CSCE 611 Operating Systems

MP3 – Page Table Management

Design Document

2nd October 2022

Ram Sankar Koripalli

The aim of this MP is to implement a page table manager to handle the page frames for the memory in an x86 machine.

The structure of the virtual address (32 bit) is as follows:

10bits for PD	10bits for PT	12bit offset
0000 0000 00	0000 0000 00	00 0000 0000

- The page directory table address is stored in cr3 register
- The page table address is calculated using page directory address by masking off the last 12 bits using the mask (FFFFF000)
- The page address is further computed by using mask 3FF

The following APIs are implemented in page_table.C

Page Table constructor

A page table is created for directly mapped memory. Its size is determined by shared memory size. Initially only the shared frames are marked as present and rest of them as not present.

Init paging

The process and kernel memory pools are created here using the work from the previous machine problem continuous frame pool manger

Load

The created page directory table in the constructor is loaded by writing its address in the cr3 register

Enable_paging

The api disables direct mapping and enables paging by setting the cr0 register and paging_enables variable.

Handle_fault

To handle a page fault, we first check whether the fault is in page directory or page table. This is done by check the entry from page directory whose index is received via the cr2 register.

The virtual address is received from the cr2 register , then we compute the corresponding page directory, page table and page address using the above stated masks

If a page fault occurs at directory level, a page table entry is created by allocating a frame from the kernel pool, then a page entry is created in the page table by allocating a frame

If a page fault occurs at page table level, only a page entry is created by requesting a frame from the process frame pool.

We use the page directory address determine whether the fault occurs at page directory level or page table level

.

RESULTS

The tests present in the kernel.C executed successfully with no errors.