**MP2-implementation**

To simplify the problem, I make some assumption:

1. Virtual address mapped by mmap() for every single process won’t more than 16 at the same time.
2. Total virtual address space size involve in mmap() and munmap() won’t larger than 1G for every single process.
3. Parameter “length” used in mmap() must be a multiple of width of pages.

Also, “addr” and “length” used in munmap() must be a multiple of width of pages.

1. munmap() will either unmap at the start, or at the end, or the whole region (but not punch a hole in the middle of a region).

Map the files information to vma in sys\_mmap(). Do handle\_page() to deal with the situation of page lack and break page. Then, allocate physical memory and map it to vma by enumerating valid vma. Write files page by page and do uvmunmap. If the page haven’t been access yet, skip it. Keep unmap by updating “addr” and “length” until length = 0, then close the fd and label the vma as invalid back. That’s how I manage my vma.