

為了適應 NUM_BYTE_BUF 的需求

- 我更改了 ...
 - 紀錄空間的 Mask 型態 (unsigned char → unsigned char array)
 - 檢查並找出剩餘且合法的空間位置 (location)
 - 維護 Mask 的方法
- 為了找出正確的 Mask bit 位置，我使用了一個簡單的 Mapping 方法。
 - $\text{mask}[\text{index}/8]$ 是決定了這個 bit 在第幾個 Mask
 - $\text{index} \bmod 8$ 決定了這個 bit 是在該mask 的第幾個位置。
 - 以 $\text{NUM_BYTE_BUF} = 23$ 為例。
 - 我們會需要 $\text{NUM_BYTE_BUF} / 8 + 1 = 3$ 個長度為 8 的 mask 來紀錄空間使用情況。
 - 檢查並找出剩餘且合法的空間位置
 - 找到第一個未使用的位置 start
 - 設定 end 為 $\text{start} + \text{size} - 1$
 - 將 end 往前檢查是否有被使用的空間，若有則跳過這段空間的檢查，並從 end 最後檢查到的空閒空間位置繼續新的一輪檢查。
 - set 跟 clean 則依據以上提供找位置的方法使用與 single bit 版本相同的設置方法。

