

Memory Management: Virtual Memory, Paging and Page Replacement Algorithms

Arsheen Qureshi¹, AshwaniSharma², Chirayu Jain³, Vaishali Sharma⁴

Department of Computer Science Engineering

Arya College of Engineering and Research Centre, Kukas, Jaipur, Rajasthan, India

E-mail: ¹arsheen.qureshi.786@gmail.com, ²sharmaashwani28121998dec@gmail.com, ³chirayujain10.cj@gmail.com,
⁴vaishali646@yahoo.com

Abstract

In this era of modernization and urbanization Computer Science is dealing with several theoretical foundations of information and computation together with practical techniques for implementation and approximation of these foundations. In order to perform several tasks an optimal utilization of memory is required i.e. memory management. Since memory management has several aspects of its own, it is significant to understand the need of Virtual Memory. Memory management revolves around Segmentation, Paging and various algorithms. So in this paper we will discuss one of the important characteristic of Memory management i.e. paging along with Replacement of Pages by using various algorithms and their pros and cons. We will also discuss about basic mechanism to deal with anomaly of Page Fault.

Keywords: Memory Management, Virtual Memory, Paging and Page Fault.

Introduction

As it is important to know, how Operating System manages memory, most of the computers have memory hierarchy. Memory management system can be classified into two divisions, those which leads to swap in and swap out between disk and main memory and those that do not. CPU communicates with main memory for the execution of each and every program. The Cache memory provides the speed and maps data present in main memory. Since the memory space of main memory is limited, in order to deal with this problem main memory converts some part of Auxiliary memory into itself but temporarily. This temporary main memory is termed as Virtual Memory.

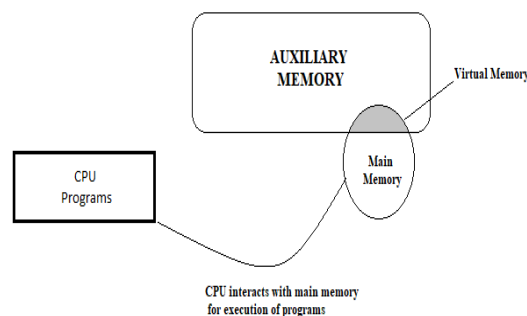


Figure 1 Pictorial representation of the formation of Virtual Memory

Paging Concepts

The logical memory constitutes information, this stored information gets converted into pages. Paging involves the breaking of physical memory into blocks of fixed size. The blocks of the fixed size are called as Frames. A Process is basically the program in execution. Paging has its hardware support which is accomplished by page number and its offset.

The address formed by the CPU consists of these two fields. In the page table the page number acts as an index. An additional bit is attached with each entry in page table is a valid-invalid bit as discussed in [1]. Every page table consists of some protection bits which defines a page is to be read-write or read only. When this bit is set to valid then the associated page is in logical address space and is considered as a legal page. Invalid bit refers that the page is not in process. Paging can be classified into two types- Pure and Demand paging. For the process to get executed some pages are required. In Pure paging the page table consists of all the pages which are not even needed by the process, hence it comprises of all the pages. In Demand paging, only entailed pages will exist in the page table.

Page Fault

When an application tries to access a block of memory that is not stored in physical memory or memory, as indicated in [2], it is called as a Page Fault. There is a special, small, and fast hardware cache, known as a TLB buffer. TLB contains only a few entries in the page table. The page number is provided by the TLB when the logical address starts with the CPU. If the page number is not required in the TLB, this status is known as a TLB slip. We will modify the page processing process to replace the page using [3]:

- Find the location of the desired page on the disk.
- Find a free frame.
- If there is a free frame, use it.
- If there is no free frame, use the page replacement algorithm to select the victim's box.
- Write the victim page on the disk. Change the page and frame of the frame accordingly.
- Read the desired page into the (newly) free frame; change the page and frame tables.
- Restart the user process.

The figure given below shows the steps needed in order to handle a Page Fault.

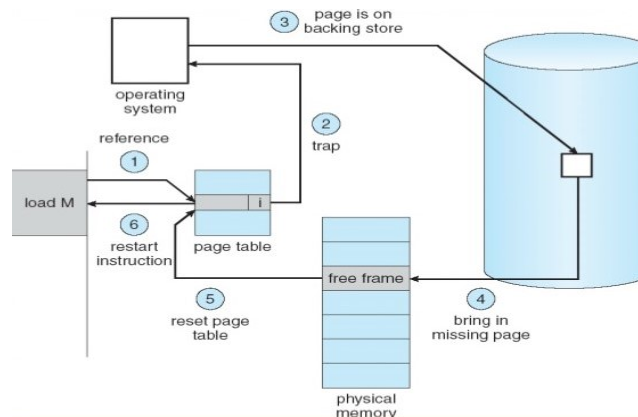


Figure 2 Steps in handling a Page Fault

Page Replacement Algorithms

There are innumerable page replacement algorithms. Three of them are explained here:

FIFO (First in First out) Algorithm

In First in First out page replacement algorithm each page has its time relevant to it. That page must be replaced which was the oldest in the reference frame. A new frame is always inserted at the tail of the queue.

In the FIFO replacement algorithm, Anomaly Belyda is defined as a phenomenon in which, for a memory access template, increasing the number of pages leads to an increase in page errors, as noted in [4].

Consider a system in which there are no pages in memory and there is an algorithm to replace the FIFO page. Consider the following reference string:

1,2,3,4,1,2,5,1,2,3,4,5

Case-1: Consider a 3-frame system using the FIFO algorithm in this reference system, a total of nine-page errors. The chart below shows the page error pattern:

1	1	1	2	3	4	1	1	1	2	5	5
	2	2	3	4	1	2	2	2	5	3	3
		3	4	1	2	5	5	5	3	4	4
PF	PF	PF	PF	PF	PF	PF	X	X	PF	PF	X

Case-2: consider the system with 4 frames for the given reference frame we use FIFO algorithm which gives 10 page faults. The diagram gives the pattern for this:

1	1	1	1	1	1	2	3	4	5	1	2
	2	2	2	2	2	3	4	5	1	2	3
		3	3	3	3	4	5	1	2	3	4
			4	4	4	5	1	2	3	4	5
PF	PF	PF	PF	X	X	PF	PF	PF	PF	PF	PF

In the above example it can be seen that on using FIFO page replacement algorithm as the number of frames increases, the number of page faults also increases from 9 to 10. This is known as Belady's anomaly.

Optimal Page Replacement Algorithm

Anomaly Baladi works using the optimal page replacement algorithm because the page error is minimal.

According to this algorithm, the page is replaced with a reference frame that is not used for the longest period of time [5].

Consider the system with no page loaded in the memory then by using optimal page replacement algorithm for the given reference string we can see:

7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2		2		2		2						7		
	0	0	0		0		4		0		0					0			
		1	1		3		3		3		1					1			

Here 9 page faults occur. This algorithm is practically impossible as finding the future usage of any page is not possible.

Least recently Used (LRU) Algorithm

Despite the fact that over the years there are many alternative algorithms, the least recent use algorithm (LRU) is predominant because of its efficiency and simplicity.

However, the LRU can show known performance problems with common access patterns on pages larger than main memory (for example, large cycles) described in [6].

Last used last used for every page in LRU. The FIFO algorithm focuses on when the page is transmitted to memory and an optimal algorithm is used around the time. The LRU is an approximation of the optimal algorithm, leading to a page replacement that has not been used for the longest period.

Consider the system with no page loaded in the memory then by using LRU page replacement algorithm for the given reference string we can see:

7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1,

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7
F	F	F	F		F		F	F	F	F		F		F		F			

The total number of page faults are 12 while the same reference string has 9 page faults with Optimal page replacement algorithms which is unrealizable. There are many of other algorithms present which is used in some special cases and are the extended versions of these three algorithms. Some of them are- Second Chance algorithm, clock page replacement algorithm, WSClock page replacement algorithm, NRU etc.

Conclusion

Memory in computer system is used for storing information either permanently or temporarily. Its allocation is a process by which different programs are allotted with physical or virtual memory space as the memory allocation retains the partial or complete of computer memory for execution of different processes.

Paging being a memory management scheme permit the physical address space of a process to be non-contiguous.

Page replacement algorithms are used to prevent the over allocation of memory by minimizing the page fault service routine to include the page replacement. Out of these three algorithms operating system can use only two. In the present scenario, there exist manifold page replacement algorithms but there is a need for more efficient way to organise the memory. This could be done by increasing the Hit Ratio which will in turn reduce the page fault.

References

1. Operating System Concepts by Abraham Silberschatz, Peter Baer Galvin and Greg Gagne section 8.6.1 Structure of Page Table, page no. 378.
ISBN: 978-81-265-0962-1.
2. Page Fault and its complete description from https://techterms.com/definition/page_fault.
3. S. Jananee, "Page Replacement", International Journal Of Compute Science and Information Technology Research, vol 2, issue 3, Month july-september, 2014.
4. Kirby McMaster, Samuel Sambasivam, Nicole Anderson, "Belady's anomaly", Proceeding Of Informing Science and IT Education Conference 2010.
5. Operating System Concepts by Abraham Silberschatz, Peter Baer Galvin and Greg Gagne section 9.4.3 Optimal Page Replacement page no. 414. ISBN: 978-81-265-0962-1.
6. Yannis Smaragdakis, Scott Kalpan, Paul Wilson, "EELRU: Simple and effective adaptive page replacement", ACM SIGMETRICS Performance Evaluation Review 27(1), 122-133, 1999.