MP3: Page Manager 1

Dhanraj Murali
UIN: 734003894
CSCE611: Operating System

Assigned Tasks:

Main: Completed.

System Design:

1. Total memory size = 32MB
2. Memory reserved for kernel = 4MB
3. Memory partition to be used by kernel = 2MB to 4MB
4. Memory partition to be used by process = Above 4MB

Code Description:

I. Paging enabled by loading the page table address in the CR3. Paging bit is set in CR0.

II. Page fault is handled by allocating the required number of frames for PTE or PDE using the contiguous frame pool from MP2 submission

Files modified:

a. page_table.H
b. page_table.C
c. kernel.C – modified to add some debug points. Reverted to original file.

All changes made as part of MP1 bonus to enable GDB integration in the following files are also included in this repository as well.

a. Makefile
b. bochsrc.bxrc

Function Description:

a. init_paging

Logic Used:

Paging parameters are initialized by loading the 3 following parameters.

1. Kernel memory pool – 2MB to 4MB
2. Process memory pool – 4MB to 32MB
3. Shared memory size – 4MB – 2MB = 2MB

b. PageTable() - Constructor

Logic Used:

Constructor creates a page directory by getting a frame from the kernel memory pool using *get_frames* function in “cont_frame_pool.C”. Once page directory is allocated a frame, all entries are initialized and write bit is enabled. For the shared memory space, a page table is created. A frame is allocated using *get_frame* function and then all entries of the page table are initiated with write and present enabled. The corresponding PDE is updated in page directory with the first address of the page table and write/present is enabled.

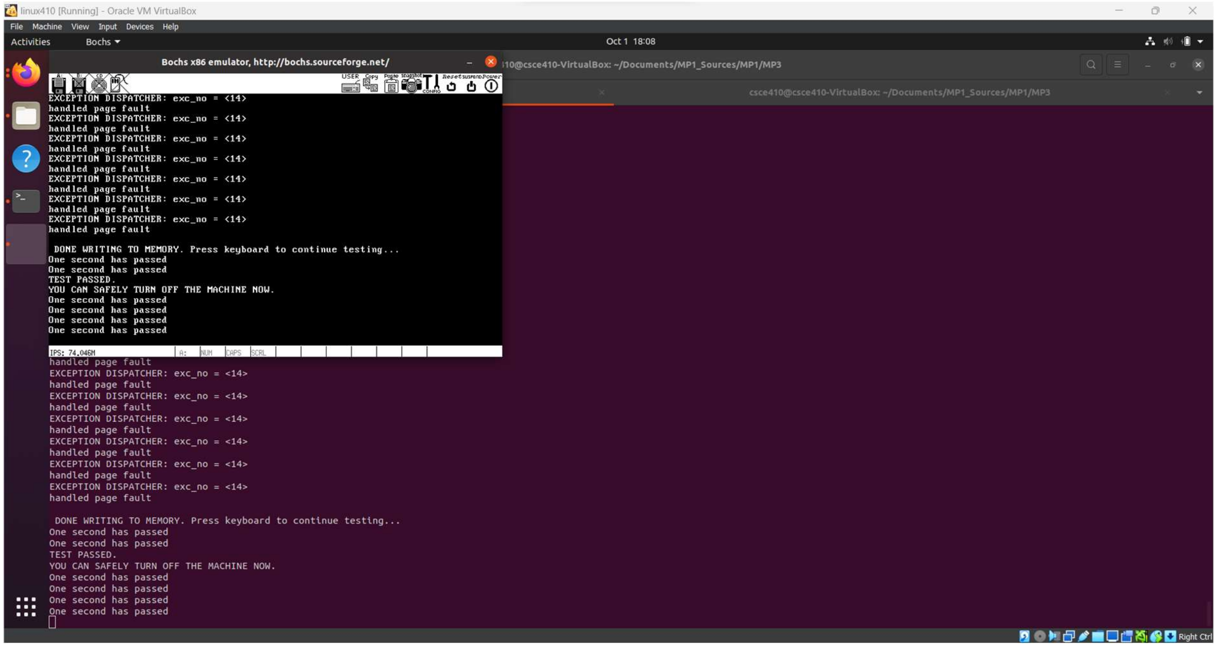
c. **enable_paging**
   Logic Used:
   Paging is enabled by setting the MSB of CR0 as 1. This bit represents paging status and should be set when paging needs to be enabled.

d. **load_paging**
   Logic used:
   The address of the page directory i.e page table base register is loaded onto the CR3.

e. **handle_faults**
   Logic Used:
   Here we have 2 types of page faults.
   I. Page table is present but page table entry is not present. So, page is faulting in page table. We just have to get a frame for the required page and update the page table with the address and enable the write/present/user bits.
   II. Page table is faulting in page directory. Now we have to get a frame for a PDE and update the write bits. Then initialize the entire page table as user and write bit enabled. Update the page directory with the address of the new page table. Now get a page for the PTE corresponding to the faulting address and update the page table with it.

   In this function we try to manipulate the faulting address in such a way that we are able to extract the 3 parts of the address.
   The address is separated into 3, 10 bits for PDE, 10 bits for PTE and 12 bits for the offset.

Testing:


linux410 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Bochs
Oct 1 18:08
Bochs x86 emulator, http://bochs.sourceforge.net/
USER Copy Paste snapshot CONFIG Reset SUSPEND Power
csce410@csce410-VirtualBox: ~/Documents/MP1_Sources/MP1/MP3
IPS: 74.046M A: NUM CAPS SCRL
Right Ctrl

```
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault

 DONE WRITING TO MEMORY. Press keyboard to continue testing...
One second has passed
One second has passed
TEST PASSED.
YOU CAN SAFELY TURN OFF THE MACHINE NOW.
One second has passed
One second has passed
One second has passed
One second has passed
```

```
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault
EXCEPTION DISPATCHER: exc_no = <14>
handled page fault

 DONE WRITING TO MEMORY. Press keyboard to continue testing...
One second has passed
One second has passed
TEST PASSED.
YOU CAN SAFELY TURN OFF THE MACHINE NOW.
One second has passed
One second has passed
One second has passed
One second has passed
```