

OPERATING SYSTEMS
CS 611, FALL 2021
Page Table Management

SIRI NAMBURI
331007601

The task at hand is to design a page table management system for x86. Upto now there is no memory management unit designed for the x86 system in consideration. With this page table management system we aim to do paging based on the user's demand. The page table system in x86 follows a two level hierarchy . The first level consists of the page directory, which page the page table pages, and the second level consists of the page tables which page the frames.

The design uses the contiguous frame pools we designed in machine problem 2 to map the virtual memory of the processes to the physical memory. In this problem, we aim to create a page directory with only one valid entry that pages 1024 pages, who in turn page 1024 entries each aiming to create a virtual address space of 4 MB.

The functions defined are :

- Init_paging : we initialize the variables in the page table class
- Constructor : Creates a page table based on the shared memory size and marks the pages as valid/present.
- Load: loads the page table object onto the variable current_page_table
- Enable_paging: conceptually this starts paging in the machine.
- Handle_fault : handles page fault. Basically page fault occurs when we are trying to access a page which is not mapped to any frame. This does so by checking if the fault is at page directory level or the page table level. If the fault is at page directory level we assign a new frame to a new page table and then make the page fault address in the new table as present. Similar steps were taken if the fault is found at page table level.

Testing:

This is already tested by pre written testing on kernel.C