



Operating Systems

(Homework 2)

Designing a Virtual Memory Manager

These lecture materials are modified from the source lecture notes written by A. Silberschatz, P. Galvin and G. Gagne.



Outline

- Objectives
- How to write a program?
- What to do
- Schedule



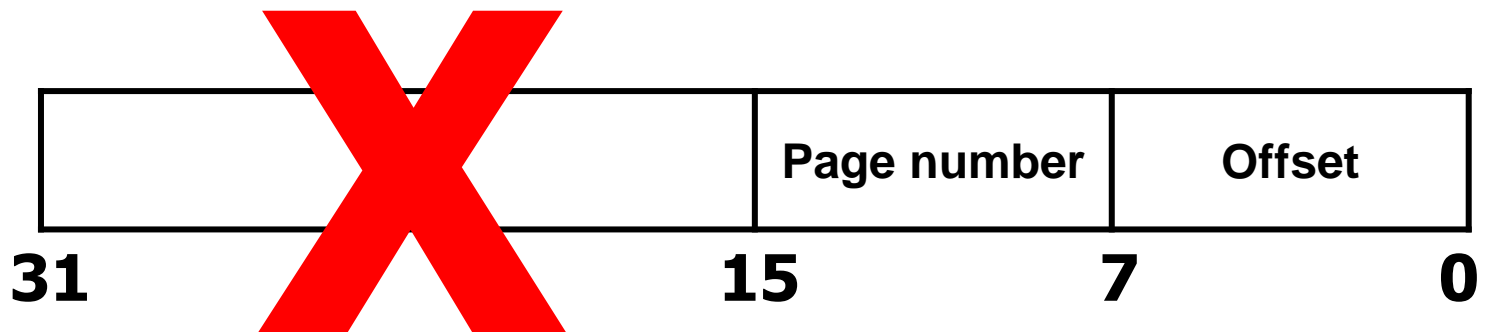
Objectives

- Understanding
 - Address translation
 - Demand paging
 - Page fault handling
 - Page replacement algorithms
 - LRU and LRU-approximation
- Solving
 - Paging simulation

How to write a program?

- Assumption

- 16-bit address space
 - $2^{16} = 65536$ bytes
 - Need to mask the rightmost 16bits of each integer number
 - Divided into 1) 8-bit page number and 2) 8-bit page offset
- Each page number is unique





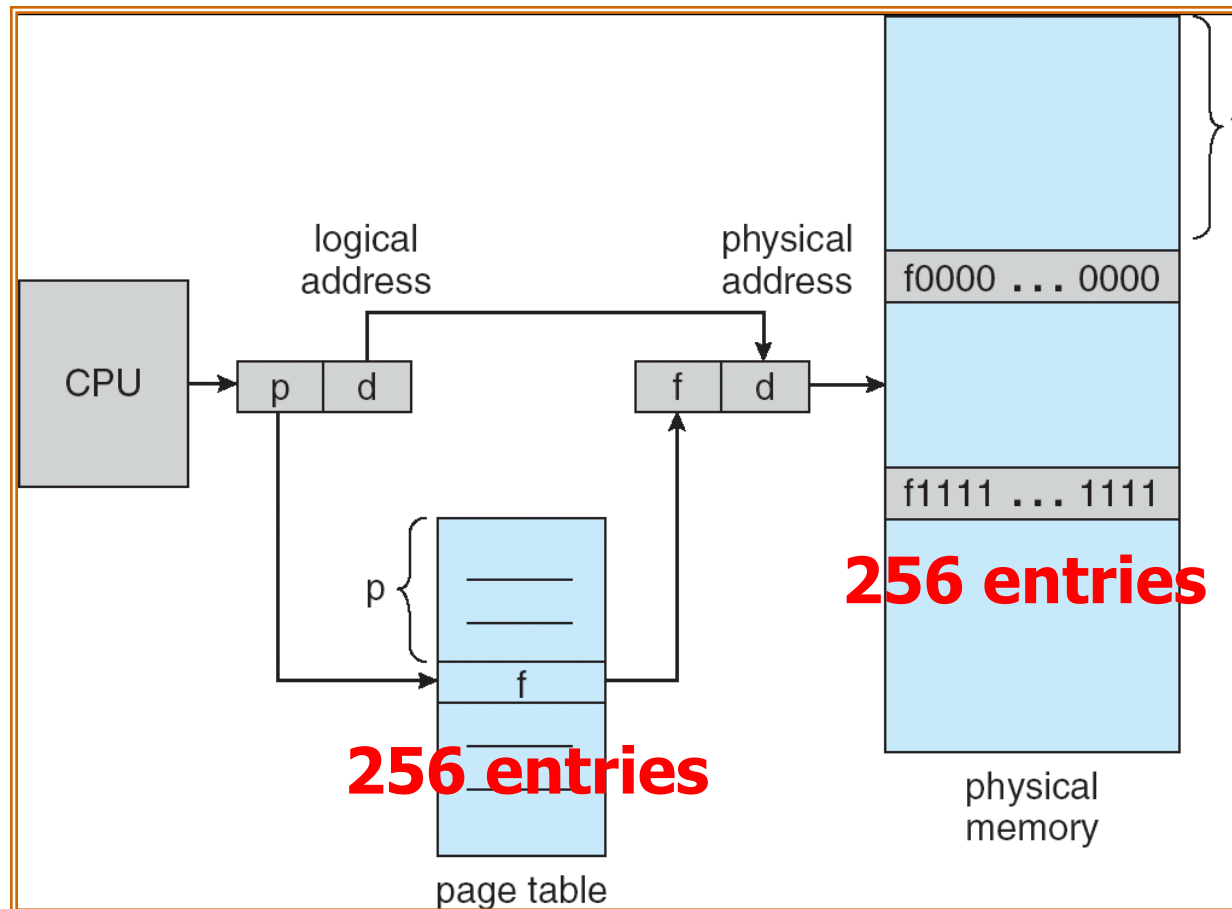
How to write a program?

- Assumption

- 2^8 entries in page table
 - Page number: 8-bit
- Page size is 2^8 bytes
 - Page offset: 8-bit
- 2^5 entries in TLB
 - There must be some page faults
- Frame size is 2^8 bytes
 - Page offset: 8-bit
- 256 frames
 - So, 65536 bytes of frame size
 - 16-bit address space

How to write a program?

- Address translation – from chapter 8, pp. 38





What to do

- Address translation

- From logical address to physical address
- Input file will be given
 - address.txt
 - 1000 logical addresses
- Generate output file
 - physical.txt
 - Contains physical addresses from logical addresses
 - frame_table.txt
 - Contains (frame #, flag, page #)
 - Flag: 0(free), 1(allocated)
 - Ex:
 - 0 1 0
 - 100 0 -1



What to do

- Address translation

- How to translate
- Ex) 4 logical address: 1 256 32768 32769
 - 1: 00000000 00000001 -> page 0, offset 1
 - 256: 00000001 00000000 -> page 1, offset 0
 - 32768: 10000000 00000000 -> page 128, offset 0
 - 32769: 10000000 00000001 -> page 128, offset 1

Page	Frame
0	0
1	1
...	-1 (invalid)
128	2
...	-1 (invalid)

-> physical.txt
1 256 512 513



What to do

- Implement page replacement algorithms
 - LRU and LRU-approximation
 - Compare TLB hit ratio
 - Generate output file
 - Contains (page #, frame #)
 - TLB_LRU.txt
 - TLB_LRUA.txt



What to do

- Summary

- Implement

- Address translation

- From logical addresses to physical addresses

- Page replacement algorithms

- LRU and LRU-approximation algorithms

- Generate 4 output files

- All output files must contain the final results at the end of the simulation
- physical.txt
- frame_table.txt
- TLB_LRU.txt
- TLB_LRUA.txt



What to do

- Requirements

- Report

- *.hwp or *.docx

- Source code

- *.c or *.cpp

- Screenshot

- Running examples must be shown in the images
 - *.jpg or *.png



Schedule

- Deadline
 - June 20th 23:59
 - No submission after the deadline