

paging-multilevel-translate.py

This fun little homework tests if you understand multi-level page tables.
Some basic assumptions:

- The page size is an unrealistically-small 32 bytes
- The virtual address space is 1024 pages, or 32 KB
- Physical memory consists of 128 pages

Thus, a virtual address needs 15 bits (5 for the offset, 10 for the VPN).
A physical address requires 12 bits (5 offset, 7 for the PFN).

The system assumes a multi-level page table. Thus, the upper five bits of a virtual address are used to index into a page directory; the page directory entry (PDE), if valid, points to a page of the page table. Each page table holds 32 page-table entries (PTEs). Each PTE, if valid, holds the desired translation (physical frame number, or PFN) of the virtual page in question.

The format of a PTE is thus:

VALID | PFN6 ... PFN0
and is thus 8 bits or 1 byte.

The format of a PDE is essentially identical:

VALID | PT6 ... PT0

You are given two pieces of information to begin with. First, you are given the value of the page directory base register (PDBR), which tells you which page the page directory is located upon. Second, you are given a complete dump of each page of memory. A page dump looks like this:

```
page 0: 08 00 01 15 11 1d 1d 1c 01 17 15 14 16 1b 13 0b ...
page 1: 19 05 1e 13 02 16 1e 0c 15 09 06 16 00 19 10 03 ...
...
```

which shows the 32 bytes found on pages 0, 1, 2, and so forth. The first byte (0th byte) on page 0 has the value 0x08, the second is 0x00, etc.

You are then given a list of virtual addresses to translate.

Use the PDBR to find the relevant page table entries for this virtual page. Then find if it is valid. If so, use the translation to form a final physical address. Using this address, you can find the final VALUE fetched from memory.

(note: seed = 1000)

```
prompt> ./paging-multilevel-translate.py -s 1000 -c | grep "page 99"
page 99: 90 86 f3 e5 a6 d5 a7 85 cb 98 97 7f e2 d3 d7 9b
       7f dc c2 bc a2 ad e8 f1 f7 e0 a1 ac a8 de 8f f5
prompt> ./paging-multilevel-translate.py -s 1000 -c | grep "page 119"
page 119: 7f 7f 7f 7f 7f ef 7f 7f 7f 7f 7f 7f 7f a3 7f 7f
       7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
prompt> ./paging-multilevel-translate.py -s 1000 -c | grep "page 111"
page 111: 07 1b 17 0c 0a 11 12 05 0f 1d 0e 0c 1c 1e 1d 01
       1a 06 0a 1b 03 04 08 06 0a 16 13 04 17 1c 12 05
```

PDBR: 99 (decimal) [This means the page directory is held in this page]

Virtual address 60b8: translates to what Phys Addr (and fetches what Value?), or Fault?