Machine problem-4:

For this MP, our goal is to support large virtual address spaces and this can be achieved by implementing the below 3 things.

Part 1: We need to move the page tables from the kernel where it is directly mapped into virtual memory. We need to implement recursive page lookup.

Part 2: Prepare the page table to support changes below.

Part 3: Implement virtual memory allocator.

Design:

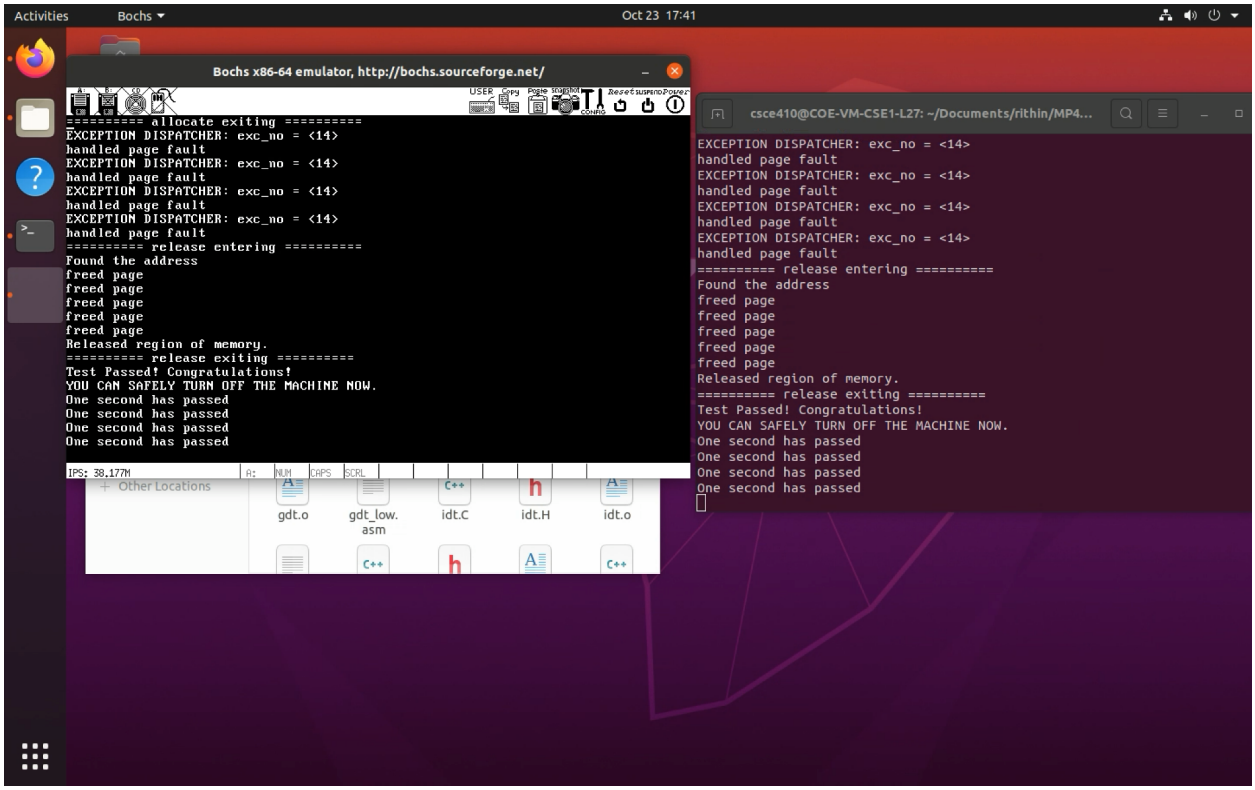
Part1: here we implement the recursive page table lookup for the old page table. It is observed that the virtual memory assertions will fail in this case when the tables are in kernel memory. To address this we need to increase the address space of the page table and directory. In the recursive page table approach, the last entry in both the page table and page directory are pointed back to itself.

Part2: Here we add an extra functionality to the page table. We write a register_pool() method which keeps track of the multiple virtual memory pools which are linked to the given page table.

Part3: Here we implement the virtual memory pool class.

- Construtor: This initializes the base address and size of the virtual memory pool. This also maintains to which page table the VM pool belongs to.
- allocate(): This simply allocates the virtual memory of required size, and returns the start address.
- release(): It frees up the memory region when provided with the base address. It further calls methods to deallocate/ release frames from process frame pool.
- legitimate(): This tells if the given address falls under the VM pool or not.

Testing:


Activities
Bochs
Oct 23 17:41
Bochs x86-64 emulator, http://bochs.sourceforge.net/
========== allocate exiting ==========
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
========== release entering ==========
Found the address
freed page
freed page
freed page
freed page
freed page
Released region of memory.
========== release exiting ==========
Test Passed! Congratulations!
YOU CAN SAFELY TURN OFF THE MACHINE NOW.
One second has passed
One second has passed
One second has passed
One second has passed
IPS: 38.177M
csce410@COE-VM-CSE1-L27: ~/Documents/rithin/MP4...
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
========== release entering ==========
Found the address
freed page
freed page
freed page
freed page
freed page
Released region of memory.
========== release exiting ==========
Test Passed! Congratulations!
YOU CAN SAFELY TURN OFF THE MACHINE NOW.
One second has passed
One second has passed
One second has passed
One second has passed
Other Locations
gdt.o
gdt_low.asm
idt.C
idt.H
idt.o