CSCE 410
MP3

Kranthi Kumar Katam
225009204

Page Table:

The page table utilizes the previously created cont frame pool to map virtual memory into physical memory. Implementation is done in two level paging, a page directory which points to 1024 pages and a page which points to 1024 memory spaces. The direct mapping maps the first 1024 pages which is the 4MB of the kernel space. The rest pages are demand based paging i.e., virtual address is mapped to physical address when demanded. These use get_frames function in cont_frame_pool.
Initialization of paging function sets the kernel memory pool and process memory pool and shared size of the kernel memory pool.
The constructor initializes the page directory frame and page table frame address from get_frames. And fill page table with set attribute of 011 as present. And page directory with set attribute of 010 as not present.
The page table load function sets this page to the current page table and then writes page directory pointer to CR3 register.
The enable paging function sets the 32nd bit of CR0 register to 1 which enables memory management unit on the hardware to do paging.
Page fault handler determines what to do if access of a page not in memory. Fault handler reads the page directory from CR3 register. It reads desired access address from CR2 register. It checks with error code. If error code last bit is 1, then check first 10 flag bits are for page directory. The last 12 bits is the offset of the pages and the flags. Then check if page table corresponding to our address is in memory. If page table is in the memory then create a new entry in the page table by calling process memory pool get_frames and then load it into the page table index. If page table is not in memory then create a new page table entry in the page directory by calling kernel memory pool and storing it.