Sultanov Yeskendir M.
Master's Student,
Kazakh-British Technical University, Almaty, Kazakhstan
Duysembaev Anuar
Doctor of physical and mathematical sciences,
Kazakh National al-Farabi University

DESIGN OF THE INVERTED PAGE TABLE

Summary. *The operating system takes one of two variants when solving any memory management problem. The first variant is dividing things into variable-sized pieces and it is called segmentation. It has inherent difficulties, when dividing a space into different-size chunks. The second variant to divide space into fixed-sized pieces and it is called paging. It may be worth considering the second variant and system that uses this variant called paged systems. The choice of the data structure for paged systems significantly effects on the performance of paged systems. In this paper, as a data structure we take Inverted Page Table (IPT), which optimizes parameters as a size of memory occupied by the page table and the time to transformation virtual address into physical address.*

Key words: *Inverted Page Table (IPT), virtual memory.*

1. Introduction

The virtual memory paging assumes that all memory of the computer as a physical as well as virtual divided into successive pages of the same size and when the programs run, then each element of the program receives a virtual address, and correspondence between the virtual and physical addresses is done automatically by the operating system.

The exchange between auxiliary and main memory is realized by whole pages, and during the exchange CPU switches to execution of commands of other programs (see Figure 1). If during the execution of the program takes place a link to a page missing in the main memory, the page faults occur (failure).

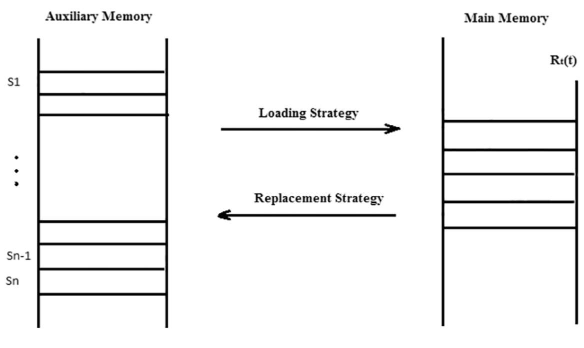

Figure 1. The exchange between auxiliary and main memory — development of author

The program is interrupted, at least, for the time necessary to swap the page you want. In this case, one or some of the pages of the program is deleted from the main memory, that is, the memory occupied by them is considered to be free. If the content of the page to be deleted, distorted during her stay in the main memory, then the system provides an overwriting of modified pages in the Auxiliary memory while preserving the original content of the pages of the original. Otherwise, the need to rewrite the page at the Auxiliary memory is not necessary. Select a page (or group of pages) to be deleted from the main memory; the system is carried out in accordance with a particular algorithm called page replacement strategy.

2. Page Table

The Page Table is a data structure that is used to map virtual addresses to physical addresses. Each row of the page table usually consists 1-bit cell, for show the presence in the main memory, virtual page number, and physical frame number [1–5]. Below in Figure 2 you can see an example of Page Table.

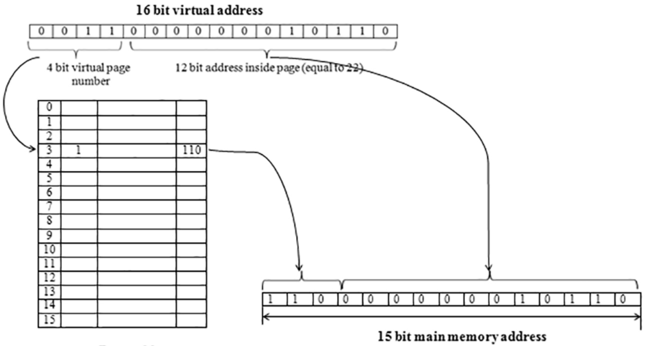

Figure 2. Page Table — development of author

Let’s look in detail, in order to understand how it works. Firstly, operating system generates a virtual address, then we go to row that equal to the virtual page number, then if valid bit is 1, it means that it exist in main memory, after that we take frame number of this row, finally, we get physical address with frame number and offset from virtual address.

3. Inverted Page Table

The presence of regular table scheme pages make such a scheme is not sufficiently effective. Storing in the main memory of the computer of excessive use information, so information on inactive pages, which makes up most of the page table, will slow down the entire storage management system and negatively affect the performance of the system as a whole. This leads to the idea that it would be more economical, at any point in the process of activity, stored in the main memory only information regarding only those pages that are currently in main memory. In this connection, instead of a regular page tables, we will use the inverted page table (see Figure 3).

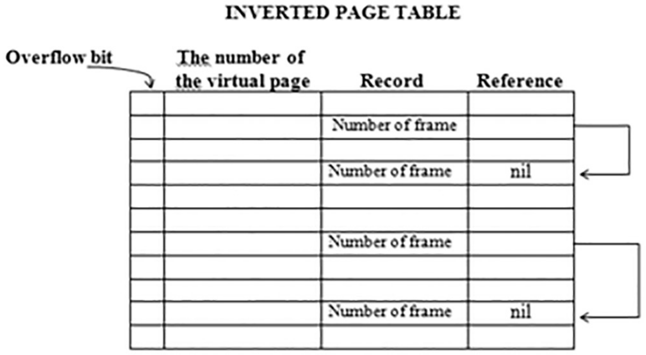

Figure 3. Inverted Page Table — development of author

IPT[I]:

bit_ovfl	nv	nf	ref

- bit_ovfl — field in which bit overflow is recorded
- nv — field in which page number is recorded
- nf — field in which frame number is recorded
- ref — link to the next link, if it exists

An inverted page table, at any time of the program, stores information about a virtual page part of the process, namely those in which the copy currently in the main memory, therefore it reduces the size of the memory occupied by page table, in spite of fact that we search through linked list elements [6–7]. Finally, if we get page faults, it means that page does not exist in the main memory and system should load it from the auxiliary memory.

4. Conclusions

We have introduced design of the Inverted Page Table for the transformation of virtual address into physical address. The Inverted Page Table in contrast to the page table stores only active pages, this leads to the idea that it would be more economical, at any point in the process of activity, stored in the main memory only information regarding only those pages that are currently in main memory. It optimizes the size of memory occupied by page table and it reduces the time of transformation virtual address into physical as system will not spend time to record information about inactive pages.

References

1. S. L. Harris, D. M. Harris. Digital Design and Computer Architecture. Elsevier Inc. 2nd Ed. — 2012.

2. Randal E. Bryant, David R. O’Hallaron, Carnegie Mellon University. Computer Systems: A Programmer’s Perspective, third edition. 2015.

3. Amir Kamil. Inverted Page Tables. Internet resource: http://web.eecs.umich.edu, 2004.

4. Madhusudhan Talluri, Mark D. Hill, Yousef A. Khalidi. A New Page Table for 64-bit Address Spaces, 1995.

5. Mark A. Holliday, Department of Computer Science of the Duke University, Durham, NC27706. Page Table Management in Local/Remote Architectures, 1988.

6. A. Tanenbaum. Computer Architecture. 5th Ed. Prentice Hall, NY, 2006.

7. W. Stallings. Computer Organization and Archtecture. Pearson Ed., 2006.