1. Consider the following memory references have been generated by a program:

0100, 0432, 0101, 0612, 0102, 0103, 0104, 0101, 0611, 0102, 0103, 0104, 0101, 0610, 0102, 0103, 0104, 0101, 0609, 0102, 0105

At 100 bytes per page, determine the no. of page faults in each of the following page replacement policies assuming 3 frames are available. Also assume that initially none of page is in main memory.

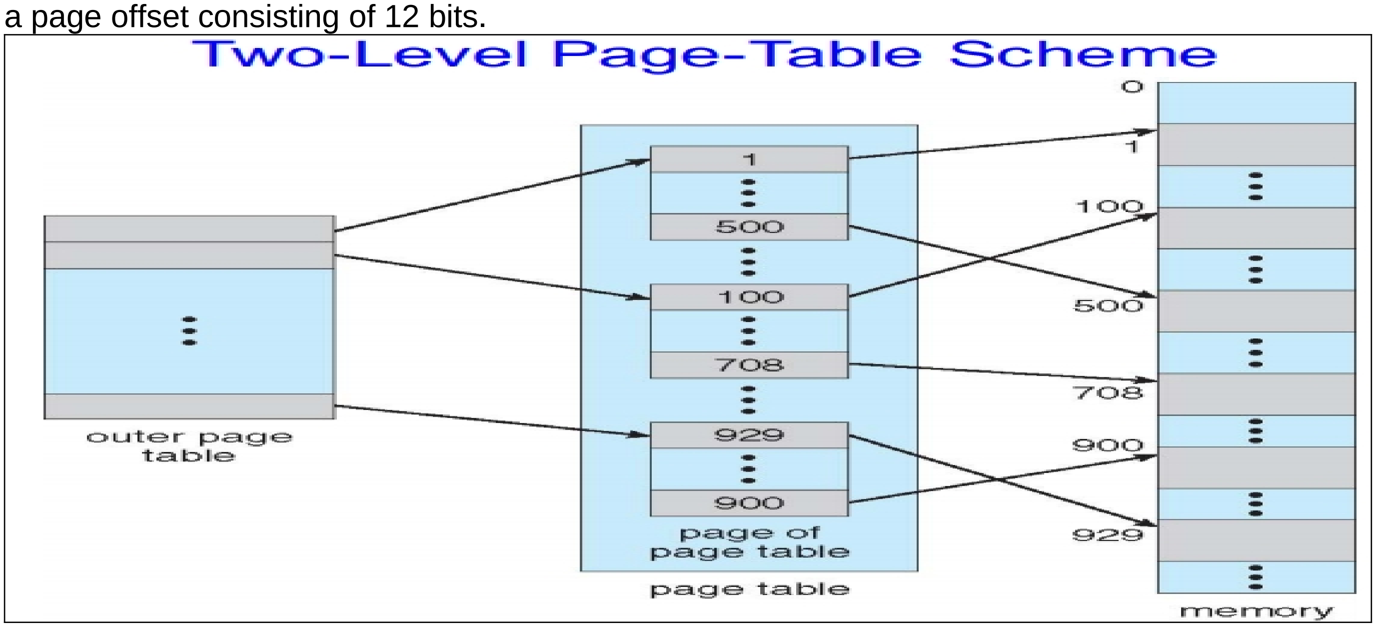
(i) LRU (ii) FIFO

Ans: Given, page size = 100 bytes. Thus, each memory reference should be divided by 100 to get the page number, eg., 0100/100 = 1, 0432/100 = 4, 0612/100 = 6 etc. This way, we can generate the page string as: 1, 4, 1, 6, 1, 1, 1, 1, 6, 1, 1, 1, 1, 6, 1, 1, 1, 1, 6, 1, 1 ➔ Page string should contain unique pages ➔ 1, 4, 1, 6, 1, 6, 1, 6, 1, 6, 1. Now you can find out the page faults with page replacement algorithms, LRU and FIFO.

2. A system with two-level paging scheme

Most modern computer systems support a large logical address space (2^{32} to 2^{64}). In such an environment, the page table itself becomes excessively large. For example, consider a system with a 32-bit logical address space. If the page size in such a system is 4 KB (2^{12}), then a page table may consist of up to 1 million entries ($2^{32}/2^{12}$). Assuming that each entry consists of 4 bytes, each process may need up to 4MB of physical address space for the page table alone. Clearly, we would not want to allocate the page table contiguously in main memory. One simple solution *to* this problem is to divide the page table into smaller pieces. We can accomplish this division in several ways.

One way is to use a two-level paging algorithm, in which the page table itself is also paged (Figure given in next page). For example, consider again the system with a 32-bit logical address space and a page size of 4 KB. A logical address is divided into a page number consisting of 20 bits and a page offset consisting of 12 bits.



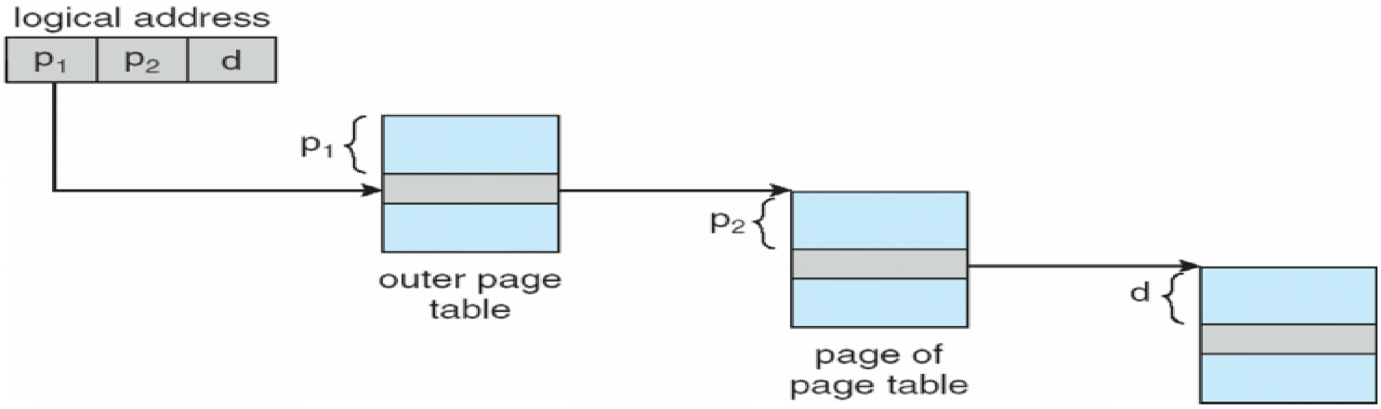
Two-Level Paging Example

- A logical address (on 32-bit machine with 1K page size) is divided into:
  - a page number consisting of 22 bits
  - a page offset consisting of 10 bits
- Since the page table is paged, the page number is further divided into:
  - a 12-bit page number
  - a 10-bit page offset
- Thus, a logical address is as follows:
-

page number		page offset
p_1	p_2	d
12	10	10

- where p_1 is an index into the outer page table, and p_2 is the displacement within the page of the inner page table
- Known as **forward-mapped page table**

Address-Translation Scheme



3. Suppose that a disk has 200 cylinders, numbered 0 to 199 . The drive is currently serving a request at cylinder 53, and the previous request was at cylinder 25. The queue of pending requests, in FIFO order is

98, 183, 37, 122, 14, 124, 65, 67

Starting from the current head position, show the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling disciplines?

I. FCFS
II. SSTF
III. LOOK
IV. C-LOOK
V. Scan
VI. C-Scan

ANSWER

I. FCFS

0 14 37 53 65 67 98 122 124 183 199

53 98

183

37

122

14

65 124

67

The total distance:

(98-53)+(183-98)+(183-37)+(122-37)+(122-14)+(124-14)+(124-65)+(67-65)= **640 Cylinders**

In another way: **FCFS:** 53 , 98 , 183 , 37 , 122 , 14 , 124 , 65 , 67

ii. SSTF

0 14 37 53 65 67 98 122 124 183 199

65

37 67

14

98

122

124

183

The total distance:

(65-53)+(67-65)+(67-37)+(37-14)+(98-14)+(122-98)+(124-122)+(183-124)= **236 Cylinders**

In another way:

SSTF: 53 , 65 , 67 , 37 , 14 , 98 , 122 , 124 , 183

iii. Scan

0 14 37 53 65 67 98 122 124 183 199

65

67

98

122

124

183

199

37

14

The total distance:

(65-53)+(67-65)+(98-67)+(122-98)+(124-122)+(183-124)+(199-183)+(199-37)+(37-14)= **331 Cylinders**

In another way:

Scan: 53 , 65 , 67 , 98 , 122 , 124 , 183 , 199 , 37 , 14

Iv. Look

0 14 37 53 65 67 98 122 124 183 199

65

67

98

122

124

183

37

14

The total distance

(65-53)+(67-65)+(98-67)+(122-98)+(124-122)+(183-124)+(183-37)+(37-14)= **299 Cylinders**

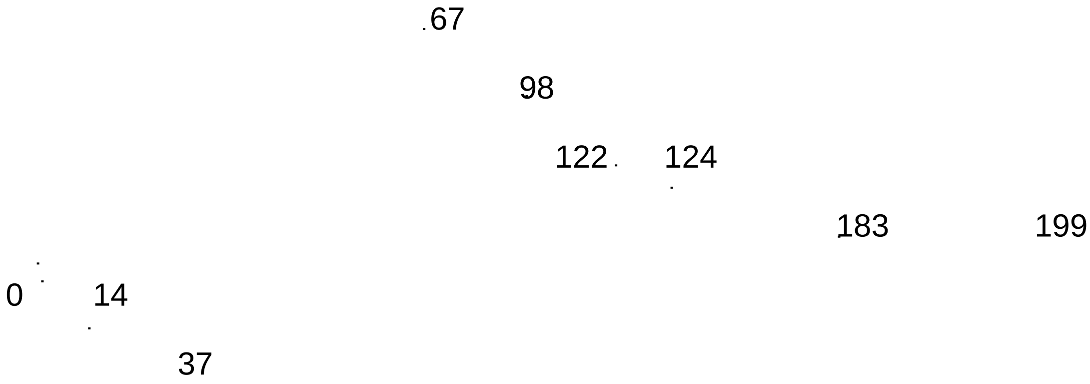
In another way

LOOK: 53 , 65 , 67 , 98 , 122 , 124 , 183 , 37 , 14

V. C-Scan

0 14 37 53 65 67 98 122 124 183 199

65



The total distance

(53-65)+(67-65)+(98-67)+(122-98)+(124-122)+(183-124)+(199-183)+(14-0)+(37-14)=

In another way:

C-Scan: 53 , 65 , 67 , 98 , 122 , 124 , 183 , 199 , 0 , 14 , 37

C-LOOK

0 14 37 53 65 67 98 122 124 183 199

The total distance:

(65-53)+(67-65)+(98-67)+(122-98)+(124-122)+(183-124)+(183-14)+(37-14) = **322 Cylinders**

In another way

C-LOOK: 53 , 65 , 67 , 98 , 122 , 124 , 183 , 14 , 37