High Level Design Document
Virtual Memory System Simulation
Operating System Assignment 1

Xu Zhang
SUNY at New Paltz
zhangx5@hawkmail.newpaltz.edu

October 20

1 Project Design Overview

1.1 Project Description

This project is going to simulate the virtual memory management which is based on paging scheme.
The program will read a file of logical addresses fetch pages as needed from a backing store which is simulated by a binary file, and output the value of the byte stored at the corresponding physical addresses and the signed byte value stored at the translated physical address.

1.2 Language

This project is going to be implemented by Java. Because I know Java better than C.

1.3 Design Purposes

I will try to make my solution as general as I can. For example:

- The parameters of each class can be changed easily, such as the entries of the TLB.
- The physical address space can be smaller than the virtual address space.
- The program can be used to test different page-replacement algorithm.
- Elegant style and good comments.

2 Project Solution

2.1 Problem Decomposing and solution

The solution is designed by decomposing the problem into several subtasks. The main subtasks include:

2.1.1 Logical Address Analysis

Read the logical addresses from the input file(addresses.txt) and extract the page number and offset from each address.
This task can be decomposed into three small tasks:

- Read each address from input file:
  Solution: Scanner(File source) class and nextInt() method.
- Convertion between desimal number and binary number:
  Solution: toBinaryString(int i) and parseInt(String s, int radix) methods.
- Extract the address from the integer, get the page number p and offset d:
  Solution: Java operators for bit-masking and bit-shifting.

2.1.2 Basic Address Translation

First consult the TLB table, In the case of a TLB-hit, the frame number is obtained from the TLB. In the case of TLB-miss, the page table must be consulted.
This task can be decomposed into two small tasks:

- Obtain the frame number from TLB or Page table:
  Solution: Write a search method which implemented by simple for-loop for each class.
- TLB replacement policies:(optional)
  Because TLB only have 16 entries. When TLB-miss or page-replacement happens, TLB table need to be updated.
  Solution: Try algorithm such as FIFO or LRU which for page replacement to select the victim entry.

The *Figure 1* is the basic translation process.

2.1.3 Handling Page Faults

When a page fault occurs, the program will read in 256-byte page from the file BACKIG_STORE.bin and store it in an available page frame in physical memory. And update the TLB table and page table.
One big problem of this task is read 256-byte page from file.
Solution: Use DataInputStream class, readInt() method to implement.
The *Figure 2* is the page faults handling process.

2.1.4 Page Replacement

Use FIFO or LRU algorithm to implement.

2.2 Data Structrues

- Page & frame
  use some instance data to represent bytes or fields
- Page Table & physical memory
  array: length fixed, more efficient
- TLB Table
  array or hash table
- backing store
  binary fields or array

2.3 File Listing

2.3.1 Minimum Class Files

There are at least seven class files to implement the project:

- Page.java
- PageTable.java
- TLB.java
- Frame.java
- PhysicalMemory.java
- MemoryManager.java
- Statistic.java

2.3.2 Related Files

- addresses.txt
- BACKING_STORE.bin
- correct.txt

3 Project's Design Evolution

Steps of evolution:

- 1. Write a simple program that extracts the page number and offset.
- 2. Use page table along to do the translation.
  Use addresses.txt to test the output of the programm.
  Keep a record of page fault.
- 3. When page table works properly integrate the TLB.
  Use addresses.txt to test the output of the programm.
  Keep a record of TLB-hit.
- 4. Implement the page fault solution.
- 5. Implement the algorithm of page replacement.

4 Detailed Technical Representation

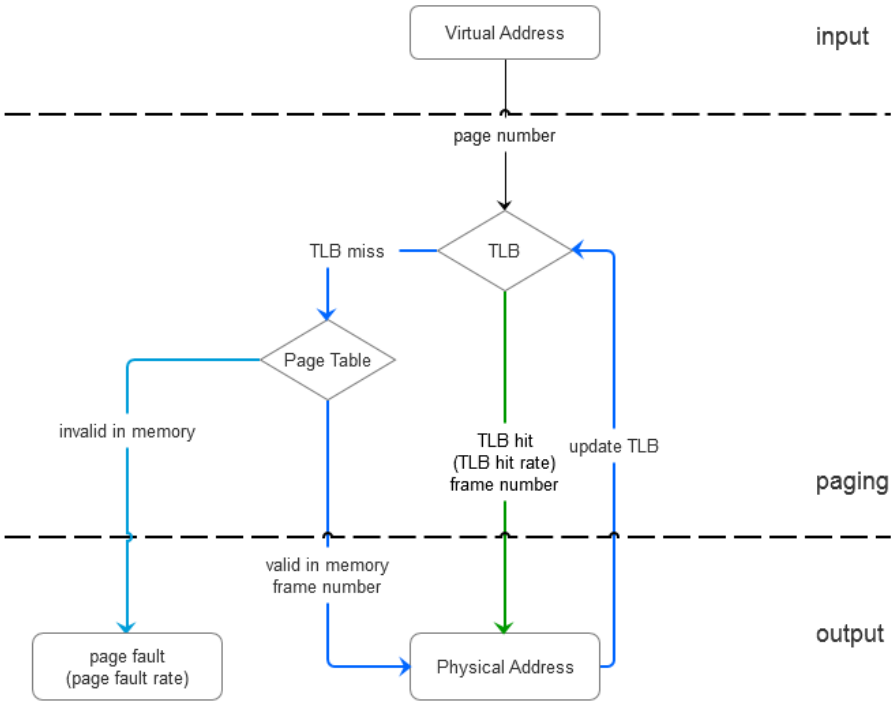

Figure 1: Basic Translation Process

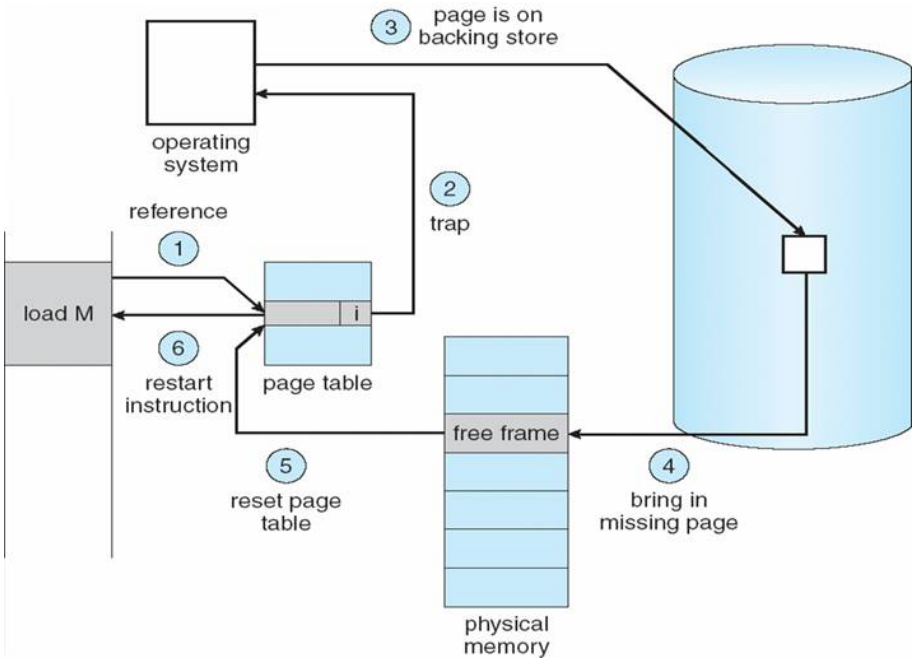

Figure 2: Page Fault handling process(from Textbook)