate c++ .h .cpp

  將滑鼠移到上面紅字上 (in html) 可得到解釋，明顯的script比c++程式來得簡潔、符合ARM spec、且容易閱讀。

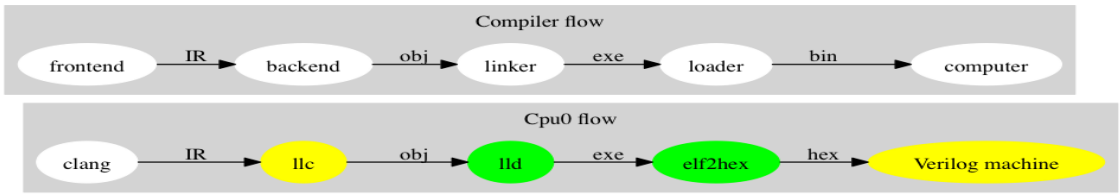
- 完成cmake以及調整python程式，使Csim軟體系統可採用一套cmake而不必維護make與windows專案檔兩套專案檔，同時也支援蘋果電腦的使用者。 推薦函
  - 介紹可改進cmake的程式寫法。
    - 安裝於公司伺服器的cmake版本是2.6.x，儘管如此，cmake 2.8.8提供shared object支援以節省不必要的編譯時間。



  - cmake與ninja兩種編譯時效評估報告。

我的llvm開放原始碼專案

- 下圖的上半部是計算機產生與執程式的流程圖。IR是中間碼的縮寫（Intermediate Representation）。下半部是llvm的流程圖。黃色與綠色分別是我書中（如上，我的開法原始碼專案），如何建立LLVM後端編譯器與如何建立LLVM後端系統工具中的作品。

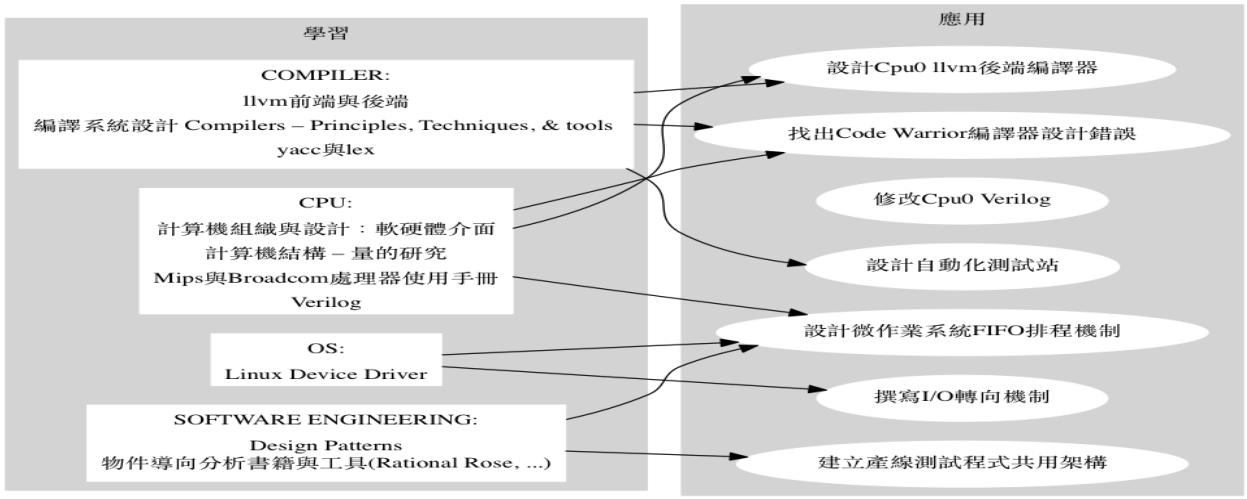


Mortorola

- 設計QIP7kP1與P2電路板的產線測試程式。超過五百萬台的此一數位機上盒機型採用此程式進行測試並出貨。其他主要產品DCX33、DCX34與DCX35(共超過一千萬台)都是從此程式移植修改而來的。
- 依主管要求，運用mips處理器與軟體工程知識，帶領、並與其他四位成員一起建立產線測試程式的共用架構。

。

計算機、編譯器、作業系統與軟體工程相關經驗



計算機結構學習與應用：

- 研讀書籍“計算機組織與設計：軟硬體介面”，“計算機結構 – 量的研究”與Mips與Broadcom處理器使用手冊。還有Verilog語言與工具研究。
- 工作應用：
  - 找出Code Warrior編譯器支援Mips處理器中的錯誤並告知Code Warrior。此錯誤來自preprocess #pack(1)的程式碼生成。
  - 使用mips組合語言製作"微作業系統"機制，以解決程式掛住、無法運行的問題。
    - 此後系統變得更加穩定且快速。當一測試項目死當時，其他測試項目皆無法繼續執行，此情況是一嚴重問題，因為測試結果是在全部測試項目完成後，才能顯示， 遇此狀況，作業員無法得知測試結果。我製作一個time out機制，讓死當的測試項目得以被略過(藉由從預存的堆疊指標與暫存器還原其值)，美國同事稱它為微作業系統， 使用c++加上幾百行的mips組合語言的"observe pattern")。
  - 運用Verilog與書籍“計算機結構 – 量的研究”所學，修改Cpu0 Verilog程式碼使其更符合llvm後端編譯器支援的實際處理器設計。

編譯器學習與應用：

- 研究llvm前端與後端設計，研讀書籍“編譯系統設計 Compilers – Principles, Techniques, & tools 2nd Aho,...”, yacc與lex工具。
- 工作應用：
  - 為教學與自我學習，設計Cpu0 llvm後端編譯器（列在此履歷前面“我的開法原始碼專案”）。
  - 運用編譯器語法分析工具yacc與lex，設計自動化測試站。

作業系統學習與應用：

- 研讀書籍“Linux驅動程式 Linux Device Driver (2nd & 3rd edition)”
- 工作應用：
  - 如上，“計算機架構學習與應用”所列，我參考作業系統並使用mips組合語言來設計"微作業系統"機制，以解決程式死當無法運行的問題。
  - 在Motorola時，撰寫I/O轉向機制，讓新版的程式更容易移植。

軟體工程學習與應用：

- 研讀書籍“Design Patterns”與"物件導向分析書籍與工具(Rational Rose, ...)"
- 工作應用：
  - 在Motorola時，依主管要求，與其他四位成員建立產線測試程式共用架構。
    - 使用Rational Ross工具設計此架構。我應用物件導向分析、設計以及UML工具，從事此任務。
    - 使用書籍“Design Patterns”中的五種patterns以及Rational Ross工具建立程式共用架構、文件，並產生程式碼。

推薦函

[前主管推薦函: https://jonathan2251.github.io/ws/ch1/RL_Marvell.pdf](https://jonathan2251.github.io/ws/ch1/RL_Marvell.pdf)