1. FUSE 的目的，原理。
2. 修改了哪些地方
3. 實作的方法
4. 結果分析

1. FUSE 的目的，原理

FUSE，Filesystem in USErspace，這個開源的library能夠讓我們能在user level 的環境下架構自己的檔案系統，由下圖可以看到我們可以藉由kernel 中的FUSE，來在/tmp/的地方掛載我們的檔案並操作，而在這個final中，簡單的SSD操作便是在FUSE上面所建立的，因為我們想要模擬寫入block及page，FUSE可以來幫我們完成這件事。

Diagram

Description automatically generated

2. 修改了哪些地方

因為傳統機械硬碟與固態硬碟的背後原理不同，因此在操作上的邏輯也會不同，但是為了讓OS端的介面可以用最低的成本來做機械硬碟到固態硬碟的轉換，因此SSD的driver上出現了FTL layer，來協助處理相關的問題。再來因為NAND的物理特性，SSD每次寫入的單位為Page，每次擦除的單位為Block，且一旦page被寫入後，便無法對該page做修改，必須重新寫入至新的page中。但page總會有被寫滿的時候，這時候我們就需要garbage collection 來，回收有用的資料，清空原有的block，這些便是SSD中的FTL主要在做的事情。

而我們主要就是要在原文件中，修改TODO的地方，基本上就是要套一層FTL要做的事情，例如FTL\_read、FTL\_write、Garbage collection、以及自己所需的資料結構。

3.實作的方法

實作上，Garbage collection會是我們的主要重點，以下我會分段的講解我的實作方式。

首先，我會走訪每一個Block，找出stale數量最多的block，並對他選為erase的對象。

Text

Description automatically generated

接著，走訪該block中的每一個page，檢查是否該page的資料仍為使用中的，如果是的話，便要找到新的page來抄寫過去。最後，在維護我們 L2P、P2L的兩張table即可。

Text, letter

Description automatically generated

4. 結果分析

我在我的final中，Physical block 數量為 13，Logical block數量為10。意思是我們會有3個多餘的block來做操控的空間。Over-provisioning (op)就是我們SSD driver自己保留，來做操控的，不同的op 數量會對整體的表現。而我有測試不同數量的，分別是 3、2、1。而以下是測試的結果。







這樣的實驗結果表明，如果我們Physical block 數量大於Logical block數量，我們這樣就可以減少作garbage collection的次數，放低WA的結果。

此外，如果要進一步降低WA，我的猜想是，我們每次分配page的時候，應該要隨機的、平均的，分配在不同的block中，而不是照原本文件中的序列給。