ECE 695 Task 2 Report

Yizhou Shan <shan13@purdue.edu>
Wan-Eih Huang <huan1031@purdue.edu>

I. Overall

In task 2, we mainly used *pte_none()* and *pte_young()* to identify if a page has been referenced. And we can only count reference to 1.

A page was referenced if and only if these two conditions are true: 1) PTE has valid bit set, which means *pte_none()* returns false, 2) PTE has accessed bit set, which means *pte_young()* returns true. Otherwise, this page was not referenced.

II. Detail

We changed two source files, 1) *fs/proc/base.c*, 2) *fs/proc/task_mmu.c*. Originally, the */proc/PID/maps* is read-only, we make it writable so we are able to turn on or turn off our own feature during runtime.

By default, the */proc/PID/maps* will print its default content.
To enable our feature, type: ***echo y > /proc/PID/maps***.
To reset, type: ***echo n > /proc/PID/maps***.

Our main API is:

```
      show_vma_page_map(struct seq_file *m,
                        struct vm_area_struct *vma,
                        unsigned long start,
                        unsigned long end)
```

which is implemented similar to ***unmap_page_range()*** function. Our function will walk through the user page table through standard kernel pgtable APIs, and check PTE valid and accessed bit.

III. Interesting Things

The last [vsyscall] segment actually does not have an associated mm_struct. The vsyscall and vDSO are both used to accelerate system call, to avoid frequent context switch.