Virtual memory manager

Ali Mohmaed Mokhtar

20160249
Andrew Karam
20160090
Ahmed Mohmed Ahmed Mohamdeen
20160038
Kirloes Boles Kamal
20160312
Kiroles Samer Ibrhaim
20160314
Mina Atef Youesf
20160460

December 20, 2018

Contents

1 Motivation

As the technology evolves the programs need more and more memory to run on, but to execute a process it must be on memory available for the CPU to access it Virtual memory is a technique that allows the execution of processes that are not completely in memory. One major advantage of this scheme is that programs can be larger than physical memory.
Our project is virtual memory manager, To convert logical addresses to physical ad-

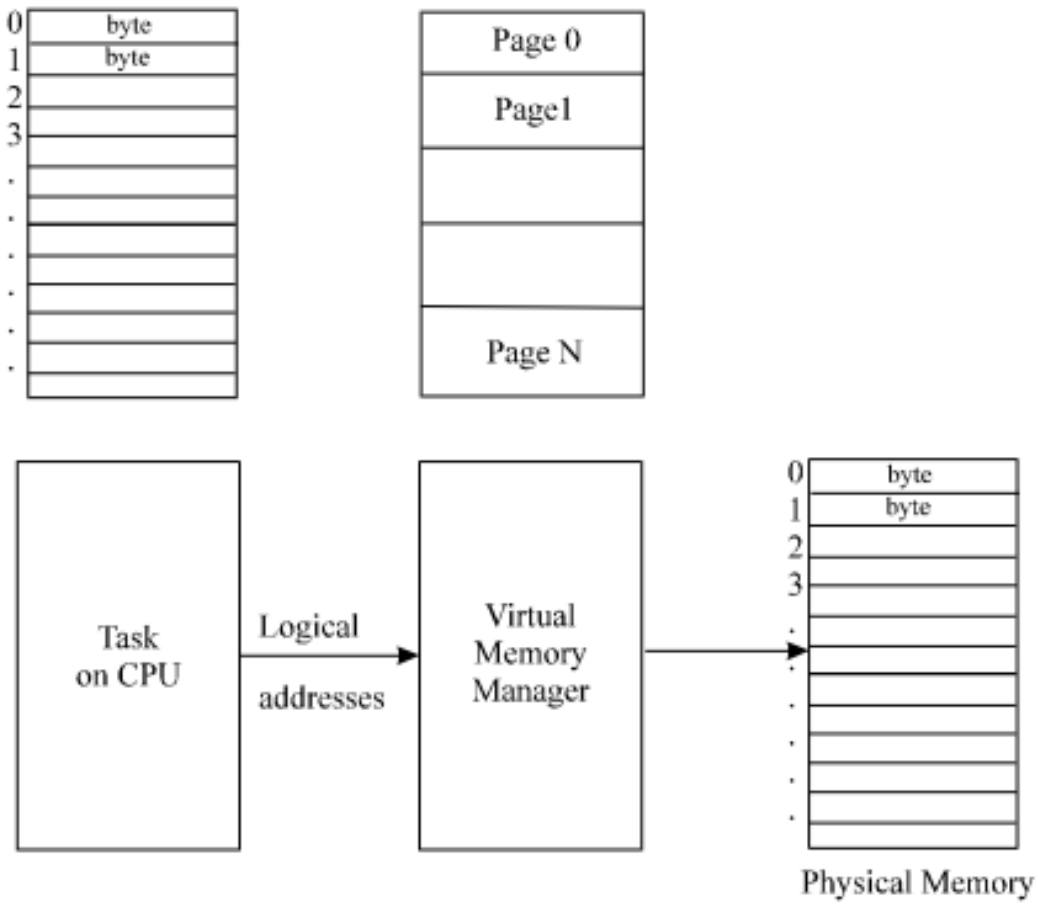

Figure 6.1: Mapping of Logical Address to Physical Memory Address

dresses to store in main memory.

2 Be Deep in Project

Our project contains of several tables

1. Physical memory
2. Page Table
   it contains the frame number of each each processes , it get from virtual memory
3. TLB
   it contains the frame number and page number, it's design to make search is very fast "cache"
4. Virtual Memory(the binary file)

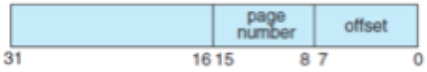

Figure 9.33 Address structure.

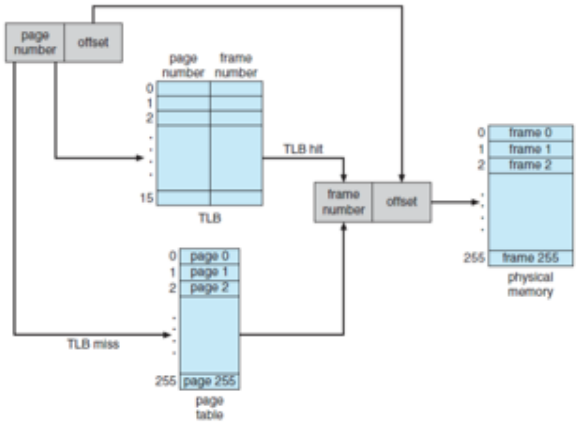

Figure 9.34 A representation of the address-translation process.

We take logical address from addresses file, each address is 32bit. We get the first 16 bit of this addresses. then, we get the offset number by masking the first 8 bits.
And the page Number by shifting the 16 bit to the right by 8 bits. Then we search the transfer look-up buffer table for the frame number if it returns that's considered a hit, if it's not found we search the page table and if it's not there that means it's a page fault and have never been into memory so we read the frame data from the virtual memory file and put it into memory and assign appropriate frame number to it.

3 Code Implementation

In the main Function simply we initialize each of the TLB and the page table to be (-1) because page/frame number can be 0.
Then we create two file streams one to read the addresses file which is given and a write stream on a file we will export our results in, Then for each read address we extract the page number and the offset from the logical address by masking as explained earlier.