HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY, VNU HCM
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

OPERATING SYSTEM

LAB 9: PAGING

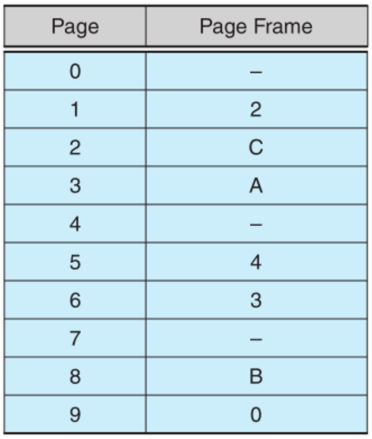
Pham Minh Tuan - MSSV: 1752595

Contents:

1 Exercises

1.1 Question

Consider the page table shown in Figure 2.1 for a system with 12-bit virtual and physical address and with 256-byte page. Assume that the list of free page frames consists of D, E, F (that is, D is at the head of the list, E is second, and F is the last)



Page	Page Frame
0	–
1	2
2	C
3	A
4	–
5	4
6	3
7	–
8	B
9	0

Convert the following virtual address into their equivalent physical address in hexadecimal. All numbers are given in hexadecimal. (A dash for a page frame indicates that the page is not in memory)

- 9EF
- 111
- 700
- 0FF

We have:

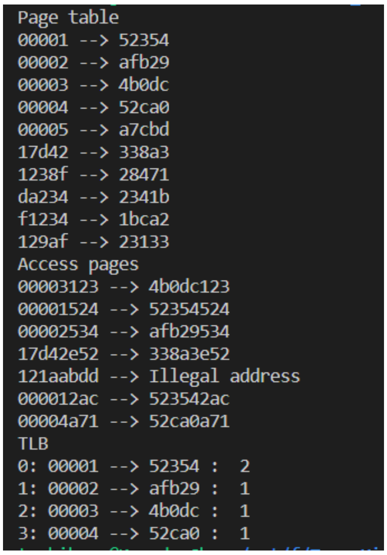
- 12-bit virtual address → Total virtual memory size is 2^{12}.
- Size of a single page is 256-byte → 2^8 → 8-bit is in LSB is page offset
- Total number of page is $\frac{2^{12}}{2^8} = 2^4$ → The remaining 4-bit is page number
- Therefore we only look for the 4-bit in MSB of the virtual address in the table, if the page we are looking for is not in memory, we will allocate it for the free page frames that is D, E, F.

Answer:

- 9EF → 0EF
- 111 → 211
- 700 → D00
- 0FF → EFF

1.2 Programming Exercise

The result:



```
Page table
00001 --> 52354
00002 --> afb29
00003 --> 4b0dc
00004 --> 52ca0
00005 --> a7cbd
17d42 --> 338a3
1238f --> 28471
da234 --> 2341b
f1234 --> 1bca2
129af --> 23133
Access pages
00003123 --> 4b0dc123
00001524 --> 52354524
00002534 --> afb29534
17d42e52 --> 338a3e52
121aabdd --> Illegal address
000012ac --> 523542ac
00004a71 --> 52ca0a71
TLB
0: 00001 --> 52354 :  2
1: 00002 --> afb29 :  1
2: 00003 --> 4b0dc :  1
3: 00004 --> 52ca0 :  1
```