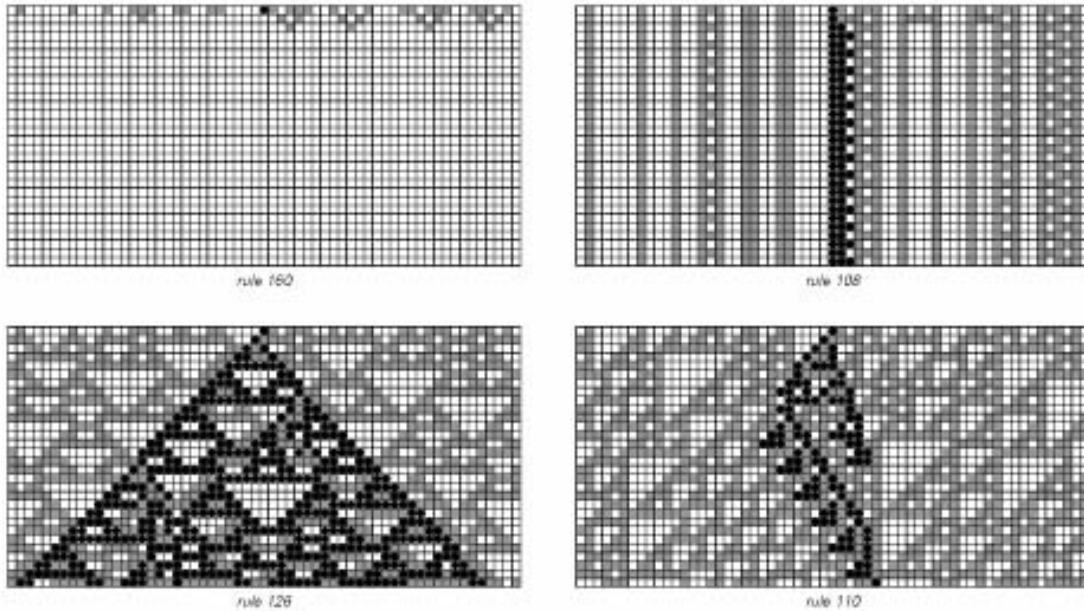


Computer Organization and Assembly Languages Final Project

細胞自動機



一維細胞自動機 ↑

細胞自動機

細胞自動機，又稱格狀自動機、元胞自動機，是一種離散模型，在可算性理論、數學及理論生物學都有相關研究。它是由無限個有規律、堅硬的方格組成，每格均處於一種有限種態。整個格網可以是任何有限維的。時也是離散的。每格於 t 時的態由 $t-1$ 時的一集有限格（這集叫那格的鄰居）的態決定。每一格的「鄰居」都是已被固定的。（一格可以是自己的鄰居。）每次演進時，每格均遵從同一規矩一齊演進。去年的雙班計算機程式其中考中，就考了這麼一題，要求受測者寫出一個 C 程式，展示一個 25×25 的陣型，其中的每一格有「無」和「有」兩種狀態，隨機產生初始陣型，之後照已下規則計算：任何一格「有」的周圍（共計 8 格），如果「有」少於兩格或多於三格，則「有」死亡，否則維持原狀。如果「無」的周圍恰有三格「有」，則「有」出生，否則維持原狀。

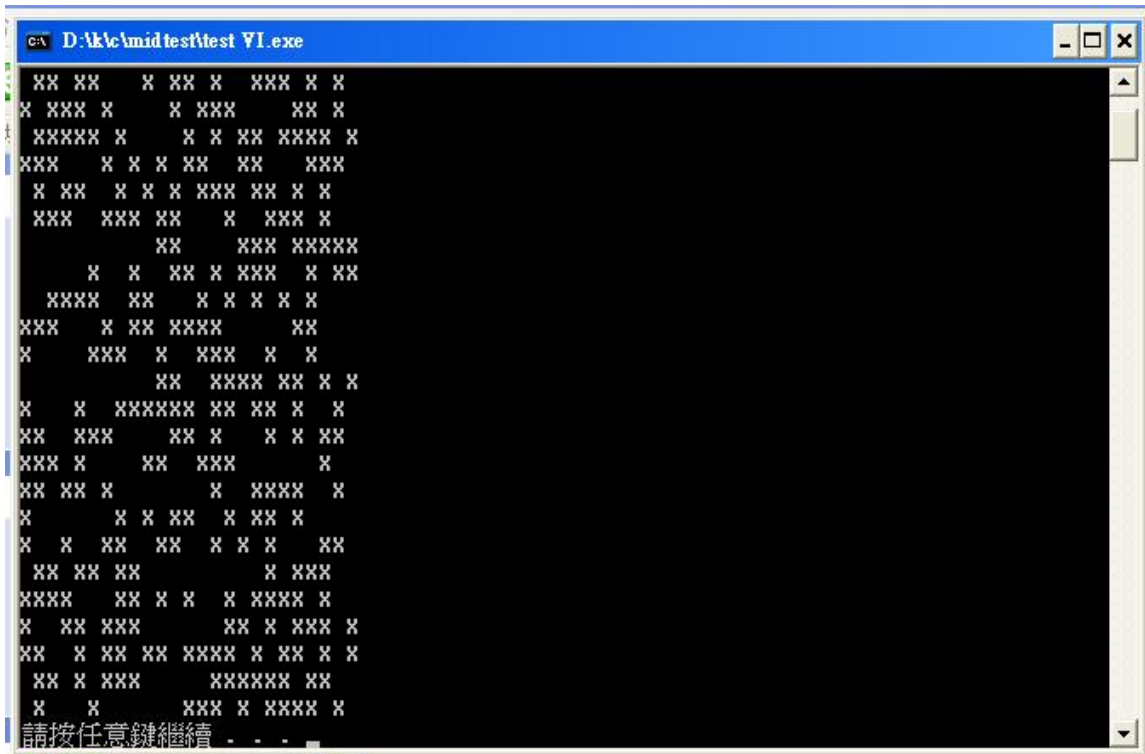
我當時作答成功，其原始碼在「細胞自動機.c」中。這次的組語報告，我便打算改寫這個程式，甚至進一步擴充。

實作內容

在改寫的過程中，我找到了 INT 10h 的使用方法，進而控制 320×200 螢幕上的每一個 pixel，於是便打算擴充原程式的規模。

前 16 行進行各種前設，RandLoop 則產生一個 $320 \times 200 = 16000$ 的隨機陣列，之後繼續改寫，MainLoop 作輸出，Life0Loop 和 LifeLoop 記算周圍有幾格「有」，並配合 LifeStay 判斷是兩格、三格還是其他狀況，之後的 LifeLB 完成整個計算，再由 CleanLoop 完成再次輸出影像前的處理，並利用 INT 16h 由鍵盤讀入一個鍵，如果是 ESC 則中止程式，否則回到 MainLoop，程式繼續。

其中值得一提的是，我無意間找到一個與原條件等價的新判斷條件：對任一格而言，



細胞自動機.exe ↑

其周圍若恰有二格”有”，則該格狀態維持(“有”還是”有”，”無”還是”無”)，若恰為三格”有”，不管原狀態為何，該格狀態改為”有”。其他任何情況則該格狀態為”無”。以上的內容位於 **final.asm** 中。

心得

編譯完成後，組語程式的大小只有 C 的二十分之一以下，而處理的格數是 C 的將近一百倍，看來組合語言在效率與城市大小這兩樣優勢上，確實是難以動搖。不過越是壓縮組語程式的過程，其可讀性與可維護性就越糟，我寫的 **final.asm** 連我有時會搞糊塗，這也是組合語言難以跨越的障礙。

資料來源

Wikipedia

(<http://zh.wikipedia.org/wiki/%E9%A6%96%E9%A1%B5>)

INT 10h Using - OpenSource - 博客園

(<http://www.cnblogs.com/BpLoveGcy/archive/2006/11/21/567976.html>)

释放比特自由

(<http://www.swarmagents.com/vm/articles/freebits.htm>)