CST 334 (Operating Systems)

Dr. Glenn Bruns

# Lab: Paging

We're going to practice address translation with paging. Here's the situation:

* + a page is 256 bytes in size
  + there are 8 pages in the virtual address space
  + there are 32 pages in the physical address space

1. How many bits will there be in a virtual address? Breaking it down, how many bits do we need in the VPN part? How many bits do we need in the offset part? (See hints at the end of this lab.)
2. How many bits in a page frame number (PFN)?
3. The page table looks like this:

|  |
| --- |
| 0x8000000d |
| 0x80000006 |
| 0x8000001a |
| 0x80000005 |
| 0x8000000a |
| 0x00000000 |
| 0x8000001f |
| 0x00000000 |

The first row gives information about VPN 0, the second row gives information about VPN 1, etc. In each row the leftmost bit says whether the VPN is valid or not. So if you see that the leftmost hex digit is 8, then is binary 1000, so the VPN is valid. Otherwise that virtual page is not valid.

The remaining bits in each row give the physical frame number. For example, the value in row 0 give the physical frame name for VPN 0, the value in row 1 gives the physical frame number for VPN 1, etc.

1. Is VPN 3 valid? Use the page table to decide.
2. What is the PFN for VPN 1? Use the page table. First check if the VPN is valid. If so, get the PFN.
3. Let's translate a virtual address to a physical address. We'll try virtual address 0x06e1. In binary it looks like this:

110 1110 0001

The red bits are the offset part. What is the VPN for this address?

1. How many offset bits in a physical address (in our scenario)?
2. What is the PFN for the VPN in the virtual address above?
3. Putting everything together, what is the physical address for the virtual address above?
4. The physical address is this:

1 1111 1110 0001

Which is 0x1FE1 in hexadecimal. The VPN for the virtual address is 110 binary, which is 6 decimal. Looking at the page table, the row for VPN 6 is the seventh row, because we start counting at 0.

The value in the row starts with an 8, so VPN 6 is valid. What is the corresponding PFN? The value in the row is 1f, which in binary is 11111. This is the PFN corresponding to VPN 6. To get the virtual address we put the VPN in front of the offset. The offset is always the same in the virtual and the physical address.

1. Now you translate the following virtual addresses. In each case either write 'invalid', or give a physical address.

0x04d3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0x030d \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0x0244 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0x0566 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0x03a7 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0x057c \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0x054b \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0x0110 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0x0624 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0x07db \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0x07c1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. If you still have time, copy these files to your home directory on hosting: (or get from OSTEP site)

/home/CLASSES/brunsglenn/OSTEP/HW-Paging-LinearTranslate/paging-linear-translate.py

/home/CLASSES/brunsglenn/OSTEP/HW-Paging-LinearTranslate/README-paging-linear-translate

1. Run the program as follows from your home directory:

./paging-linear-translate.py

Your job is to translate virtual addresses to physical addresses using a paging table.

Play with the simulator and create some problems for yourself. To create the problems in whic lab I used:

$ ./paging-linear-translate.py -P 256 -a 2k -p 8k -c -s 10 -n12

## Hints:

1. There are 8 virtual pages, so we need 3 bits in the VPN part. With 3 bits we can identify any of 8 pages. We need 8 bits for the offset part, because pages are 256 bytes in size. With 8 offset bits we can identify any of the 256 bytes on a page. So the total number of bits in a virtual address is 11 (in other words 3 + 8).
2. There are 32 physical pages, so we need 5 bits to identify them. The total number of bits in a physical address is therefore 5 + 8.
3. -
4. Yes, the 4th row in the page table starts with bit value of 1.
5. For info on VPN 1, we look at the second row of the page table. It is a valid VPN, and the offset is 6, which in binary is 0110.
6. The VPN is 110
7. 8, because there are 256 bytes in a page. You need 8 bits to identify any byte in a page of 256 bytes.
8. 0x1F, or 11111 binary
9. 1 1111 1110 0001
10. -
11. The answers are:

00000ad3

0000050d

00001a44

Invalid

000005a7

Invalid

Invalid

00000610

00001f24

Invalid

Invalid

1. -
2. -
3. -
4. -