CS 486: Assignment 5
Page Table Create

Jon A. Solworth

Due: 15 Apr 2015

Due:	**Midnight 15 Apr 2015**
Demo:	Working code

1 Overview

NanoOS should fit comfortably within 4 MB of memory. No matter what your kernel size is, Xen will allocate a multiple of 4MB of mapped virtual address space. We'll call the size of this mapped address space S.

Xen enables you to request, via the config file, more memory than S. In this assignment the config file requested memory should be $S + 4MB$. You should write a function

```
virtualMap(vaddr_t start, long pageCount, long *mfn)
```

Where start is the initial virtual memory address, pageCount is the number of pages to be mapped, and mfn is the machine frame numbers which will be mapped. Thes pages should be mapped in PHYSMAP order.

Xen guarantees that at least 1/2 MB of S is free (the last 1/2 MB). You can use this 1/2 MB (128 pages) to create 3rd level page table nodes during initialization. In x86 PAE, each third-level page spans 2MB of virtual memory address space. Thus to add 4 MB to the address space,

- zero out 2 pages of the 1/2 MB
- make those pages read only
- make the entries in the second-level page block point to the 2 new third level pages.
- make the third-level pages point to the additional memory.

To do the last two steps, you'll need to invoke HYPERVISOR_mmu_update, both to increase the address space and to manipulate the page flags to make the page read-only.

Note at this point that these 2 physical pages are in the page table at two points; at the leaves (so they have virtual addresses) and at the third level of the page tree to provide address space. This provides 2*512*4096 bytes of address space, or 2^{21} bytes.

In general, when you are updating page tables, you need to invalidate cached page information kept in the core. This is a bit difficult to do on Intel architectures, the easiest way to do this is by reloading CR3 after making the page table changes (but before accessing the new pages). To do this, you'll need MMUEXT_NEW_BASEPTR on HYPERVISOR_mmuext_op.

You should put in as many check as possible to ensure that these interfaces are being used correctly and are returning correct values.

2 Background

In order to complete this assignment, you should be (or become) familiar with the following concepts: Page tables. These topics will be/have been covered in class; further information may be found in the text, on the course webpage, and in the Xen headers.