238P Operating Systems, Spring 2019

Topics : Paging and Stacks

13 December 2019

Aftab Hussain
University of California, Irvine

Review of Address Translation using Paging

```
mov (%EBX), EAX   # mov value from the  location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809
```

20 983 809 = 

> With paging enabled, xv6 can deal with 32 bit virtual addresses

> Size of the virtual address space is 2^32 bytes = 4GB

> A page is of size 4KB

> There are ~1 million pages in the virtual address space.

> The pages are mapped to a physical address space, limited by the physical memory.

> We shall take the help of two tables (2-level tree) to map from virtual to physical.

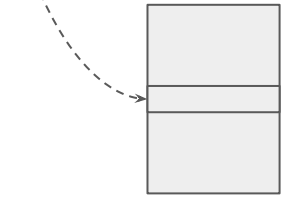
> These tables are stored in the physical memory.

Fun note - bits and bytes

```
mov (%EBX), EAX   # mov value from the  location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809
```

20 983 809 = 

points to entry in the 1st table (the PD)

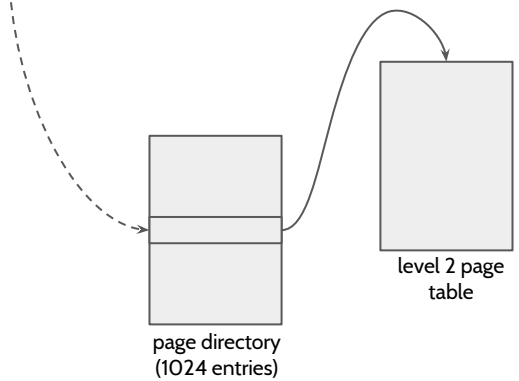


page directory (1024 entries)

```
mov (%EBX), EAX   # mov value from the  location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809
```

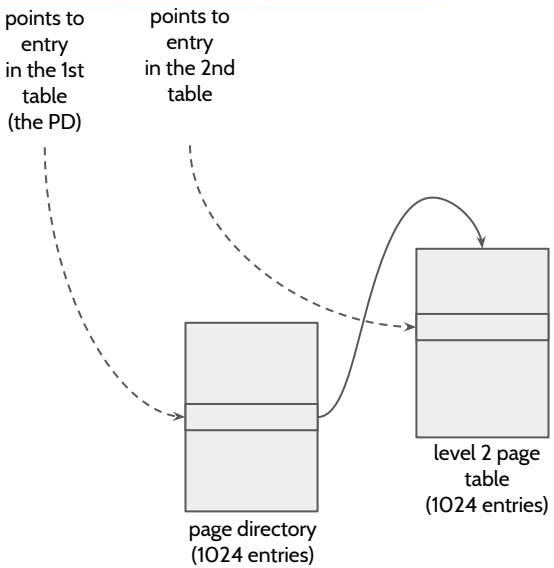
20 983 809 = | 00 0000 0101 | 00 0000 0011 | 0000 0000 0001 |

points to entry in the 1st table (the PD)



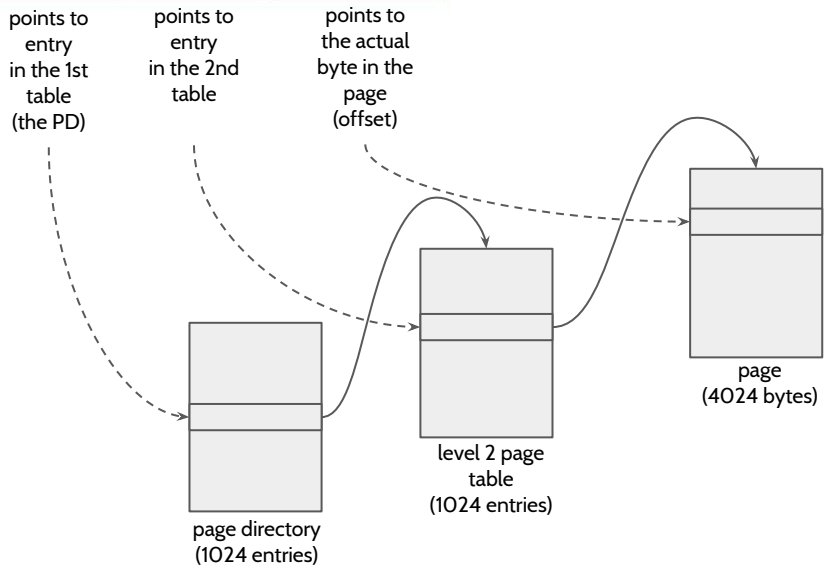
```
mov (%EBX), EAX   # mov value from the  location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809
```

20 983 809 = 

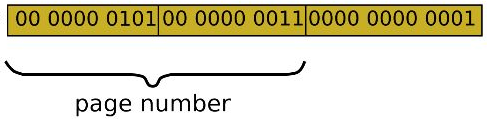


```
mov (%EBX), EAX   # mov value from the  location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809
```

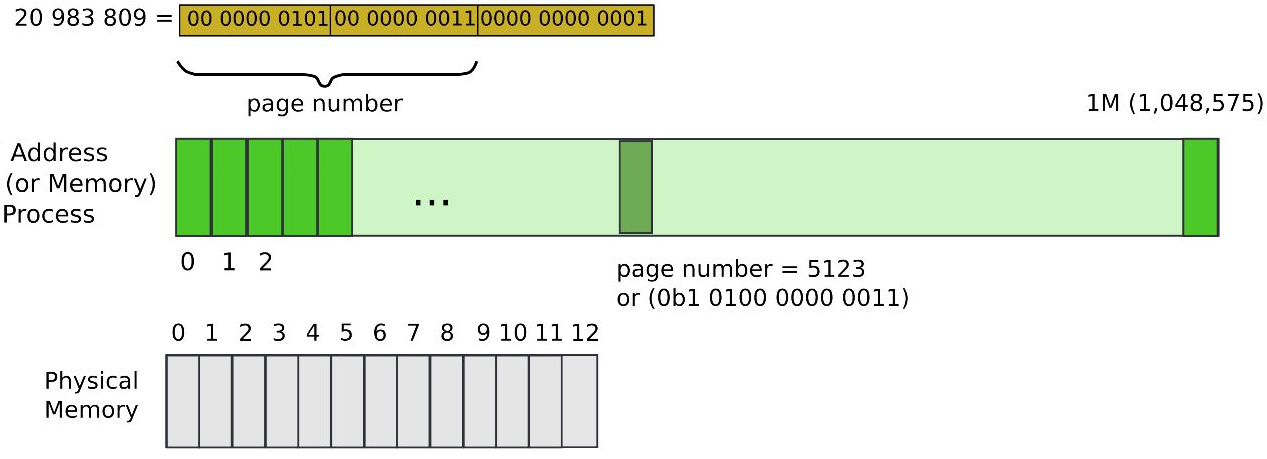
20 983 809 = 



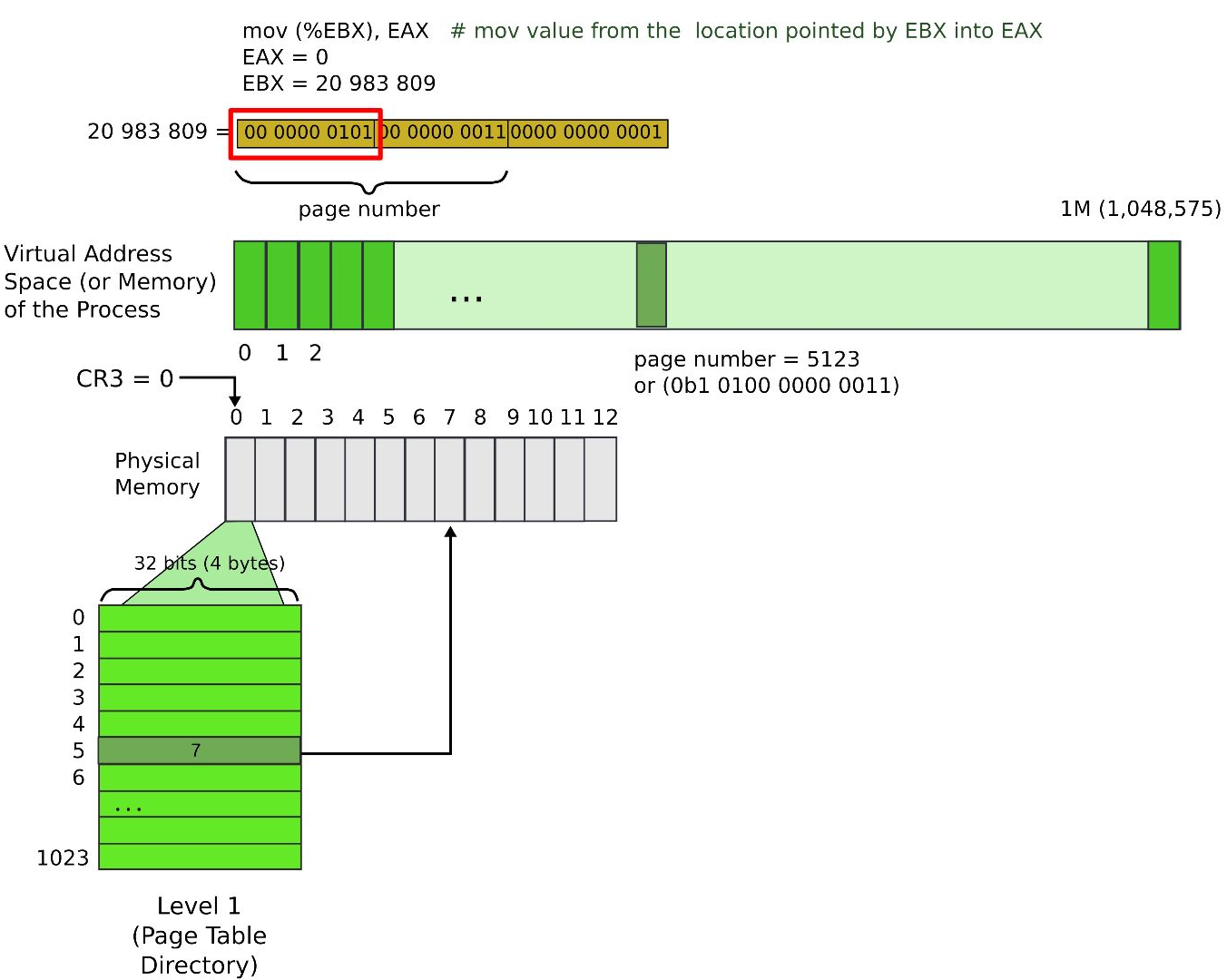
```
mov (%EBX), EAX   # mov value from the  location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809
```

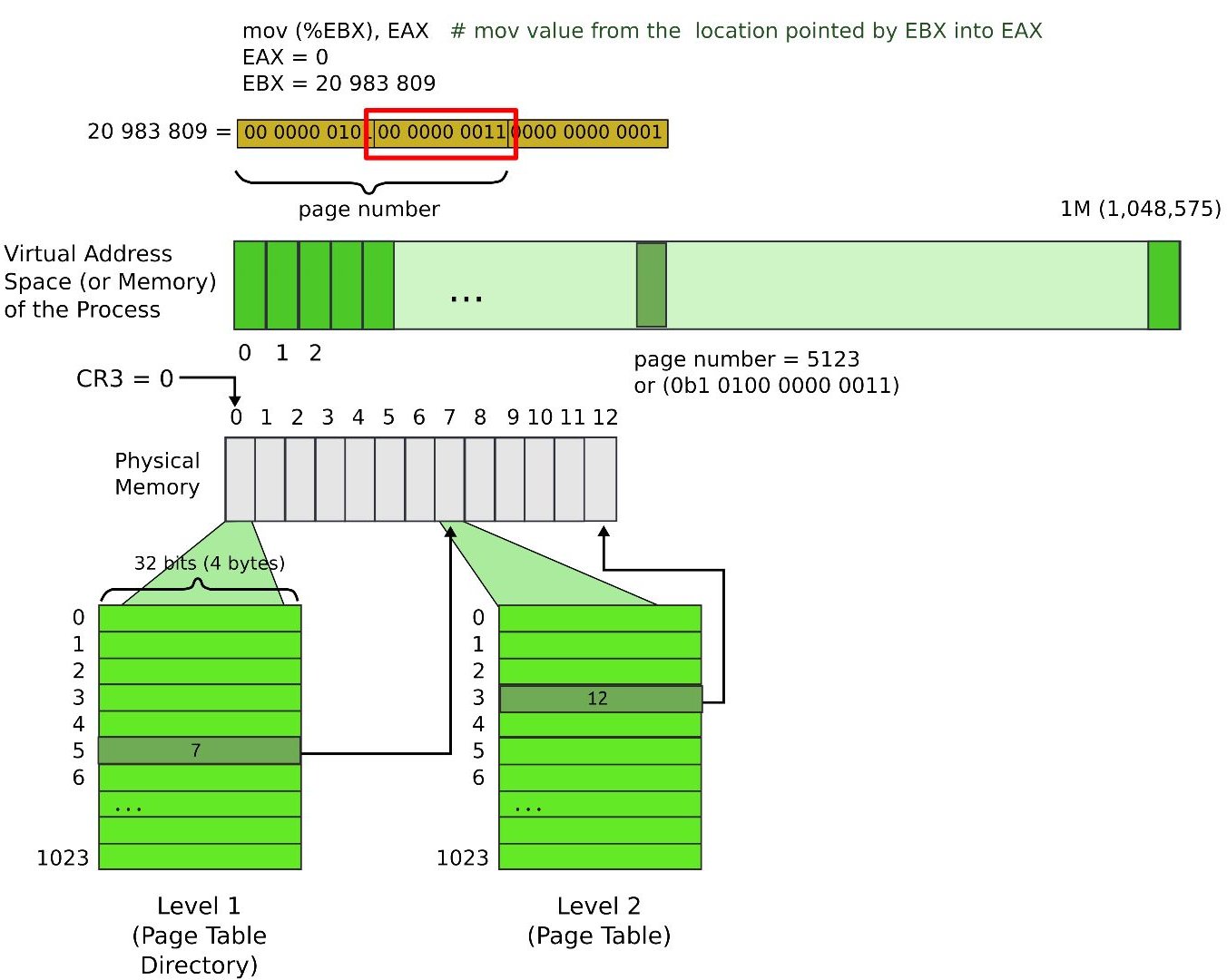
20 983 809 = 

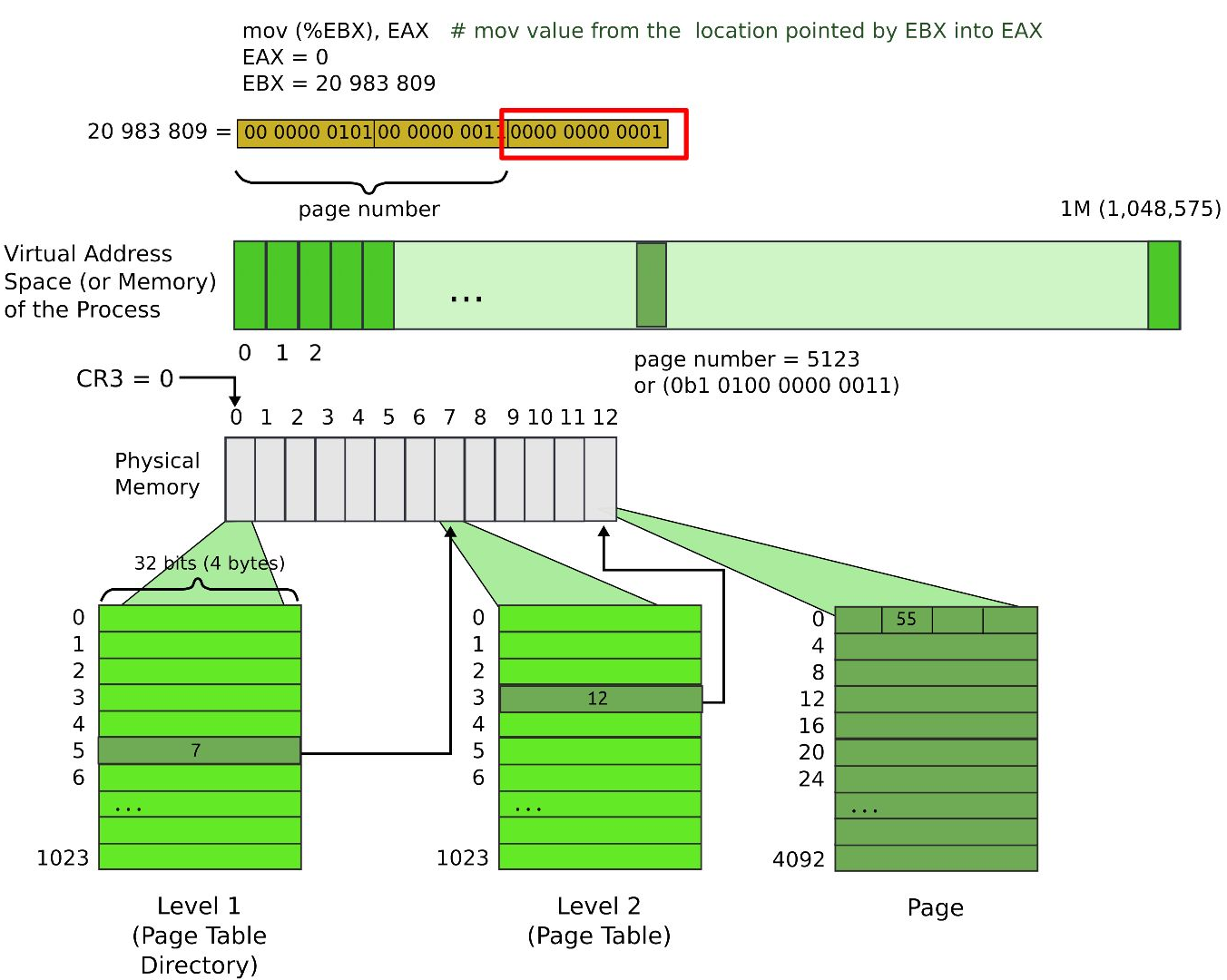
mov (%EBX), EAX # mov value from the location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809

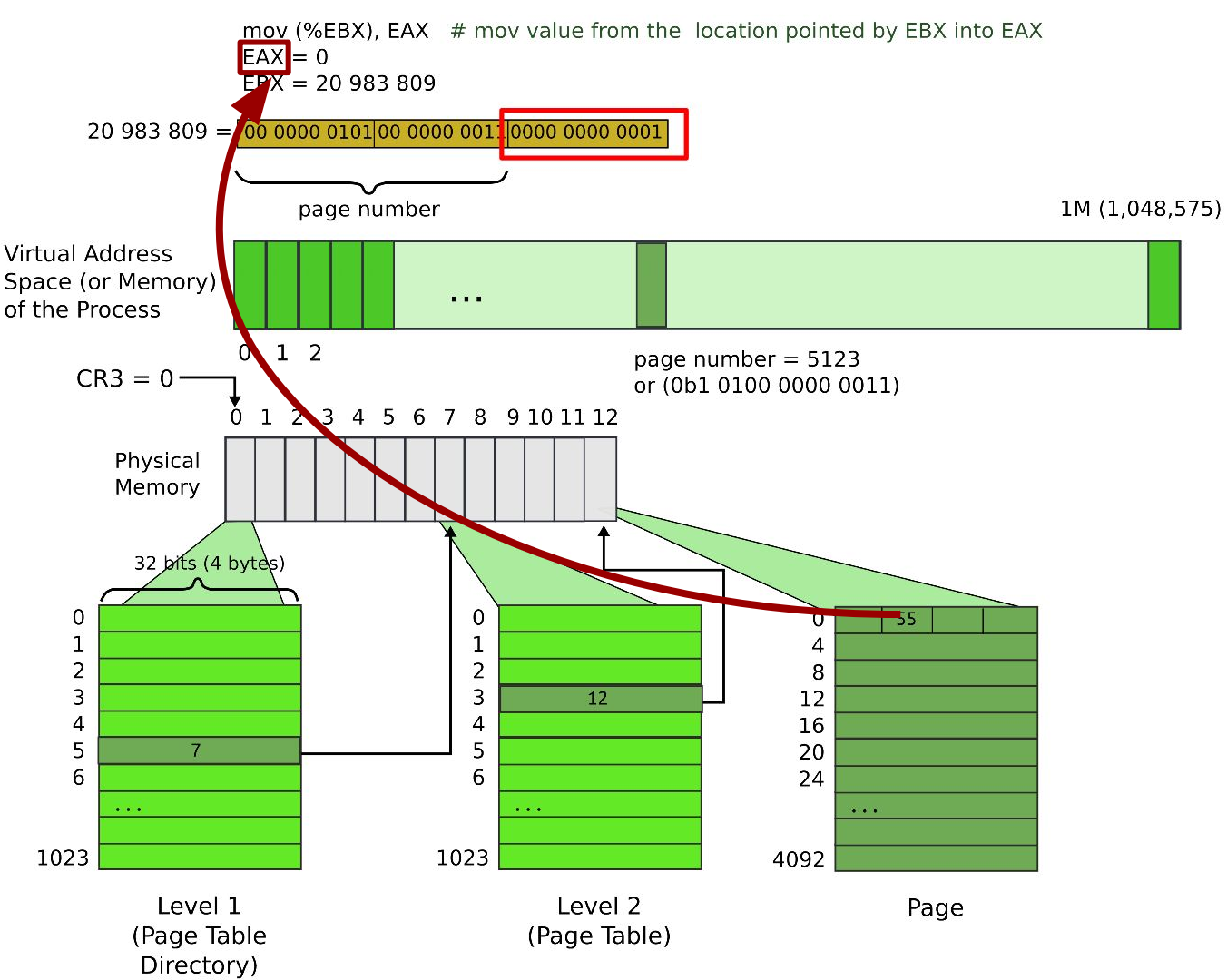


Ref. Anton Burtsev, Fall 17, CS 143A Lecture


mov (%EBX), EAX # mov value from the location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809
20 983 809 =
00 0000 0101
00 0000 0011
0000 0000 0001
page number
1M (1,048,575)
Virtual Address Space (or Memory) of the Process
0 1 2
page number = 5123
or (0b1 0100 0000 0011)
CR3 = 0
0 1 2 3 4 5 6 7 8 9 10 11 12
Physical Memory
32 bits (4 bytes)
0
1
2
3
4
5
6
7
1023
Level 1 (Page Table Directory)


mov (%EBX), EAX # mov value from the location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809
20 983 809 = 00 0000 010 00 0000 0011 0000 0000 0001
page number
1M (1,048,575)
Virtual Address Space (or Memory) of the Process
0 1 2
page number = 5123
or (0b1 0100 0000 0011)
CR3 = 0
0 1 2 3 4 5 6 7 8 9 10 11 12
Physical Memory
32 bits (4 bytes)
0 1 2 3 4 5 6 7 ... 1023
0 1 2 3 4 5 6 12 ... 1023
Level 1 (Page Table Directory)
Level 2 (Page Table)


mov (%EBX), EAX # mov value from the location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809
20 983 809 =
00 0000 0101
00 0000 0011
0000 0000 0001
page number
1M (1,048,575)
Virtual Address Space (or Memory) of the Process
0 1 2
page number = 5123
or (0b1 0100 0000 0011)
CR3 = 0
0 1 2 3 4 5 6 7 8 9 10 11 12
Physical Memory
32 bits (4 bytes)
0
1
2
3
4
5
6
1023
7
12
55
0
4
8
12
16
20
24
4092
Level 1 (Page Table Directory)
Level 2 (Page Table)
Page


mov (%EBX), EAX # mov value from the location pointed by EBX into EAX
EAX = 0
EBX = 20 983 809
20 983 809 = 0000 0000 0101 00 0000 0011 0000 0000 0001
page number
1M (1,048,575)
Virtual Address Space (or Memory) of the Process
0 1 2
page number = 5123
or (0b1 0100 0000 0011)
CR3 = 0
0 1 2 3 4 5 6 7 8 9 10 11 12
Physical Memory
32 bits (4 bytes)
0 1 2 3 4 5 6 7 ... 1023
0 1 2 3 4 5 6 12 ... 1023
0 4 8 12 16 20 24 55 ... 4092
Level 1 (Page Table Directory)
Level 2 (Page Table)
Page

> Problem 2, from CS238P Fall 2018, Midterm
> Problem 1, from CS238P Winter 2018, Midterm

Stack

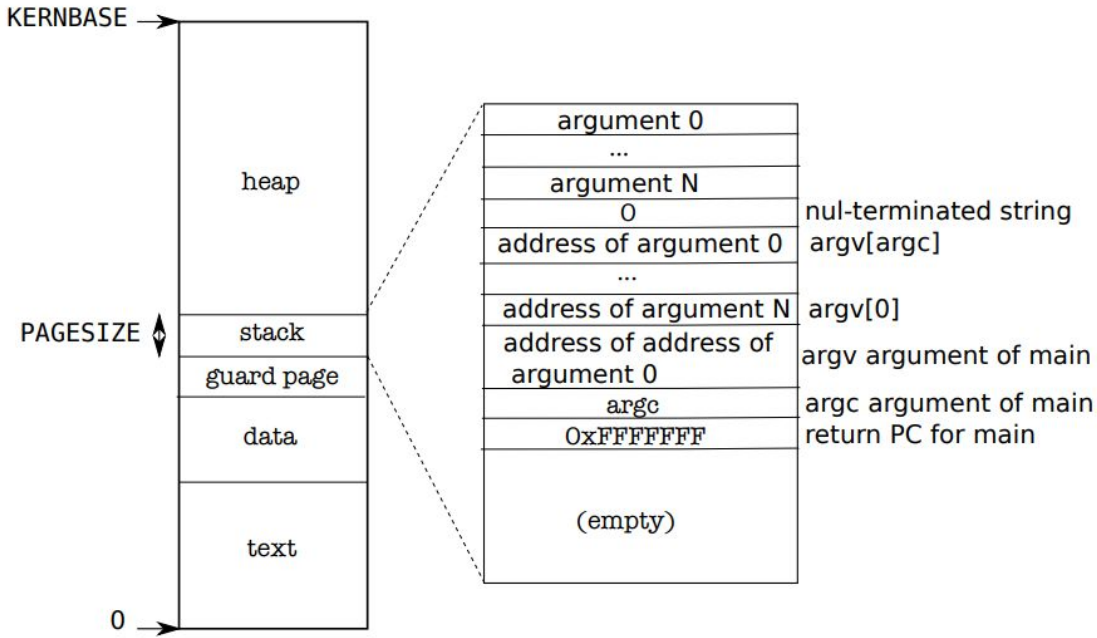


Figure 2-3. Memory layout of a user process with its initial stack.

> The stack is populated from the higher to the lower addresses.
> Arguments of function calls occupy the higher addresses of the stack.
> The guard page is unmapped, and is aimed to prevent the stack from growing beyond the size of the stack (which is 1 page.)

xv6 book, https://pdox.csail.mit.edu/6.828/xv6

> Problems 2 and 3 from CS238P Winter 2018, Midterm
> Problem 3, from CS238P Fall 2018, Midterm