1. RISC-V

RISC-V 開放指令集處理器，是建立在 RISC 原則上的 ISA ，為[開源軟體](https://zh.wikipedia.org/wiki/%E9%96%8B%E6%BA%90%E8%BB%9F%E9%AB%94)運動相對應的一種開源硬體。RISC 精簡指令集，為電腦中央處理器的一種設計模式。ISA 指令集架構，是與電腦程式設計相關的一部分。RISC-V內含有32個整數暫存器，如果加上浮點數運算，則在多32個浮點暫存器。這些暫存器又可分為堆疊站存器、臨時變數暫存器、參數暫存器、還有一個暫存器沒有被列入，那就是PC全名為Program Counter ，RSIC-V的PC是一個獨立的暫存器，一般指令無法存取，因此想要讀取PC必須使用auipc 這樣特殊的指令。RISC-V 還定義一群控制暫存器 CSR (Control Status Registers)，CSR暫存器的保留上限為4096個，而CSR被用於設定或是紀錄處理器的運作狀況。

RISC-V指令集的設計考慮了體積小、速度快、低功耗的情況來實做。RISC-V的指令集可以分為很多群，例如 : [ RV32I：基础整数指令集 (固定不變了)、RV32M：乘法和除法、RV32F：單精度浮点操作（和 RV32D:雙精度浮點操作）、RV32A：原子操作、RV32C：可选的压缩扩展 (對應 32 位元的 RV32G)、RV32B：基本擴展。、RV32V：向量扩展（SIMD）指令、RV64G：RISC-V 的 64 位地址版本。]

處理器是藉由接收指令進行運作的，若處理器在執行指令流的時候遇到無法預期的情況，便稱之為異常，而異常是發生在硬體、程序上的故障。異常可分為兩種，同步異常、異部異常。同步異常 在同樣的程序以及環境執行 n 次都能將同樣的異常狀態重現，該異常就可以被歸類到同步異常。而異部異常又能在被細分為兩種狀態: 精確的異步異常(外部中斷)以及不精確的異步異常(讀寫內存時出現錯誤)。現今的作業系統都具備多工處理的能力，處理器都設計有中斷的能力，讓作業系統能指派不同的程式給處理器運行，達成所謂的多工。在 RISC-V ISA 的定義中，中斷分成四種，分別是:External Interrupt、 Timer Interrupt、 Software Interrupt、 Dubug Interrupt。

External Interrupt指的是因為處理器外部造成的中斷、Timer Interrupt是計時器在比較值得時後造成的中斷、Software Interruptup 是軟體觸發而造成中斷、Dubug Interrupt該中斷用於 Debugger 的實作。

2. c 語言

C 語言是 Dennis Richie 與 UNIX 發明者 Ken Thompson 合作一起建構出來的一種程式語言，C語言的優點和缺點是相互綁再一起，以下幾種優缺點，速度很快。沒有垃圾蒐集機制 ，分配記憶體得自行回收優點是很快但缺點是麻煩而且容易出錯。有指標因此可以用來儲存記憶體位址，可透過指標存取任何記憶體。其優點是低階存取但缺點是指標會亂指。編譯式優點是執行快，缺點是需要花時間編譯。因此C語言適合拿來寫底層運作，而不是高層應用。

3. 心得

看完RISC-V的簡介， RISC-V有著面積小，速度快，低功耗的優點。並且可以用來設計各種不同用途，可允許任何人製造、生產、設計。能夠讓大眾去學習，在閱讀這本書的過程，我感受到這本書的內容對於我還是有點難以理解，雖然可能對於其他人來說很簡單，但我需要一直上網尋找一些解釋，希望之後我能多學習到這方面的知識，讓我自己能清楚接收到這本書的知識且完整了解RISC-V。

看完Ｃ語言的簡介，Ｃ語言因為其靈活以及快速的特性大部分只運用在底層運作，但為什麼沒辦法運用在高層上，其原因是因為它對於錯誤處理較差，當app那些較為複雜的系統，語法容易因為繁瑣而出錯，但我想Ｃ語言重要的原因並不只是因為快速，它的設計使它成為25年來最廣泛的程式語言，包括我們目前使用的Microsoft Windows 以及macOS。看完了書中的簡介，我知道c語言並不只是一個程式語言，它也代表著這些開發程式軟體的基石。