Ng Chi Hon 1155116317

CSCI-3150: Introduction to Operating Systems

Assignment Three: Virtual Memory

Deadline: Dec. 6, 2020 23:59

Suppose the page size is 8 bytes, the first-level page table has 4 entries and the second-level page table has 8 entries. You need to answer the following questions:

1 Calculation

1. What is the size of the virtual address space?
2. How many bits does a virtual address have?
3. How many bits should be reserved for the first-level page table index, the second-level page table index and the offset respectively?

2 Address Translation

For the virtual address space defined by the two-level page tables as shown in Figure 1, given the following virtual addresses: **7, 68, 140, 231**.

1. Which virtual addresses are mapped?
2. If a virtual address is mapped, what is its corresponding physical address?

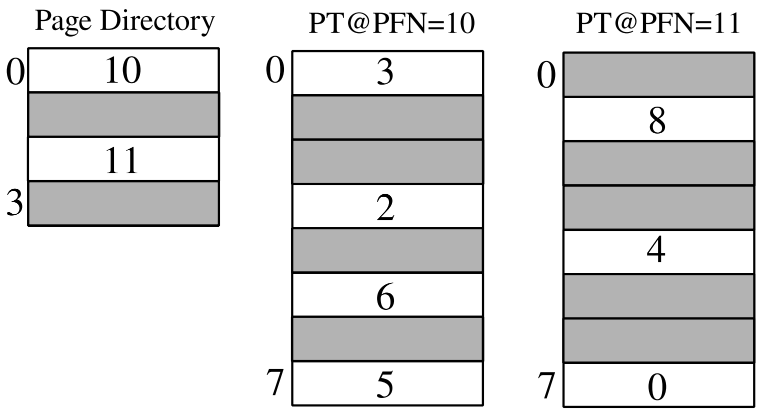


Figure 1: The two-level page mapping table. The numbers in entries are physical page frame numbers. Entries in gray color are not mapped.

3 Page Mapping Setup

A program has three segments that should be loaded and mapped:

1. **Code segment** is loaded at physical memory range [0, 24) and should be mapped to virtual memory range [8, 32)
2. **Data segment**: PA [24, 48) and should be mapped to VA [40, 64)
3. **Stack segment**: PA [48, 80) and should be mapped to VA [224, 256)

You should fill the **physical page frame numbers** in the following page tables in Figure 2 to set up the page mapping. If some entries in the page tables are not used, you can just leave them empty.

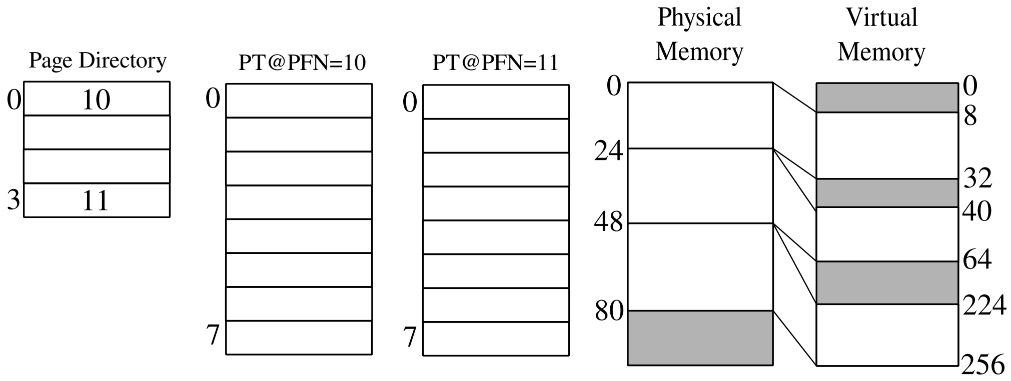
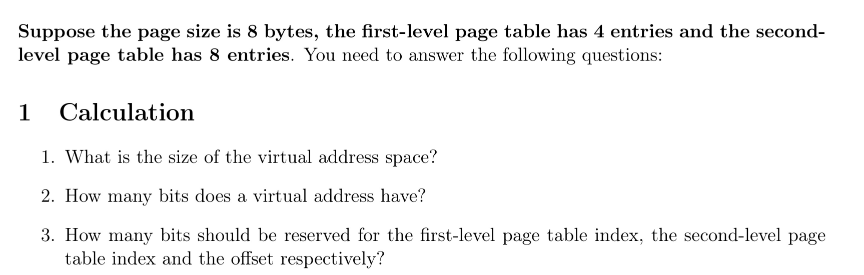


Figure 2: Page tables and address mapping.

Note: the page directory has been set up. You only need to attach the filled **PT@PFN=10** and **PT@PFN=11** in your answer.

Submission

You only need to submit a PDF that contains your answer to Blackboard. If you have any questions about this assignemnt, please send a email to jinxue@cse.cuhk.edu.hk.



Suppose the page size is 8 bytes, the first-level page table has 4 entries and the second-level page table has 8 entries. You need to answer the following questions:

1 Calculation

1. What is the size of the virtual address space?

2. How many bits does a virtual address have?

3. How many bits should be reserved for the first-level page table index, the second-level page table index and the offset respectively?

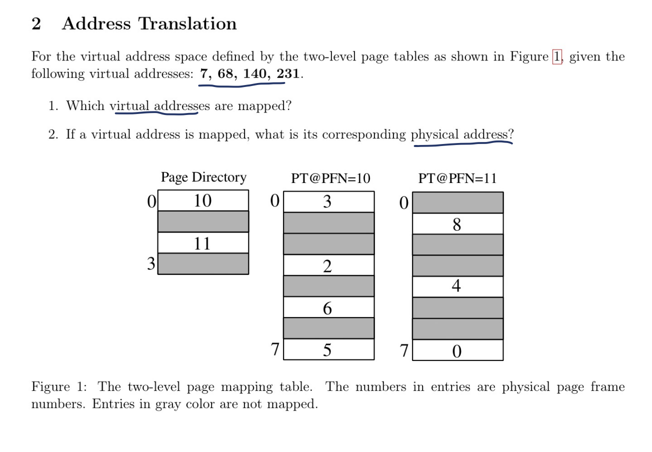
1, In 2nd level table, there has 8 entries, (page size is 8 byte)
one 2nd level page table map to $8 \cdot 8 = 64$ byte
In 1st level page table has 4 entries,
one 1st level page table map to $4 \cdot 64 = 256$ byte
Since 1st level page is top level, the virual address space has 256 byte

2, Virtual address space is 256 byte
There are 256 VA $= 2^8$ VA
$\therefore$ We need 8 bit for VA

3, a page has 8 bytes (8 unique offset)
$8 = 2^3$ $\therefore$ Need 3 bits to locate an offset inside page

2nd level page table has 8 entries,
$8 = 2^3$ $\therefore$ Need 3 bits to locate entry inside 2nd level page table

1st level page table has 4 entries
$4 = 2^2$ $\therefore$ Need 2 bits to locate entry inside 1st level page table



2 Address Translation

For the virtual address space defined by the two-level page tables as shown in Figure 1, given the following virtual addresses: **7, 68, 140, 231**.

1. Which virtual addresses are mapped?
2. If a virtual address is mapped, what is its corresponding physical address?

	Page Directory
0	10
	11
3	

	PT@PFN=10
0	3
	2
	6
7	5

	PT@PFN=11
0	
	8
	4
7	0

Figure 1: The two-level page mapping table. The numbers in entries are physical page frame numbers. Entries in gray color are not mapped.

2)(1) Page Size is 8 byte

PT has 8 entries, PT map 8·8 = 64 byte

For VA = 7

VPN1 = 7/64 = 0 which mean page director entry 0,
∴ VA = 7 is mapped to PT @PFN=10 //

The offset within PT is 7% 64 = 7
∴ VA = 7 is mapped to PT @PFN=10 & the corresponding Physical Address for the byte at offset 7 inside PT
VPN2 = floor(7/8) = 0, so it is first page in PT (PFN = 3)

∴ VA = 7 is mapped to PT [0) in PT @ PFN=10

For VA = 68
VPN1= 68/64 =1
entry 1 is not mapped in Page Directory, Thus VA=140 is not mapped

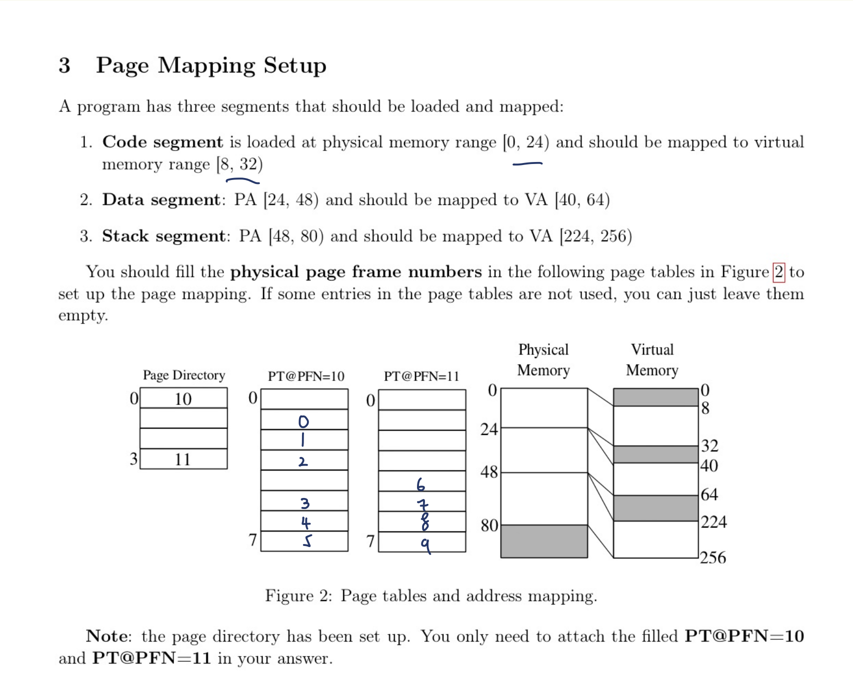
For VA = 140
VPN1 = 140/64=2 which mean page directory entry 2
∴ VA =140 is mapped to PT @PFN=11

The offset within PT is 140%64 =12
VPN2 = floor(12/8) = 1, entry 1 of PT@PFN=11 is 2nd page (PFN = 8)
∴ VA= 140 is mapped in PT[1] in PT@PFN=11

For VA = 231
VPN1= 231 / 64 =3
Entry 3 is not mapped in Page Directory, Thus VA = 231 is not mapped

(2) For VA = 7
The offset inside page is 7%8 = 7. So the PA of VA = 7 is 3·8+7 = 31 //

For VA =140
The offset inside Page is 12%8 = 4 ∴ PA of VA =140 = 8·8 +4 = 68 //



3 Page Mapping Setup

A program has three segments that should be loaded and mapped:

1. **Code segment** is loaded at physical memory range [0, 24) and should be mapped to virtual memory range [8, 32)
2. **Data segment**: PA [24, 48) and should be mapped to VA [40, 64)
3. **Stack segment**: PA [48, 80) and should be mapped to VA [224, 256)

You should fill the **physical page frame numbers** in the following page tables in Figure 2 to set up the page mapping. If some entries in the page tables are not used, you can just leave them empty.

Figure 2: Page tables and address mapping.

Note: the page directory has been set up. You only need to attach the filled **PT@PFN=10** and **PT@PFN=11** in your answer.

For VA [8, 32), there are 3 page [(32-8)/8 = 3],

First page (PFN=0), VPN1 = floor (8/64 = 0), offset is 8%64 = 8
VPN2 = floor (8/8) = 1

∴ First page of Code segment is in PT[1] in PT @ PFN = 10

And eanse 2nd & 3 rd page of code segment is in PT[2] & PT[3] in PT @ PFN = 10

For VA = [40, 64), there are 3 page,

First page (PFN = 3), VPN1 = floor (40/64) = 0, offset is 40%64 = 40
VPN2 = floor (40/8) = 5

∴ page of Data segment (PFN = 3) is in PT[5] in PT @ PFN = 10

By Calculation,

PFN = 4 & PFN = 5 is in PT[6] & PT[7] in PT @ PFN = 10

For VA = [224, 256) there are 4 page

First page [PFN = 6), VPN1 = floor (224/64) = 3 offset is 224%64 = 32
VPN2 = floor (32/8) = 4

∴ Page of Stack segment (PFN=6) is in PT[4] in PT @ PTN = 11

& By calculation, PFN = 7, PFN = 8 & PFN = 9 is in PT[5], PT[6], PT[7]
in PT @ PTN = 11