Example

Assume a computer with 32-bit addresses, paging, 64KB of memory, 4KB pages and LRU replacement.

In this computer a program is executed with the following features:

- The code starts in address 0x0A020 and its size is 2KB.
- Every time the code is executed it needs to read 8 images, 2KB each, stored in the following addresses: 22000, 25000, 28000, 32000, 1B000, 2B900, 1C000, 0E800.
- Assume that memory is empty when the program begins.

Answer the following questions:

1. Virtual address and physical address format.
2. Size of the address space. Number of pages of the process. Size of the page table.
3. Evolution of the main memory and page table during the execution of the program.
4. 2-level page table

Example: 1) Virtual address

Assume a computer with 32-bit addresses, paging, 64KB of memory, 4KB pages and LRU replacement.

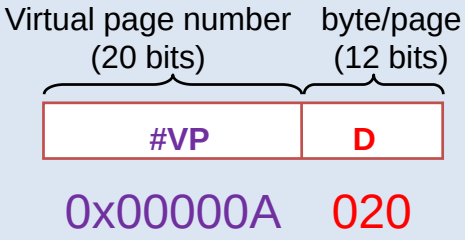
Code Virtual Address: 0x0000A020

Address space: 2^{32} bytes = 4 GB

Page size: 4 KB = 2^{12} bytes

1.048.576 pages

Fields of the virtual address



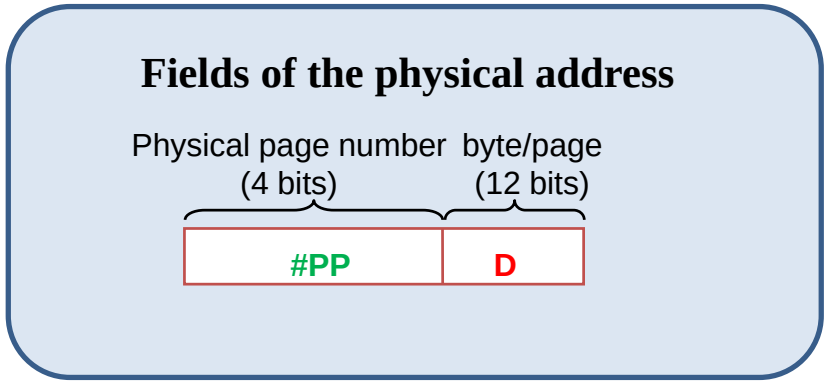
Example: 2) Physical address

Assume a computer with 32-bit addresses, paging, 64KB of memory, 4KB pages and LRU replacement.

Memory: 64 KB = 2^{16} bytes

So, physical addresses have 16 bits

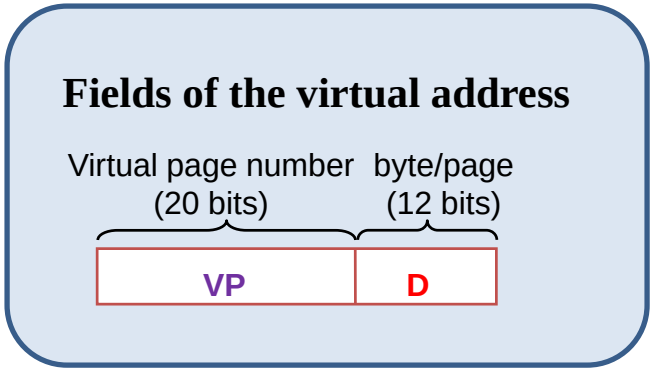
There are 16 page frames in memory: 0x0 to 0xF



Page size: 4 KB = 2^{12} bytes

Example: 3) Page table

Assume a computer with 32-bit addresses, paging, 64KB of memory, 4KB pages and LRU replacement.



Number of pages = $2^{32} / 2^{12} = 2^{20}$

4 bytes per entry

Page table size = Number of entries x Size = $2^{20} \times 2^{2}$ = 4MB (1024 pages)

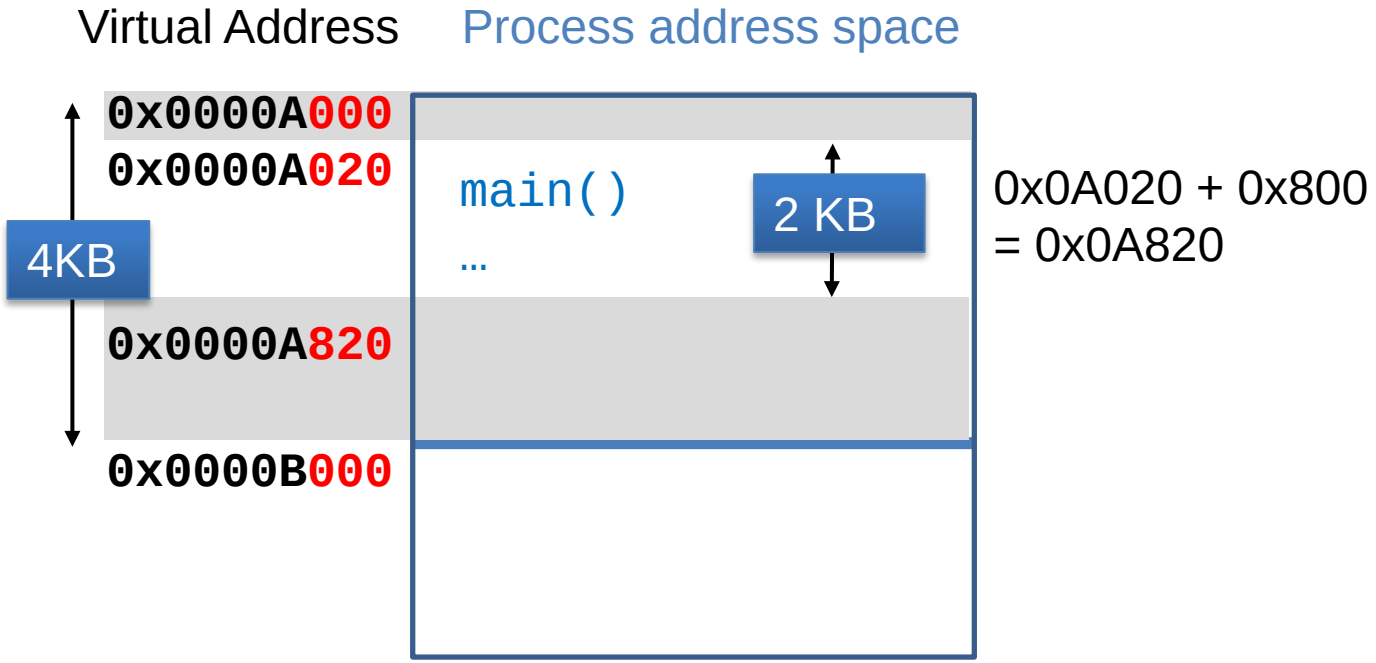
Example: 4) Address space

Code starts in address 0x0A020 and its size is 2 KB (0x800 B). Every time the code is executed it needs to read 8 images, 2 KB each, stored in the following addresses: 22000, 25000, 28000, 32000, 1B000, 2B900, 1C000, 0E800.

	Virtual address	Size	Virtual Pages
Code	0x0000A020	0x0800	0x0000A
Image 1	0x00022000	0x0800	0x00022
Image 2	0x00025000	0x0800	0x00025
Image 3	0x00028000	0x0800	0x00028
Image 4	0x00032000	0x0800	0x00032
Image 5	0x0001B000	0x0800	0x0001B
Image 6	0x0002B900	0x0800	0x0002B, 0x0002C
Image 7	0x0001C000	0x0800	0x0001C
Image 8	0x0000E800	0x0800	0x0000E

Example: 4) Address space (2)

Code starts in address 0x0A020 and its size is 2KB.
Pages: 4KB.



Grey areas are not part of the program code but they are stored in the same page, together with the code.

Example: 4) Address space (3)

Image 6 starts in address 0x2B900 and its size is 2KB.
Pages: 4KB.

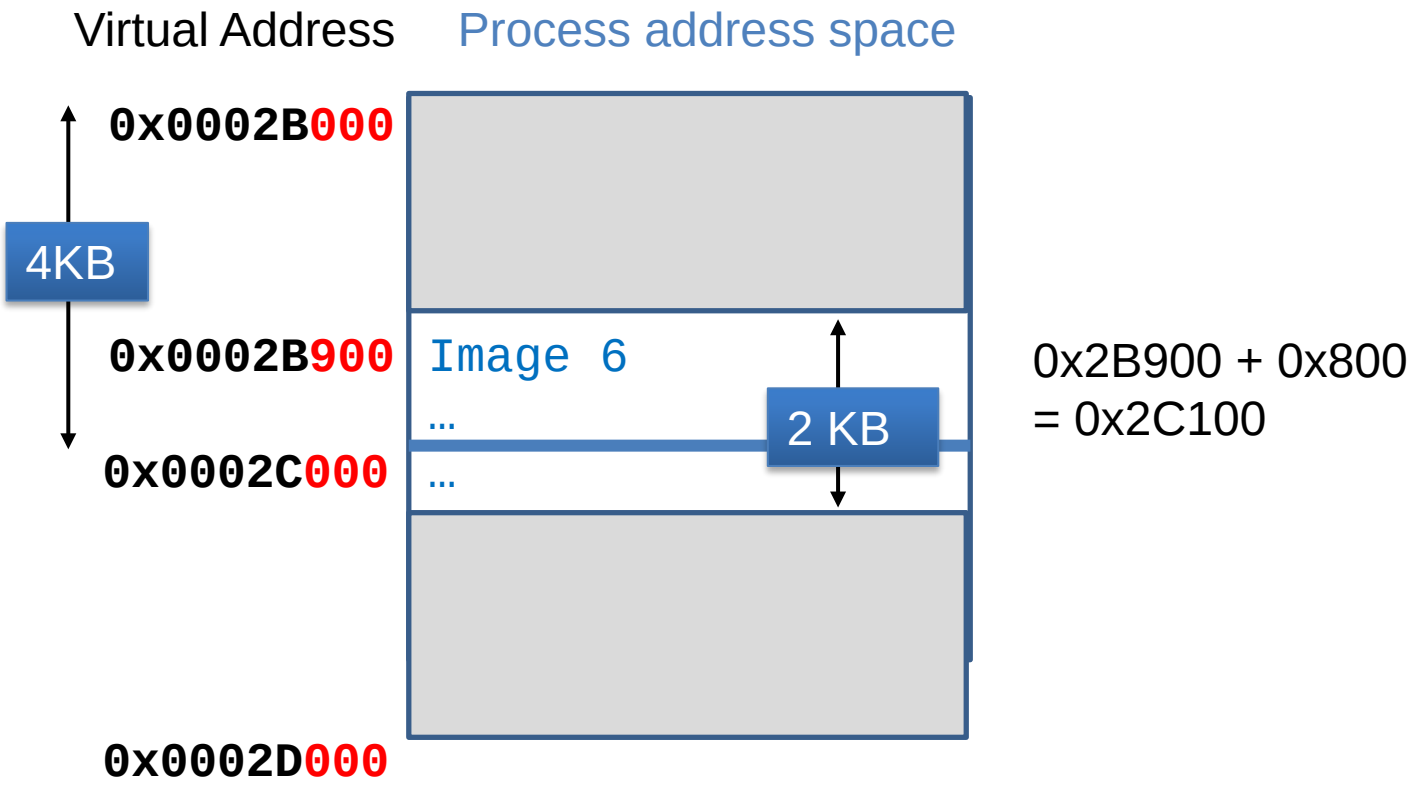


Image 6 is stored in addresses 0x2B900 to 0x2C0FF, so it takes up pages 0x2B and 0x2C

Example: 5) Execution of the code

Initial content: empty

1. Read virtual page 0x0000A

Page fault (V=0)

V. P. number	Page table V	Ref.	P. P. number
0x0000A	1	1	0x5 (*)

Sequence of accesses:

0x0000A (code) ⟵

0x00022

0x00025

0x00028

0x00032

0x0001B

0x0002B, 0x0002C

0x0001C

0x0000E

* Any page frame (0-F) would be valid

Since there are 16 page frames in MM and only 10 virtual pages there will be no replacement.

Example: 5) Execution of the code (2)

2. Read virtual page 0x00022

   Page fault (V=0)

3. Read virtual page 0x0000A

V. P. number	Page table V	Ref.	P. P. number
0x0000A	1	3	0x5
0x00022	**1**	**2**	**0x7 (*)**

Sequence of accesses:

0x0000A (code)

0x00022

0x00025

0x00028

0x00032

0x0001B

0x0002B, 0x0002C

0x0001C

0x0000E

Code is always the most recently accesses page

* Any page frame but 5 would be OK

Example: 5) Execution of the code (3)

V. P. number	Page table V	Ref.	P. P. number
0x0000A	1	17	0x5
0x0000E	1	18	
0x0001B	1	10	
0x0001C	1	16	
0x00022	1	2	0x7
0x00025	1	4	
0x00028	1	6	
0x0002B	1	12	
0x0002C	1	14	
0x00032	1	8	

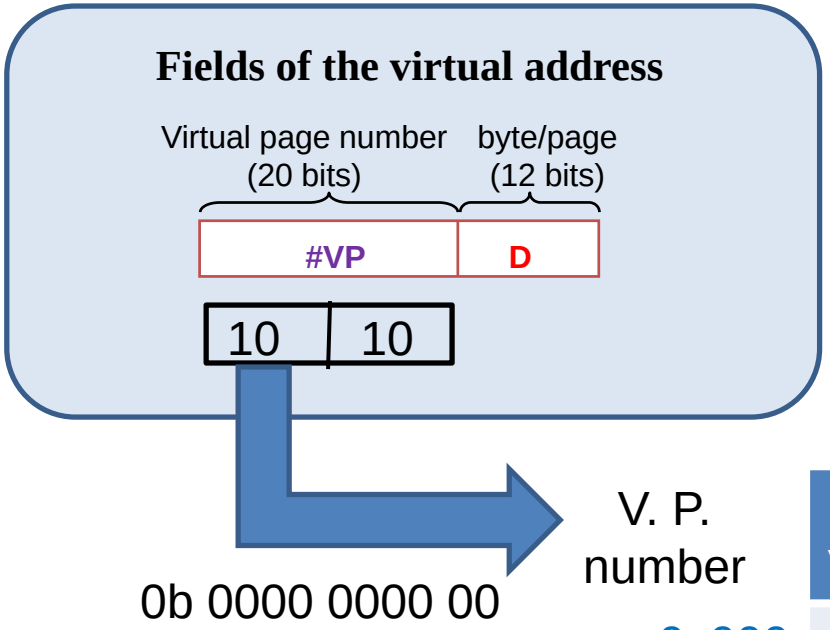
Sequence of accesses:

0x0000A (code)
0x00022
0x00025
0x00028
0x00032
0x0001B
0x0002B, 0x0002C
0x0001C
0x0000E

All other 1.048.566 entries in the page table are empty (V = 0).
The page table takes up 1024 pages but only the first one will be in memory.

Example: 6) 2-level page table

1st level Page table

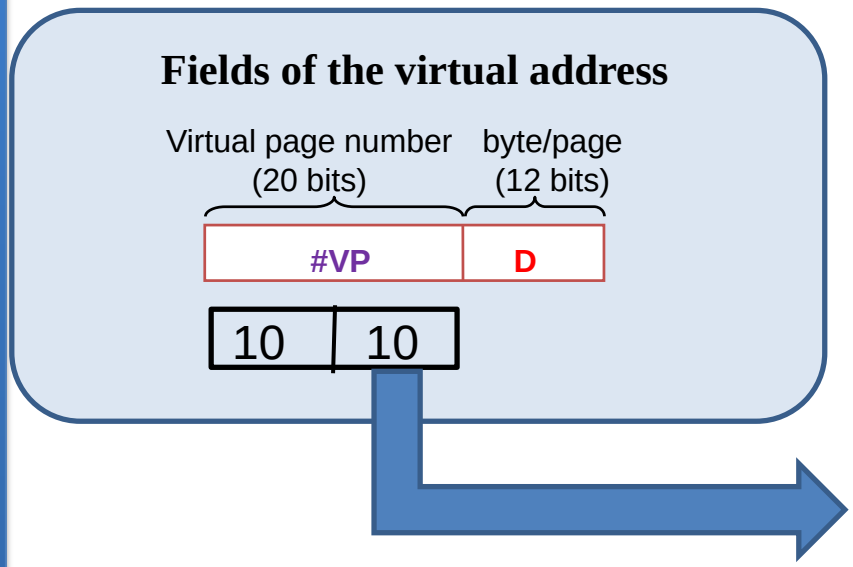


Virtual pages:

0x0000A (code)
0x00022
0x00025
0x00028
0x00032
0x0001B
0x0002B, 0x0002C
0x0001C
0x0000E

V. P. number	1st level Page table V	P. Address
0x000	1	0xE
0x001	0	
…	…	
0x3FF	0	

Example: 6) 2-level page table



2nd level Page table

V. P. number	2nd level Page table V	Ref.	P. P. number
0x00A	1	17	0x5
0x00E	1	18	
0x01B	1	10	
0x01C	1	16	
0x022	1	2	0x7
0x025	1	4	
0x028	1	6	
0x02B	1	12	
0x02C	1	14	
0x032	1	8	

All other 1014 entries in the page table are empty (V = 0).

Example: 6) 2-level page table

V. P. number	1st level Page table V	P. Address
0x000	1	0xE
0x001	0	
…	…	
0x3FF	0	

V. P. number	2nd level Page table V	Ref.	P. P. number
0x00A	1	17	0x5
0x00E	1	18	
0x01B	1	10	
0x01C	1	16	
0x022	1	2	0x7
0x025	1	4	
0x028	1	6	
0x02B	1	12	
0x02C	1	14	
0x032	1	8	

The page table only takes up 2 pages. Both are in memory, so 2 page frames are needed to store the page tables (0xE and 0xF in the example)

Example: 7) TLB

Assume a TLB of 4 entries, fully associative with LRU replacement.

Virtual Page number (TAG)	V D R/W Ref. Physical Page number
	0
	0
	0
	0

V (valid)
D (dirty)
R/W,… (permissions)
Ref. (LRU bits). For TLB replacement

Example: 7) TLB

Sequence of accesses:

0x0000A (code) ⟸

0x00022 — 1. TLB miss

0x00025 — 2. Page fault

0x00028

0x00032

0x0001B

0x0002B, 0x0002C

0x0001C

0x0000E

V. P. number

Page table		
V	Ref.	P. P. number
0		
0		
0		
0		
0		
0		
0		

Virtual Page n.	V D R/W Ref. Physical Page n.
	0
	0
	0
	0

Example: 7) TLB

Sequence of accesses:

0x0000A (code) ⟸

0x00022 — 1. TLB miss

0x00025 — 2. Page fault

0x00028

0x00032

0x0001B

0x0002B, 0x0002C

0x0001C

0x0000E

V. P. number	Page table V	Ref.	P. P. number
	0		
0x0000A	1	1	0x5 (*)
	0		
	0		
	0		
	0		
	0		

Virtual Page n.	V	D	R/W	Ref.	Physical Page n.
0x0000A	1	0	0	1	0x5
	0				
	0				
	0				

Example: 7) TLB

Sequence of accesses:

0x0000A (code)

0x00022 ⟵

0x00025

0x00028

0x00032

0x0001B

0x0002B, 0x0002C

0x0001C

0x0000E

1. TLB miss
2. Page fault

V. P. number	Page table V	Ref.	P. P. number
	0		
0x0000A	1	1	0x5
	0		
0x00022	1	2	0x7
	0		
	0		
	0		

Virtual Page n.	V	D	R/W	Ref.	Physical Page n.
0x0000A	1	0	0	1	0x5
0x00022	1	0	1	2	0x7
	0				
	0				

Example: 7) TLB

Sequence of accesses:

0x0000A (code) ⟸ TLB HIT

0x00022

0x00025

0x00028

0x00032

0x0001B

0x0002B, 0x0002C

0x0001C

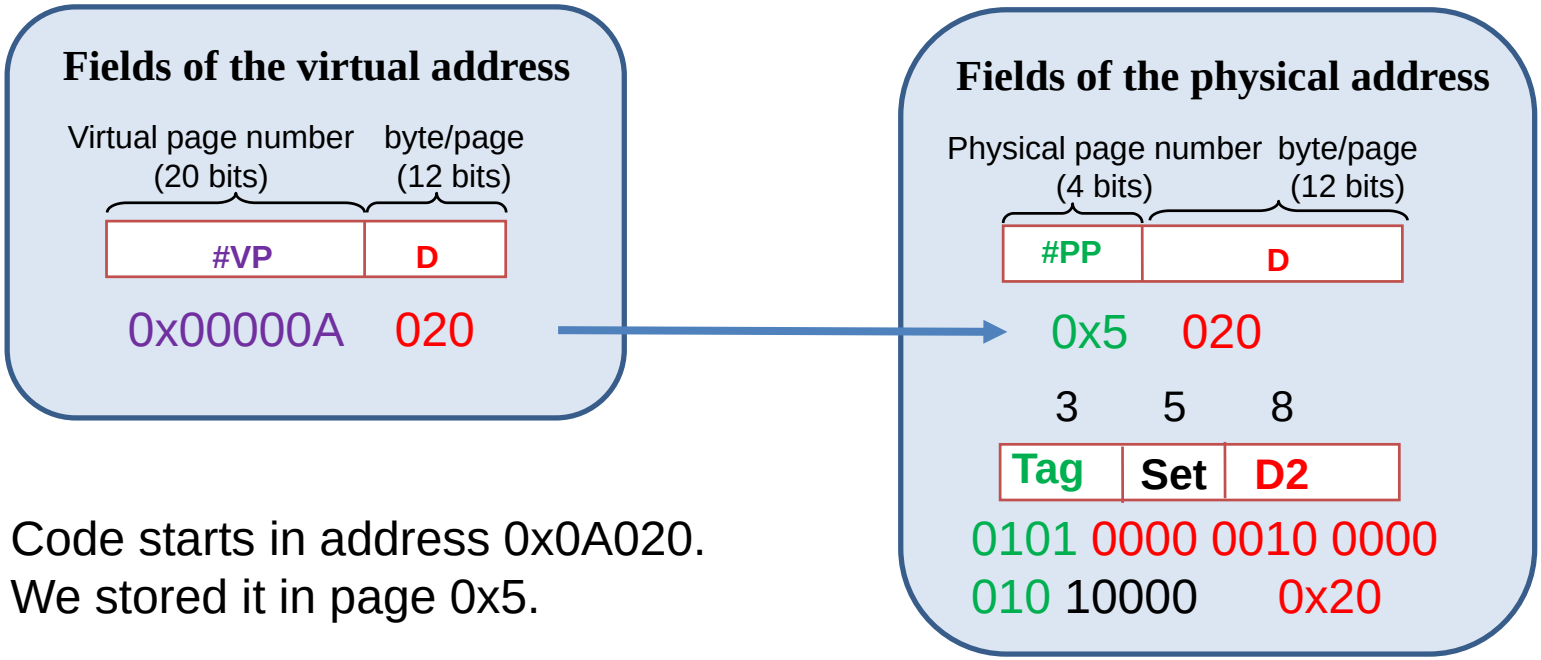
0x0000E

V. P. number	Page table V	Ref.	P. P. number
0x0000A	1	1	0x5
0x00022	1	2	0x7

Virtual Page n.	V	D	R/W	Ref.	Physical Page n.
0x0000A	1	0	0	3	0x5
0x00022	1	0	1	2	0x7
	0				
	0				

Example: 8) Physical cache

Assume a direct mapped 8 KB cache memory with line of 256B



Code starts in address 0x0A020.
We stored it in page 0x5.

If we had chosen physical page 0x4, then it will be in set 0, instead of set 16.

$$\frac{2^{13}\ bytes}{256\ bytes/line} = 32\ lines$$