Lancaster University
Leipzig

VMA Translation in Multi-process OS

LZMSCI521M | Week 1

Dr Nono Saha

What Will You Learn Today?

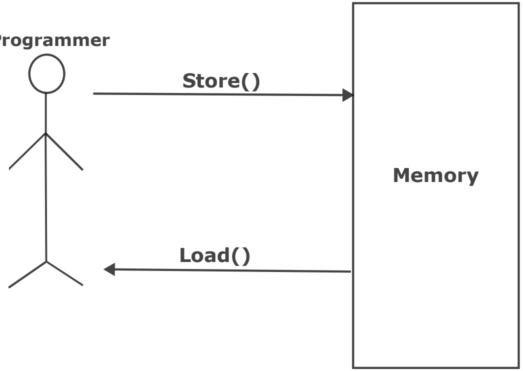
1.1

Virtual Memory

For Second-year Computer Science Students

University of Leipzig/ ScaDS.AI

Why Virtual Memory?



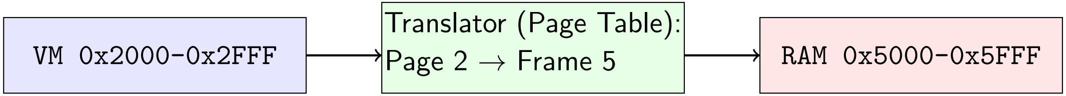
Possible problems

1. Computers have limited physical memory, but modern programs expect a large address space.
2. Different programs (processes) need to be isolated: can't let one process overwrite another's memory!
3. The ISA often supports a larger address space than there is physical memory.

Virtual Memory Concepts

- **Idea**: Give the programmer the illusion of a large addressspace while having a small physical memory.
  - So that the programmer does not worry about managing physical memory.
- Programmer can assume having an "infinite" amount of physical memory
- Hardware and software cooperatively and automatically manage the physical memory space to provide the illusion
  - Illusion is maintained for each independent process

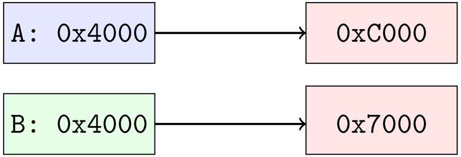
Example: Suppose a process wants to read byte `0x2400` (virtual).



Thus, virtual address `0x2400` maps to physical address `0x5400`.

Example of Translation in Multi-Process Situation

Process A: Virtual `0x4000 (Page 4)` → Frame 12 → `0xC000`
Process B: Virtual `0x4000 (Page 4)` → Frame 7 → `0x7000`

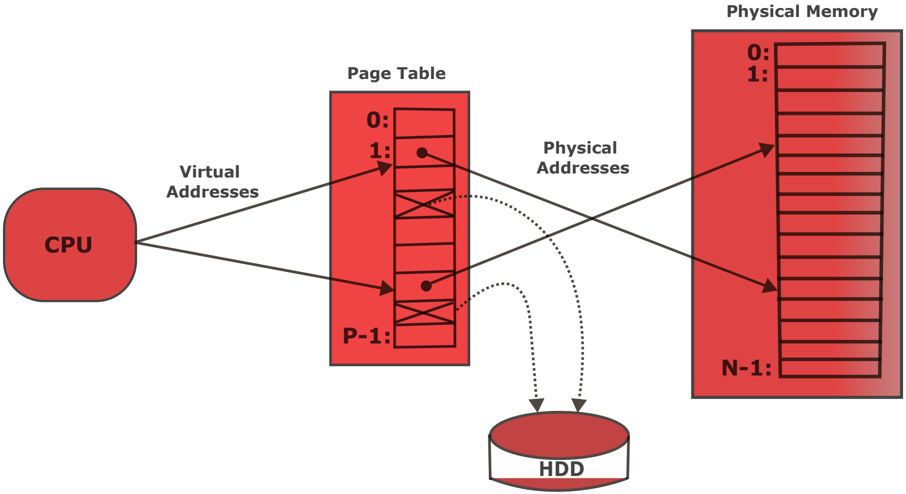


Both processes use the same virtual address but access different physical locations.

Basic Mechanism

- Indirection (in addressing)
- Address generated by each instruction in a program is a “virtual address”
  - i.e., it is not the physical address used to address main memory
  - called “linear address” in `x86`
- An “address translation” mechanism maps this address to a “physical address”
  - called “real address” in `x86q`
  - Address translation mechanism can be implemented in hardware and software together

A system with Virtual Memory (Page based)



- **Address Translation**: The hardware converts virtual addresses into physical addresses via an OS-managed lookup table (page table).

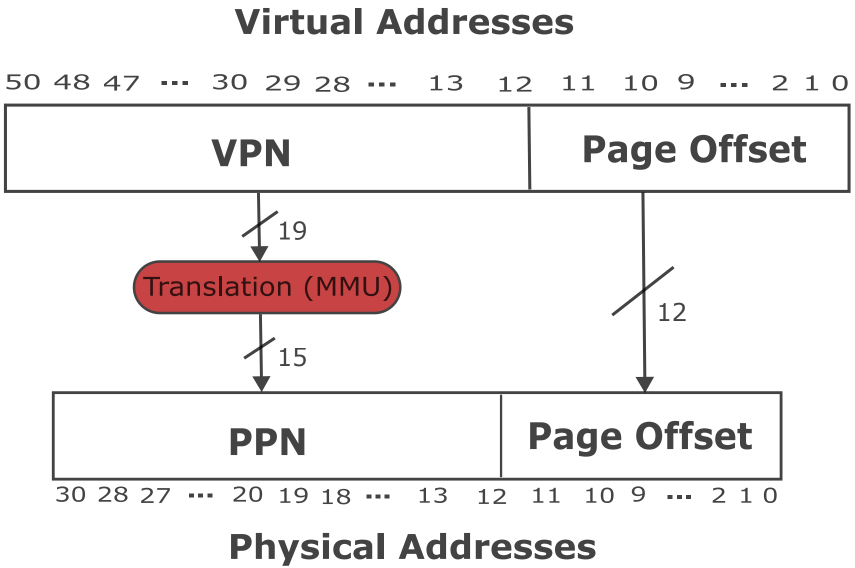
1.2

Address Translation

For Second-year Computer Science Students

University of Leipzig/ ScaDS.AI

Address Translation



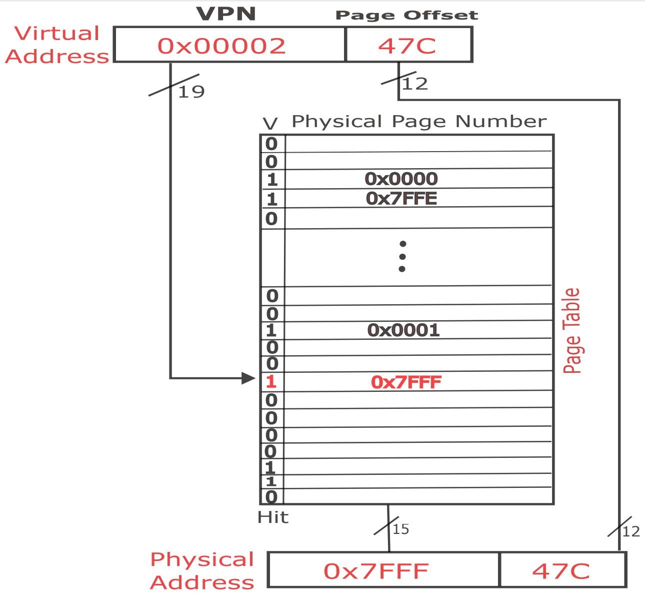
Virtual Memory Example

- System:
  - Virtual Memory capacity: 2 GB $= 2^{31}$ bytes
  - Physical Memory capacity: 128 MB $= 2^{27}$ bytes
  - Page size: 4 KB $= 2^{12}$ bytes
- Organisation:
  - Virtual address: 31 bits
  - Physical address: 27 bits
  - Page offset: 12 bits
  - Virtual pages $= 2^{31}/2^{12} = \mathbf{2^{19}}$ (VPN $=$ 19 bits)
  - Physical pages $= 2^{27}/2^{12} = \mathbf{2^{15}}$ (PPN $=$ 15 bits)

How Do We translate Addresses?

- Page table
  - Has entry for each virtual page
- Each page table entry has:
  - Valid bit: whether the virtual page is located in physical memory (if not, it must be fetched from the hard disk)
  - Physical page number: where the virtual page is located inphysical memory
  - (Replacement policy, dirty bits)

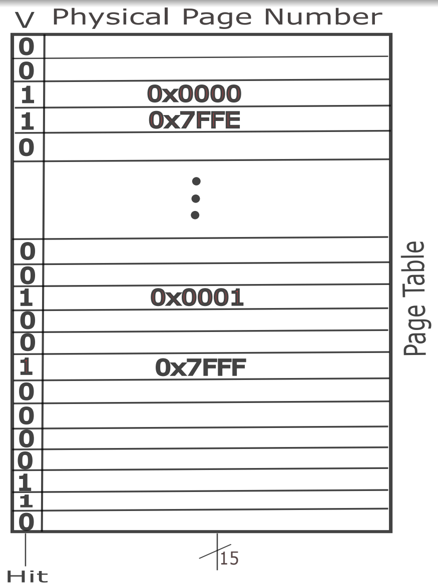
Page Table Example



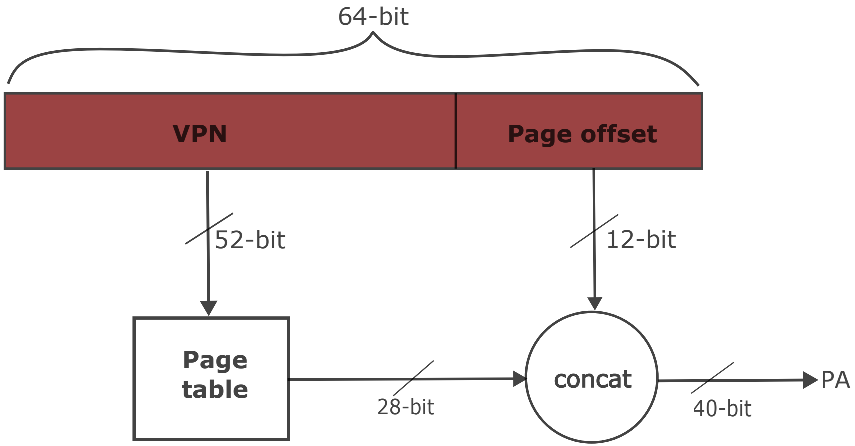
Page Table Example (Homework 1)

Consider the PPN on the right.

1. What is the physical address of virtual address **0x5F20**?
2. What is the physicaladdress of virtual address **0x73E0**?



Issues with the Page Table Size



- Suppose a 64−bit VA and a 40−bit PA, how large is the page table?

Page Table Challenges

- Page table accesses have a lot of temporal locality
  - at least part of it needs to be located in physical memory
  - Data accesses have temporal and spatial locality
  - Large page size (say 4KB, 8KB, or even 1-2GB), so consecutive loads/stores likely to access same page
- Each load/store requires at least two memory accesses:
  1. one for address translation (page table read)
  2. one to access data with the physical address (after translation)
- Two memory accesses to service a load/store greatly degrades load/store execution time
  - Unless we are clever. . . 31

1.3

Translation Lookaside Buffer (TLB)

For Second-year Computer Science Students

University of Leipzig/ ScaDS.AI

Translation Lookaside Buffer (TLB)

Idea: Cache the page table entries (PTEs) in a hardwarestructure in the processor

Advantages:

- Small cache of most recently used translations (PTEs)
- Reduces number of memory accesses required for mostloads/stores to only one
- High associativity
- Typically 16 - 512 entries
- $>$ 95-99 % hit rates typical (depends on workload)

Multiprossessing and Address Translation

Memory Protection

- Multiple programs (processes) run at once
  - Each process has its own page table
  - Each process can use entire virtual address space without worrying about where other programs are
- A process can only access physical pages mapped in its page table—cannot overwrite memory of another process
  - Provides protection and isolation between processes
  - Enables access control mechanisms per page

Page Table is per Process

- Each process has its own virtual address space
  - Full address space for each program
  - Simplifies memory allocation, sharing, linking and loading.

1.4
Takeaway

For Second-year Computer Science Students

University of Leipzig/ ScaDS.AI

Key Takeaways

- **Virtual memory** gives each process a large, private address space.
- The **OS and hardware** jointly translate virtual addresses to physical addresses.
- **Page tables** do the mapping; **TLB** makes it fast.
- Per-process page tables enforce **memory protection**.
- A subset of virtual pages are located in physical memory

1.5
Practical Exercises

For Second-year Computer Science Students

University of Leipzig/ ScaDS.AI

Practical Exercises

1. Consider a virtual address space of 64 pages of 1024 words each, mapped onto a physical memory of 32 frames.
   a. How many bits are there in the logical address?
   b. How many bits are there in the physical address?
2. Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):
   a. 3085
   b. 42095
   c. 2000001
3. Suppose process X and process Y both have a virtual page 2.
   a. X's page 2 maps to frame 13; Y's page 2 to frame 21.
   b. If Y reads from virtual address `0x2004` (4KB pages), what physical address in RAM is accessed?

Lancaster University
Leipzig

1.6

What Next?

For Second-year Computer Science Students

University of Leipzig/ ScaDS.AI

What next?

Page Fault and Page replacement, Structure of Page tables.

Beyond Virtual Memory

1. Cache Memory (Cache of Level 1, 2, 3 and 4)
2. Swapping Memory
3. IA-32 and x86-64 Memory Architecture

Some Important Materials

- Operating System Concepts, Abraham Silbershatz, Peter Baer G. and Greg Gagne, Tenth Edition, 2018.
- Youtube Course on Virtual Memory by Prof. Onur Mutlu, ETH Zürich
- Code and Course Material