P4 – Design Document

The first change that needs to happen will be in the constructor.

unsigned long \*page\_table = (unsigned long \*)(process\_mem\_pool->get\_frames(1) \* PAGE\_SIZE);

The page table will need to have a frame from the process\_mem\_pool instead of kernel\_mem\_pool in order for the recursive lookup to work.

for (i = 1; i < 1023; i++){

        page\_directory[i] = 0 | 2;

    }

    // Assigning last index in page\_directory back to the page directory

    page\_directory[1023] = (unsigned long) page\_directory | 3;

Here, the last index of the page\_directory will point back to the beginning of the page directory so that we are able to use the recursive trick.

In handle\_fault, there are many more changes that need to be made.

unsigned long page\_table\_address = process\_mem\_pool->get\_frames(1) \* PAGE\_SIZE;

Firstly, the page\_table\_address is using the process\_mem\_pool now instead of the kernel\_mem\_pool.

for (i = 0; i < 1024; i++){

            unsigned long address;

            unsigned long p1\_section = 0xFFC00000;

            unsigned long pd\_section = page\_dir\_index << SHIFT\_12;

            unsigned long i\_section = i << SHIFT\_2;

            address = p1\_section + pd\_section + i\_section;

            trick\_address = (unsigned long\*)address;

            \*trick\_address = 2;

        }

In the for loop that deals with the inner page table, this needs to be modified to accommodate for the trick address. I divided the bit shifting and such into different sections so that it would not be too complicated. The address is comprised of p1\_section (the 10 1s at the beginning to signify the 1023 value), pd\_section (the page directory index), and the i\_section (the value of i in the for loop).

unsigned long address2;

        unsigned long p1\_section = 0xFFC00000;

        unsigned long pd\_section = page\_dir\_index << SHIFT\_12;

        unsigned long pt\_section = page\_table\_index << SHIFT\_2;

        address2 = p1\_section + pd\_section + pt\_section;

        trick\_address = (unsigned long \*) address2;

        page\_table2 = (unsigned long \*)(page\_dir[page\_dir\_index] & 0xFFFFF000);

        \*trick\_address = (process\_mem\_pool->get\_frames(1) \* PAGE\_SIZE) | 3;

The next thing that needs to be changed is in my else statement (which deals with when

page\_dir[page\_dir\_index] & 1

is true. Here, the trick address is a little different in that instead of the i getting used earlier, this one will have the page table index in there, so the construction is a little different (I have pt\_section instead of i\_section). There is still some similar bitshifting involved.