Machine Problem 3

OBJECTIVE - The objective of this machine problem is to get started on a demand-paging based virtual memory system for our kernel.

OVERVIEW -
we need to implement the below functions in page_table.C -

1) PageTable() - class constructor.
2) init_paging() - sets the global parameters for the paging system.
3) load() - sets the given page table as the current page table.
4) enable_paging() - enables paging on the CPU.
5) handle_fault() - this is the page fault handler.

IMPLEMENTATION-
1 file changed - page_table.C

cont_frame_pool.C and cont_frame_pool.H are replaced with mine.

init_paging() - sets the global parameters/pointers i.e. process_mem_pool, kernel_mem_pool, and shared_size.

PageTable() -

1) allocates one frame for the page directory in the kernel memory pool.
2) Allocates one frame for the first table in the kernel memory pool because it is simple to create a first page table before paging is enabled.
3) Directly map the memory in the first page table and enable R/W and present bit.
4) Store the address of the first page table in the first page directory entry and mark the rest as with R/W bit set.

load()-

1) Sets the current page table pointer to the given page table.

enable_paging()-

1) Write the address of the page directory in the CR3 register.
2) Set the paging_enabled variable to 1.
3) Set the paging bit in CR0(bit 31).

handle_fault()-

1) Load the err_code.
2) Load the address where fault occurred from cr2 register and Decode the page directory index and page table index from this.
3) Read the page_directory address from cr3 register.
4) If present bit is 0 then page is not present, then we check whether page directory has the page table entry or not at our decoded index, if it's there then the issue is in page table, we will allocate frame to the page table entry.
   If the page directory has no entry we will create a page table and then allocate frame to the page table entry.

Testing -
Completed testing using kernel.C and checked by changing process memory pool size, the code was able to pass the tests.