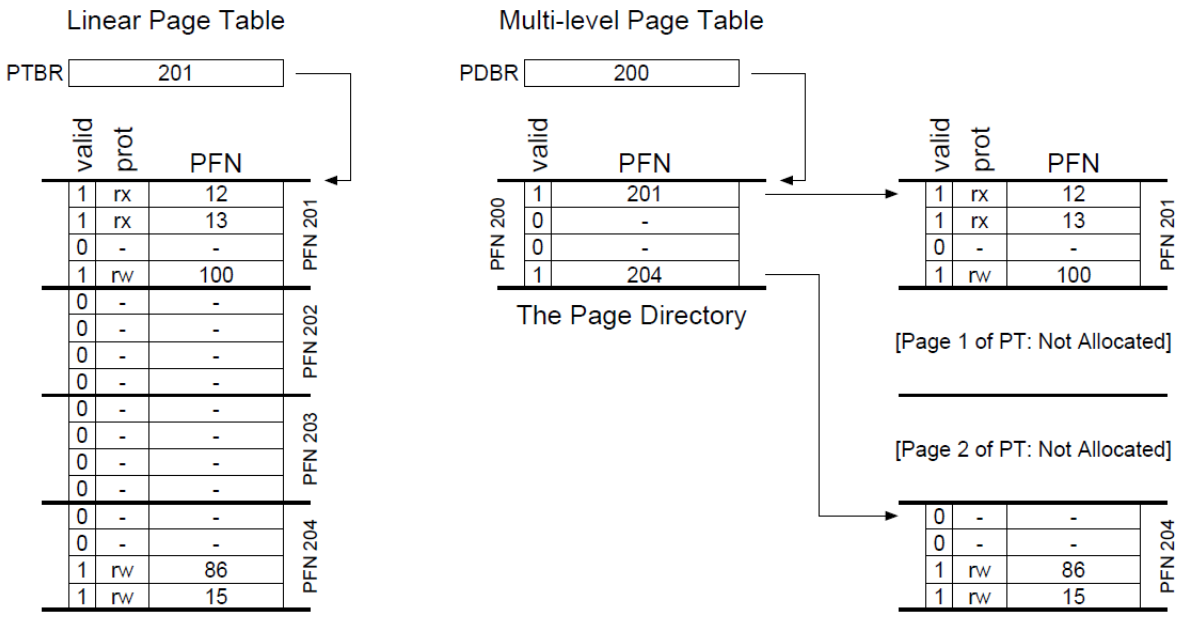

Translations with Multi-Level Paging

Assume a system with the following characteristics:

- A virtual address space is 32KB
- Physical memory is 4KB
- The page size is 32B
- A page table entry (PTE) is 1 byte

How many bits are needed for a virtual address?

How many bits are needed for the VPN, and how many for the offset?

How many bits are needed for the physical address?

How many bits are needed for the PFN, and how many for the offset?

In a linear page table, how many PTEs are there?

As stated above, a PTE is 1B. How big is the linear page table?

Now let's assume a multi-level page table. We want to break our linear page table into many pages, which we'll access via a page directory. How many pages will we need to represent our entire page table?

How many entries will there be in the page directory?

How many bits of our virtual address will be needed to index into the page directory? These bits constitute the page-directory index (PDIndex).

Thus, our PDE is located at $PageDirBase + (PDIndex \times sizeof(PDE))$.

If we find a valid PDE, we need to create an page-table index (PTIndex) into the corresponding page table page. How many bits will be necessary to index into a page table page?

So, for a given virtual address, indicate which bits are the PDIndex, which are the PTIndex, and which are the offset:

1 1 1 0 0 1 1 1 0 0 1 0 0 0 1

We won't do a translation now, but to do one, you'll also need the PTE, which is located at $(PDE.PFN << SHIFT) + (PTIndex \times sizeof(PTE))$

Now that you've worked through how to deconstruct a virtual address to allow access to the PDE and the PTE, you should be all set to perform address translations. Use the homework at the end of the book chapter for practice.