## The data/control flows in your program

資料讀進來 ByteString -> 切塊 [ByteString] -> 依序標上編號 [(Int, ByteString)] ->
丟到 input MVar 裡給 compressor 爭食 ->
collector 不按順序一個一個的收集回來 ->
照編號排好接回來 ( concat . snd . unzip . sortBy (comparing fst) ) -> 寫檔

## The use of semaphores to synchronize threads

在 Haskell 裡有相當多層級的 Concurrency 的抽象
而 Data Parallelism 和 Concurrency 有相當明顯的不同之處（不像在 Imperative 語言裡只有 explicit concurrency 可以使，所以大家都搞混了 … ）

這個作業很明顯的重點在於 Data Parallelism。
而在 Haskell 裡 Data Parallelism 做起來幾乎是不費吹灰之力的無腦。

但是因為這次作業應該是想讓我們淌淌混水，學學信號機要怎麼用 thread 要怎麼開。
所以我也就用 explicit concurrency 的方式來做這題作業了。

我用的是一種叫做 MVar 的一種變數，是一種數值為 1 的信號機與變數的組合。
當他滿的時候 producer 想放東西會卡住，而 cosumer 可以拿東西；空時反之亦然。

程式當中有用到三個 MVar
分別叫 input, output, exit
input 的 producer 是 dispatcher，consumer 是 compressor
output 的 producer 是 compressor，consumer 是 collector
（而 exit 是因為 main thread 是 daemonic 的，要等所有工作做完在結束用的）

## Problems experienced and your solution

遇到的問題主要是對 Haskell 的 Foreign Function Interface (FFI) 和 Concurrency 的不熟悉。

因為作業提供的壓縮程式是 C 寫的函式庫，所以我要想辦法在 Haskell 裡去操作他。

然後還有像是 thread 要怎麼開，開出來要和 OS thread 怎麼做 mapping，怎麼和 FFI 去做配合，和之後執行時的 runtime system 要下什麼參數也研究了很久。

## The performance (time) with respect to # of worker threads

**chunk size: 4096 bytes**

| # of worker threads | elapsed time | | |
|---|---|---|---|
| 1 | 2.33 real | 4.01 user | 0.83 sys |
| 2 | 1.28 real | 2.88 user | 0.61 sys |
| 4 | 0.96 real | 2.44 user | 0.63 sys |
| 8 | 0.94 real | 2.39 user | 0.68 sys |

**chunk size: 8192 bytes**

| # of worker threads | elapsed time | | |
|---|---|---|---|
| 1 | 1.33 real | 2.59 user | 0.54 sys |
| 2 | 0.72 real | 1.83 user | 0.46 sys |
| 4 | 0.81 real | 2.07 user | 0.51 sys |
| 8 | 0.84 real | 2.15 user | 0.53 sys |

## The performance (compression ratio) with respect to different chunk sizes

**input size: 38418300 bytes**

| chunk size | compression ratio |
| --- | --- |
| 1024 bytes | 0.0666 |
| 2048 bytes | 0.0528 |
| 4096 bytes | 0.0458 |
| 8192 bytes | 0.0412 |

## 附註

### 我的破機器：

1.8GHz 雙核心 Intel Core i5 與 3MB 共享 L3 快取
Mac OS X 10.8.2, kernal 是 BSD Darwin 這樣

### Repo:

整個專案也都放在 Github 上了