Machine Problem 1 - 使用 MD5 進行版本控制

程式名稱

我們將在本作業使用目前上課學到的系統呼叫（read、write、lseek 等等）來實作一個 git 的超級弱化版。這支版本控制的程式該取什麼命字才好呢？維基百科告訴我們原來 git 是 Linus 的自嘲，其意義是「混帳」。但助教連當混帳都不配，就是個人生的輸家，所以我們這次編譯出來的程式請命名爲 “loser” 。

功能

loser 支援三個子指令（都類似於 git ）： status(顯示目前的修改), commit(提交目前的修改), log(顯示過去提交的歷史) 。

不同於 git 能夠記錄過往的所有歷史而擁有自由自在回到任一 commit 的能力， loser 只能夠知道某些檔案曾被修改過而已，並且不需要處理檔案被移除的問題。

loser 的運作原理十分簡單，它會將過去所有訊息記錄於所求目錄下的 .loser_record 檔案，所有的子指令都離不開這個檔案，每次 commit 會新增一組記錄，而 status 會去讀取最後一條 commit 記錄以與當前狀態進行比對，log 則根據要求輸出 ./loser_record 的資料。

但是因爲我們不用像 git 一樣需要回到過去的 commit ，記錄過往的所有資料太浪費空間，我們只需要比較過去檔案的 MD5 摘要即可知道是否產生過變化。

此外，類似於 git 可以用 .gitconfig 這個設定檔來做許多初始設定， loser 也會去讀取 .loser_config 來做設定。