OS_PROJECT3_REPORT

1. READING CODE:

1 How readahead is called when page faults occur?

1.1 mmap()在 kernel 裡面叫做 mmap_pgoff()，會呼叫 do_mmap_pgoff()

1.2 do_mmap_pg()會 return mmap_regin()

1.3 當 file->f_op->mmap,此 mmap 在 ext4格式的硬碟代表 ext4_file_mmap

1.4 vma->vm_ops = &ext4_file_mmap.

1.5 設置 vm_ops.fault=filemap_fault

1.6 Call filemap_fault()後，用 file_get_page 尋找 page

1.7 找到後會 call do_async_mmap_readahead

1.8 如果是 PageReadahead，call page_cache_async_readahead()

1.9 To do read-ahead，call ondemand_readahead

1.10 較小 size、隨機讀的才 call __do_page_cache_readahead()
(__do_page_cache_readahead 是 readahead 最核心的演算法)

2 Implementation of readahead algorithm：
__do_page_cache_readahead()

2.1 首先，preallocate 我們需要預讀的 pages
在 page_offset 超過 end_index(我們想預讀的最後一個 page)之前
用 radix_tree_lookup 找出 page，注意這要用 RCU 的 readlock
接著將 page 加入 list 中

2.2 開始 I/O
call read_pages 讀出 ret 中的 pages

2. REVISE THE READAHEAD ALGORITHM

On HDD

To reduce the cost of time, we enlarge the amount of pages for readahead algorithm

In mm.h

```
#define VM_MAX_READAHEAD	1024
#define VM_MIN_READAHEAD	512
```

為了提升精準度，我們將 a.out 的 output 輸出至 out 中。

Before revise: 39.24s

```
[ 1887.138183]  page fault test program starts !
[ 1926.378466]  page fault test program ends !
rendybig@rendybig-VirtualBox:~/Downloads/hw3$
```

After revise: 36.1s

```
rendybig@rendybig-VirtualBox:~/Downloads/hw3$ sudo time ./a.out > out
0.01user 0.20system 0:36.10elapsed 0%CPU (0avgtext+0avgdata 106704maxresident)k
100528inputs+160outputs (4204major+6037minor)pagefaults 0swaps
rendybig@rendybig-VirtualBox:~/Downloads/hw3$
```

Due to the message generate in filemap_fault(), we did not calculate time with dmesg.