# Virtual Memory

Application address and system memory can be totally decoupled

图片包含 文档

描述已自动生成

Map from program-view address to physical-view address (use page table)

* Frame: physical cache line
* Page: memory block

Virtual – physical translation

图片包含 文字

描述已自动生成

图片包含 文字

描述已自动生成

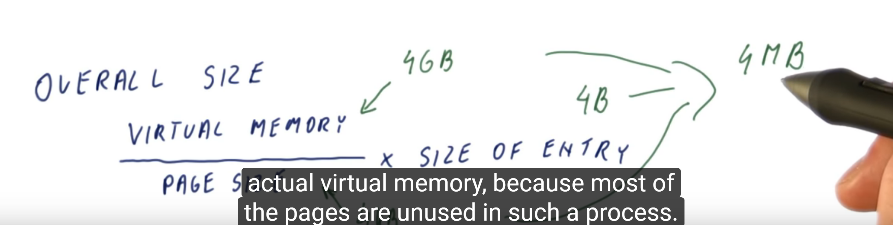
With 4 page-table entries -> leading 2 bits in virtual address is index, page offset 14bit. The physical address index is in page table.

## Flat Page Table

Size of flat page table (one entry for all virtual address space)

1. Bits for actual address
2. Extra bits for validation

Flat page table is large (lots of page is unused in page table)



## Multi-level Page Tables

Flat table + Avoid use large page tables with lots of entries unused

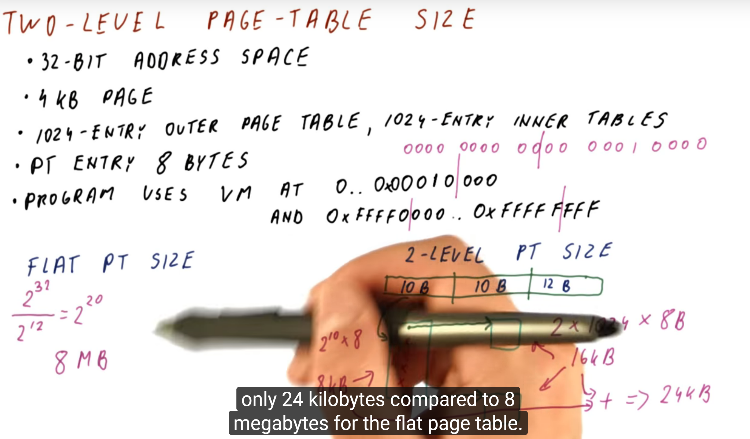
* Partition page number to inner page number (which specific entry in that part of page table to use) and outer page number (which part of page table to use) -> the total size of inner page table will be the size of original flat page table. The overhead is the outer page table. If not used, the pointer will pointer nothing in outer page table.

图片包含 文字, 地图

描述已自动生成

图片包含 文字

描述已自动生成



Only last two level of page tables will be used.

图片包含 人员, 文字

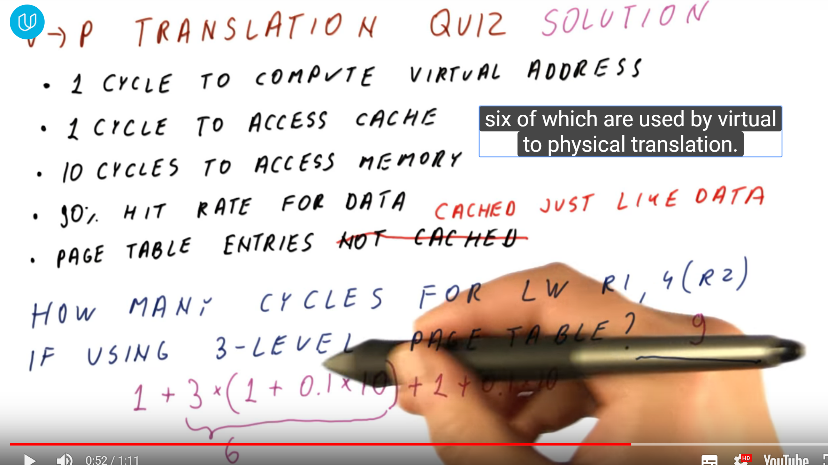
描述已自动生成

## Page Size

Large page size -> small page table, fewer entries

But for large page size -> internal fragmentation (since memory allocated in page manner)

Page table need to be in memory -> we need page table to get physical address



Need Translation Look-Aside Buffer (TLB) for caching page tables

* Cache is big, while TLB is small (thus fast)
* TLB only stores only the frame number (not intermediate, no multi-level)
* When TLB miss, use page table to translation and put page number in TLB
* TLB can be replaced by Operating System (software TLB miss handling) or Processor automatically read page tables and update TLB (hardware TLB miss handling, fast)

图片包含 人员

描述已自动生成

If cache only access 32Kb data -> just need 8 TLB entries to cover (32kB/4kB), HOWEVER, it is possible that each block in cache is in different page, thus require 512 pages (512 TLB entries) -> choose 8 to 512 as choice

Since TLB is small, just use Fully Associative (high associative) -> multi-level TLB (not make TLB large) L1, L2

图片包含 文字

描述已自动生成

1MB/4kB = 256 pages -> L1 only 128 -> 10 \* 1 \* 256 miss in Level 1, 10 \*4095 \* 256 hit

L2 hit = L1 miss – L2 miss, L2 miss = 256