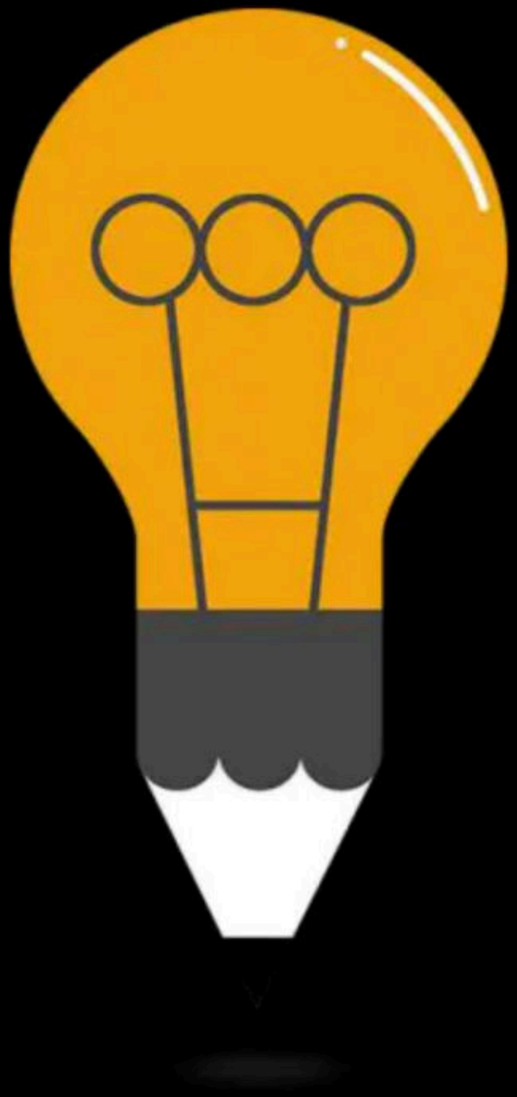
unacademy

File Allocation Methods

Comprehensive Course on Operating System for GATE - 2024/25

Vishvadeep Gothi • Lesson 39 • June 2, 2023

unacademy



Operating System
File System 2

By: Vishvadeep Gothi

Disk Formatting

Disk Blocks

Disk Blocks

Number of disk blocks = 2^{16}

Size of each block = 1KB

Total Size of disk?

Disk Blocks

Disk block address= 24-bits

Size of each block = 2KB

Total Size of disk?

Disk Blocks

Total disk size = 256GB

Block Size = 2KB

Disk block address?

Free Space Management

1. Free List
2. Bitmap Method

Free Space Management

1. No searching in free list, but in bitmap we search for first zero
2. Free list is faster in allocating a free block
3. Free list size is variable, where as bitmap size is constant

Question

A particular disk unit uses a bit string to record the occupancy or vacancy of its disk blocks with ‘0’ denoting vacant block and ‘1’ denoting occupied block. A 32-bit part of this string has Hexadecimal value of D4F2A001. The percentage of occupied blocks on the disk for this part is?

Question

A system directory is kept in 4 disk blocks each of size 2Kbytes. It is a single level-directory and each directory entry is of size 32-bits. The maximum number of files possible in this system is?

Question

Disk block address = 14 bits

Each disk block size = 1KB

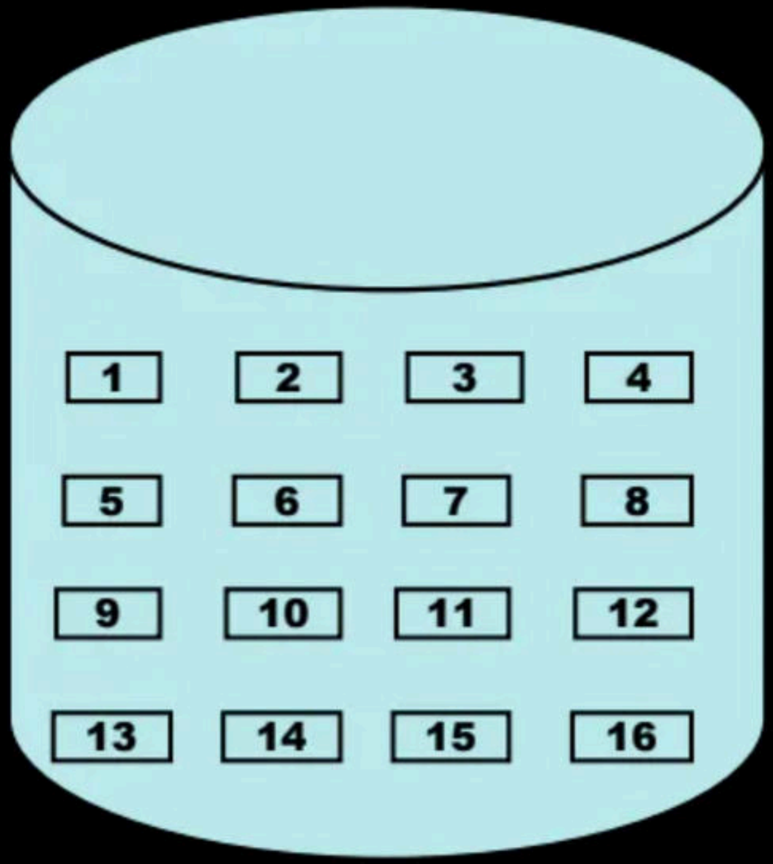
Maximum size of a file = ?

File Allocation Methods

File Allocation Methods

1. Contiguous Allocation
2. Linked Allocation
3. Indexed Allocation

Contiguous Allocation



Table

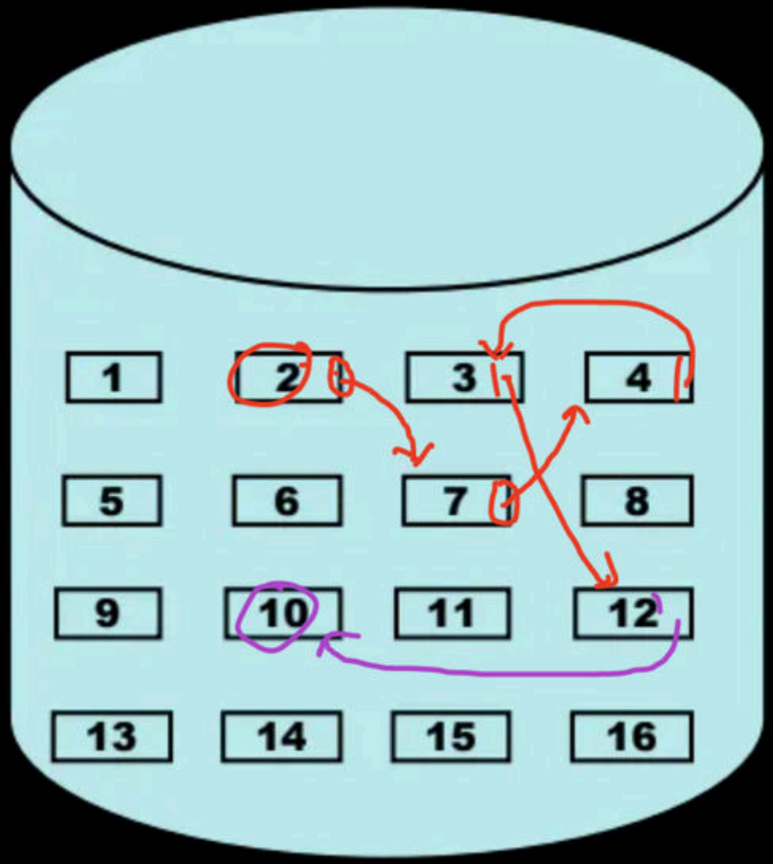
file name	starting block	no. of blocks

Contiguous Allocation

Performance:

1. Fragmentation: Internal, External
2. Increase in File size: Inflexible
3. Type of access: Sequential, Random/direct

Linked Allocation



file name	starting block no.	last block no.
abc.docx	2	12

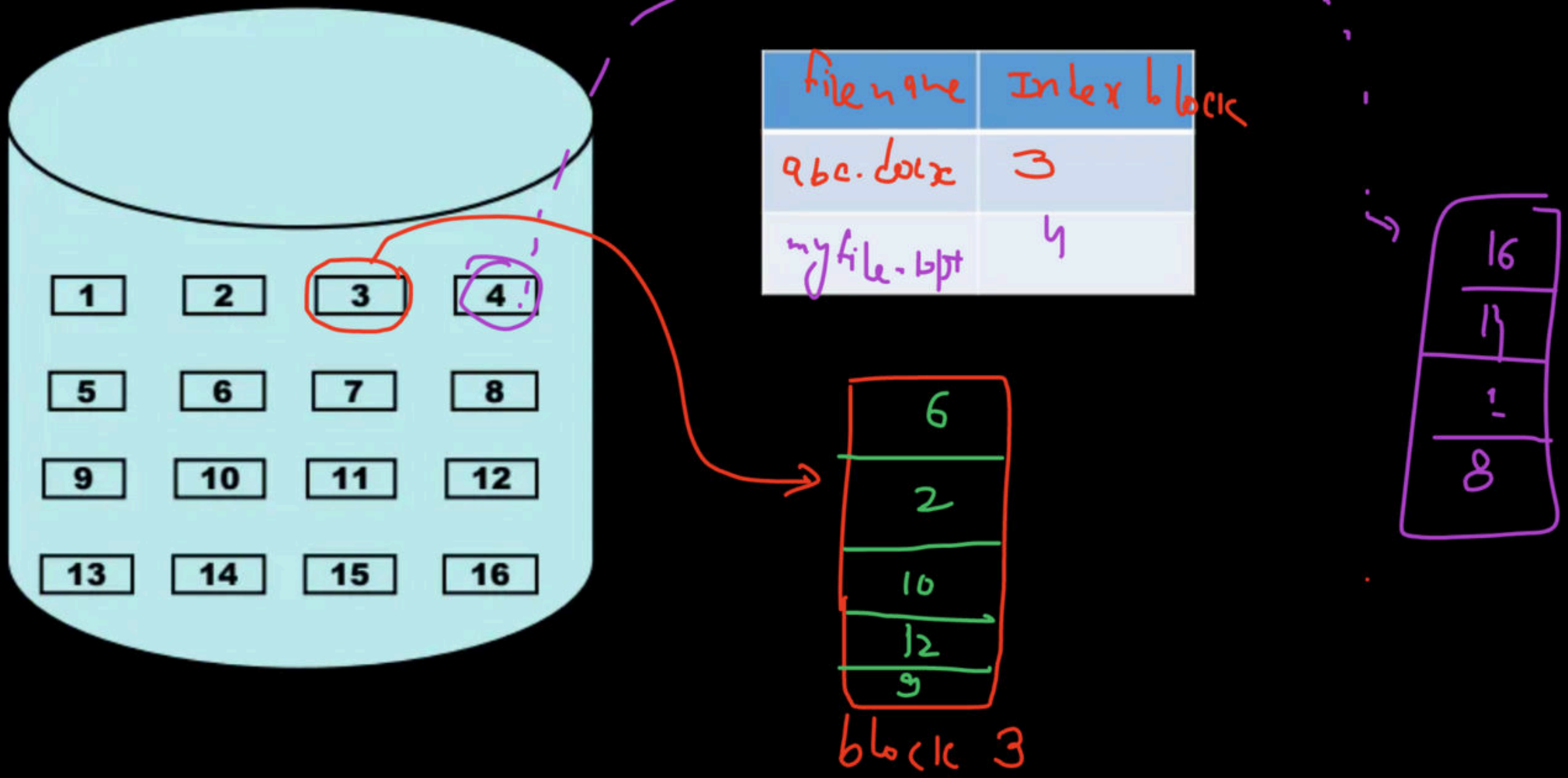
Linked Allocation

Performance:

1. Fragmentation: Internal
2. Increase in File size: Flexible
3. Type of access: Sequential

no direct access

Indexed Allocation



Indexed Allocation

Performance:

1. Fragmentation: Internal
2. Increase in File size: Flexible
3. Type of access: Sequential, Random/direct

Ques) Consider a disk which is having 256 disk blocks. Among all starting 4 blocks are used for index blocks.

no. of blocks remaining for file = 256 − 4 = 252

Question

Disk block address = 16 bits = 2B

Disk block size = 1KB

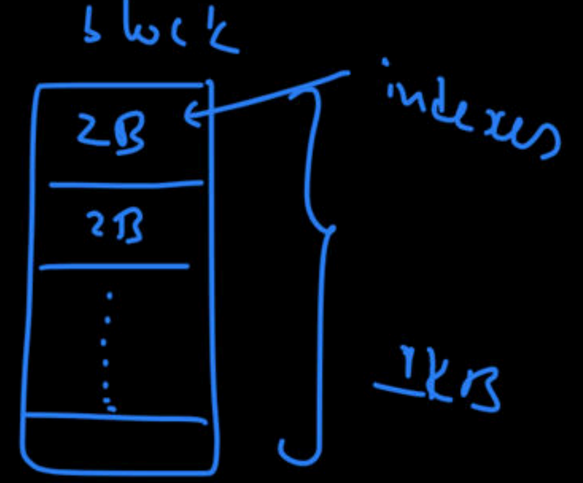
Index block = 1KB

Maximum file size?

no. of blocks in disk = 2^{16}

= Total no. of blocks − no. of index blocks

no. of indexes to be stored = 2^{16}



no. of indexes per block = $\frac{1KB}{2B}$

$= 5'2$

$= 2^9$

no. of blocks required to store 2^{16} indexes

$$= \frac{2^{16}}{2^3} = 2^7 = 128$$

no. of blocks remaining to store files $= 2^{16} - 128$

$$= 65408$$

max file size ≤ 65408 blocks

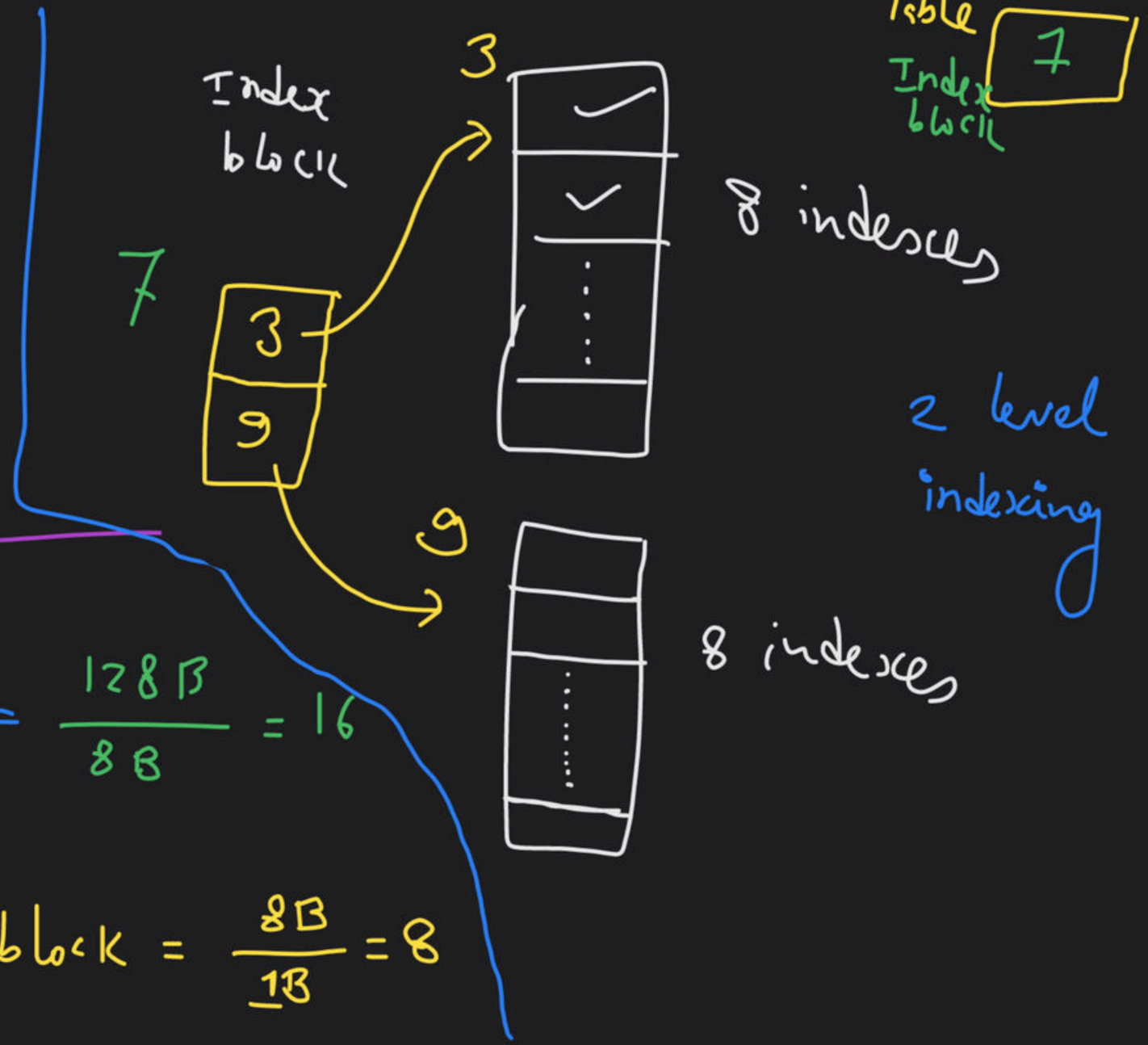
≤ 65408 KB

Example :-

D.B.A. = 8 bits = 1 byte

Block size = 8 bytes

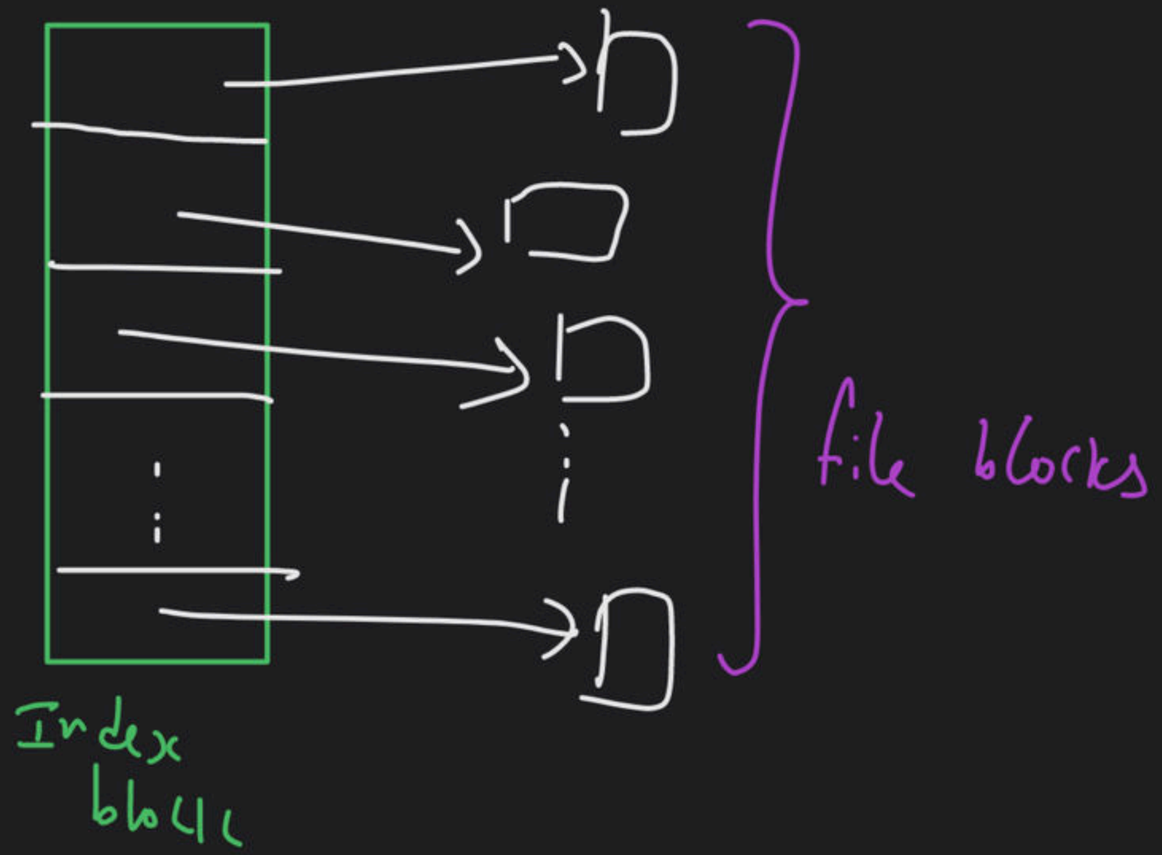
File = 128 bytes



no. of blocks to store file = $\frac{128\,B}{8\,B} = 16$

no. of d.b.A. (Indexes) stored in one block = $\frac{8\,B}{1\,B} = 8$

Single level indexing :-



max file size = 256 * 1kB = 256 kB

ex:-

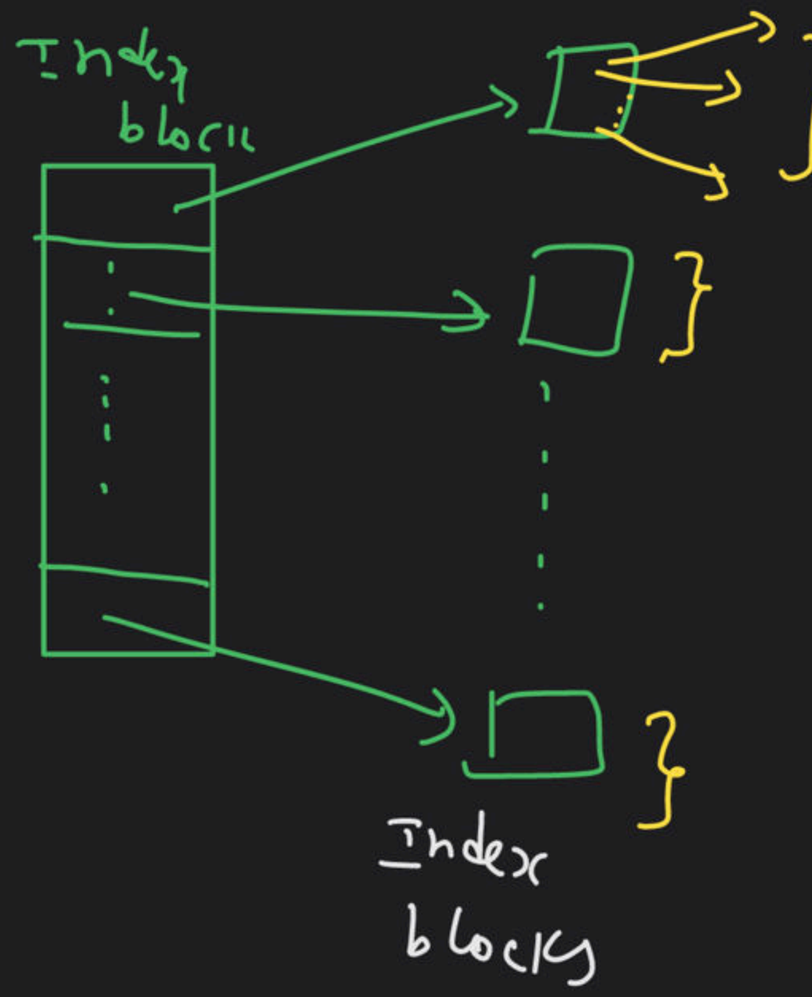
block size = 1 kB

D.B.A. = 4 bytes

no. of indexes per block = $\frac{1\text{kB}}{4\text{B}}$ = 256

2 unacademy

2- level indexing:-



max file size = 256 * 256 * 1KB

= 2^{16} KB

= 64 MB

3-level indexing:-

max file size = $256 * 256 * 256 * 1\,KB$

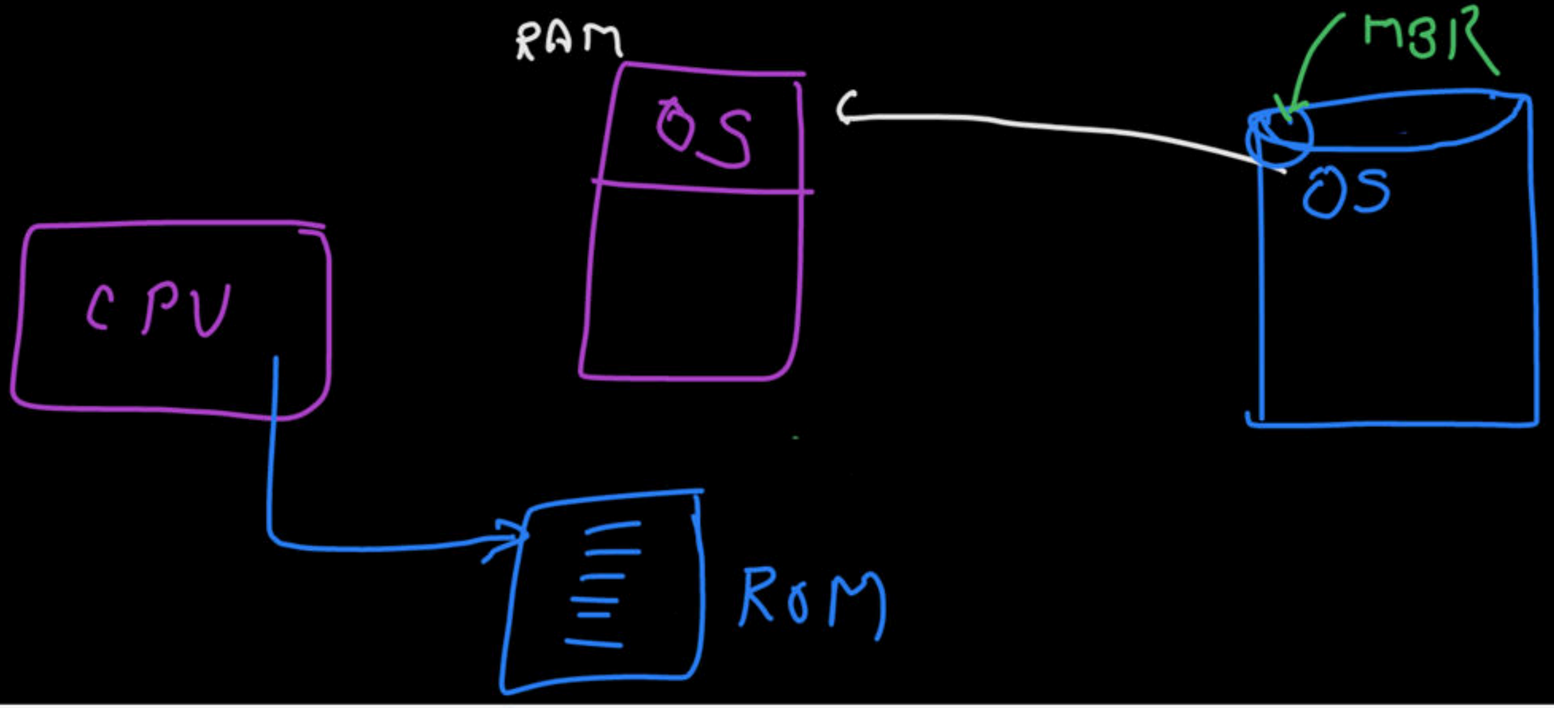
$= 2^{24} * 1\,KB$

$= 16\,GB$

Master Boot Record

A master boot record (MBR) is a special type of boot sector at the very beginning of partitioned computer mass storage devices.

- Contains the information regarding how and where the OS is located in the hard disk so that it can be booted in the RAM.



Unix I-node Structure

The inode (index node) is a data structure in a Unix-style file system that describes a file-system object such as a file or a directory.

Unix I-node Structure

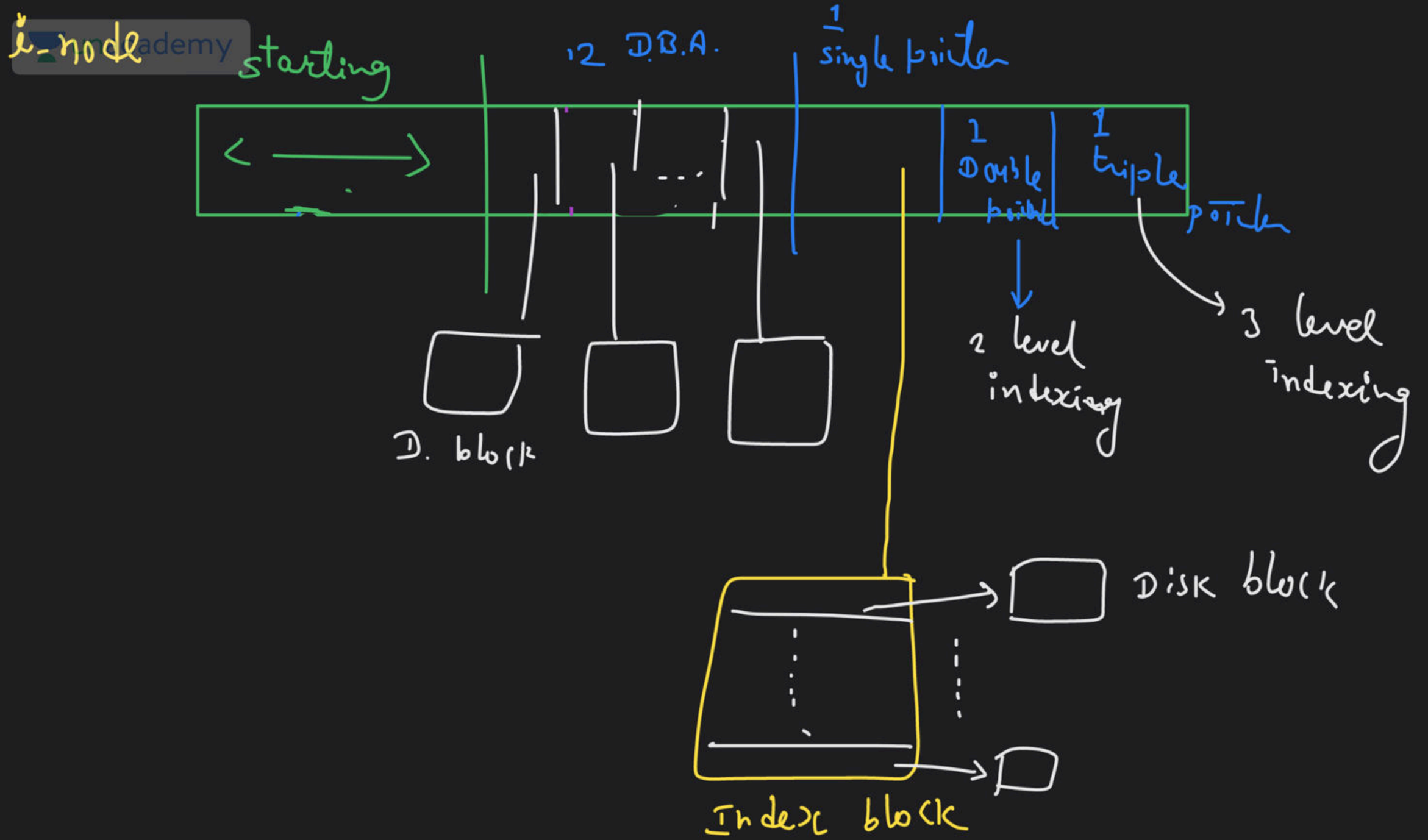
The inode (index node) is a data structure in a Unix-style file system that describes a file-system object such as a file or a directory.

- Each inode stores the attributes and disk block locations of the object's data.
- The number of Inode limits the total number of files/directories that can be stored in the file system.

Unix I-node Structure

The Inode contains the following information:

1. Administrative information (permissions, timestamps, etc.)
2. A number of direct blocks (typically 12) that contains to the first 12 blocks of the files
3. A single indirect pointer that points to a disk block which in turn is used as an index block, if the file is too big to be indexed entirely by the direct blocks
4. A double indirect pointer that points to a disk block which is a collection of pointers to disk blocks which are index blocks, used if the file is too big to be indexed by the direct and single indirect blocks.
5. A triple indirect pointer that points to an index block of index blocks of index blocks


i-node
starting
12 D.B.A.
1 single pointer
1 Double pointer
1 triple pointer
D. block
2 level indexing
3 level indexing
Disk block
Index block

Question GATE-2019

Ans = 4 GB

The index node (inode) of a Unix-like file system has 12 direct, one single-indirect and one double-indirect pointer The disk block size is 4 kB, and the disk block addresses 32-bits long. The maximum possible file size is (rounded off to 1 decimal place) _____ GB?

4 bytes

no. of indexes per block = $\frac{4\text{KB}}{4\text{B}} = 1\text{K}$

=

$$\left[12 + 1\text{K} + 1\text{K} * 1\text{K}\right] * 4\text{KB}$$

$12 * 4\text{KB} + 1\text{K} * 4\text{KB} + 1\text{K} * 1\text{K} * 4\text{KB}$

$48\text{KB} + 4\text{MB} + 4\text{GB} = 4.004008\text{ GB}$

≈ 4.0 GB

Question

A system directory is kept in 4 disk blocks each of size 2Kbytes. It is a single level-directory and each directory entry is of size 32-bits. The maximum number of files possible in this system is?

unacademy

GATE-1996

A file system with a one-level directory structure is implemented on a disk with disk block size of $4K$ bytes. The disk is used as follows:

Disk-block 0	File Allocation Table, consisting of one 8-bit entry per data block, representing the data block address of the next data block in the file
Disk-block 1	Directory, with one 32 bit entry per file:
Disk-block 2	Data-block 1;
Disk-block 3	Data-block 2; etc.

a. What is the maximum possible number of files?

unacademy

GATE-2004

In a particular Unix OS, each data block is of size 1024 bytes, each node has 10 direct data block addresses and three additional addresses: one for single indirect block, one for double indirect block and one for triple indirect block. Also, each block can contain addresses for 128 blocks. Which one of the following is approximately the maximum size of a file in the file system?

A. 512 MB
B. 2 GB
C. 8 GB
D. 16 GB

$10 * 1KB + 128 * 1KB + 128 * 128 * 1KB + 128 * 128 * 128 * 1KB$

$+ 2^{31} B$

2 GB

unacademy

GATE-2005

In a computer system, four files of size 11050 bytes, 4990 bytes, 5170 bytes and 12640 bytes need to be stored. For storing these files on disk, we can use either 100 byte disk blocks or 200 byte disk blocks (but can't mix block sizes). For each block used to store a file, 4 bytes of bookkeeping information also needs to be stored on the disk. Thus, the total space used to store a file is the sum of the space taken to store the file and the space taken to store the book keeping information for the blocks allocated for storing the file. A disk block can store either bookkeeping information for a file or data from a file, but not both.
What is the total space required for storing the files using 100 byte disk blocks and 200 byte disk blocks respectively?

(A) 35400 and 35800 bytes

(B) 35800 and 35400 bytes

(C) 35600 and 35400 bytes

(D) 35400 and 35600 bytes

Block Size = 100B

Solution

File Size	No. of blocks to store file	No. of blocks to store bookkeeping	Total blocks	Total Size
11050 Bytes	$\lceil 11050B / 100B \rceil = 111$	$\lceil 111 * 4B / 100B \rceil = 5$	116	
4990 Bytes	$\lceil 4990/100 \rceil = 50$	$\lceil 50 * 4 / 100 \rceil = 2$	52	
5170 Bytes	$\lceil 5170/100 \rceil = 52$	$\lceil 52 * 4 / 100 \rceil = 3$	55	
12640 Bytes	$\lceil 12640 / 100 \rceil = 127$	$\lceil 127 * 4 / 100 \rceil = 6$	133	
			356	356 * 100B = 35600B

unacademy

Solution

File Size	No. of blocks to store file	No. of blocks to store bookkeeping	Total blocks	Total Size
11050 Bytes				
4990 Bytes				
5170 Bytes				
12640 Bytes				

177 blocks

35400 Bytes

unacademy

GATE-2008

The data blocks of a very large file in the Unix file system are allocated using

A. continuous allocation
B. linked allocation
C. indexed allocation
D. an extension of indexed allocation

→ multilevel indexing

unacademy

GATE-2014

A FAT (file allocation table) based file system is being used and the total overhead of each entry in the FAT is 4 bytes in size. Given a 100×10^6 bytes disk on which the file system is stored and data block size is 10^3 bytes, the maximum size of a file that can be stored on this disk in units of 10^6 bytes is _____.

no of blocks in disk $= \frac{100 * 10^6}{10^3} = 10^5$ blocks

Total FAT entry size $= 4 * 10^5$ Bytes

$= 0.4 * 10^6$ Bytes

unacademy

max file size = $100 * 10^6 - 0.4 * 10^6$

$= 99.6 * 10^6$ Bytes

Ans = 99.6

unacademy

GATE-2017

In a file allocation system, which of the following allocation scheme(s) can be used if no external fragmentation is allowed ?

1. Contiguous
2. Linked
3. Indexed

A. 1 and 3 only
B. 2 only
C. 3 only
D. 2 and 3 only

unacademy

GATE-2019

The index node (inode) of a Unix-like file system has 12 direct, one single-indirect and one double-indirect pointer The disk block size is 4 kB and the disk block addresses 32-bits long. The maximum possible file size is (rounded off to 1 decimal place) _____ GB.

unacademy

Happy Learning.!

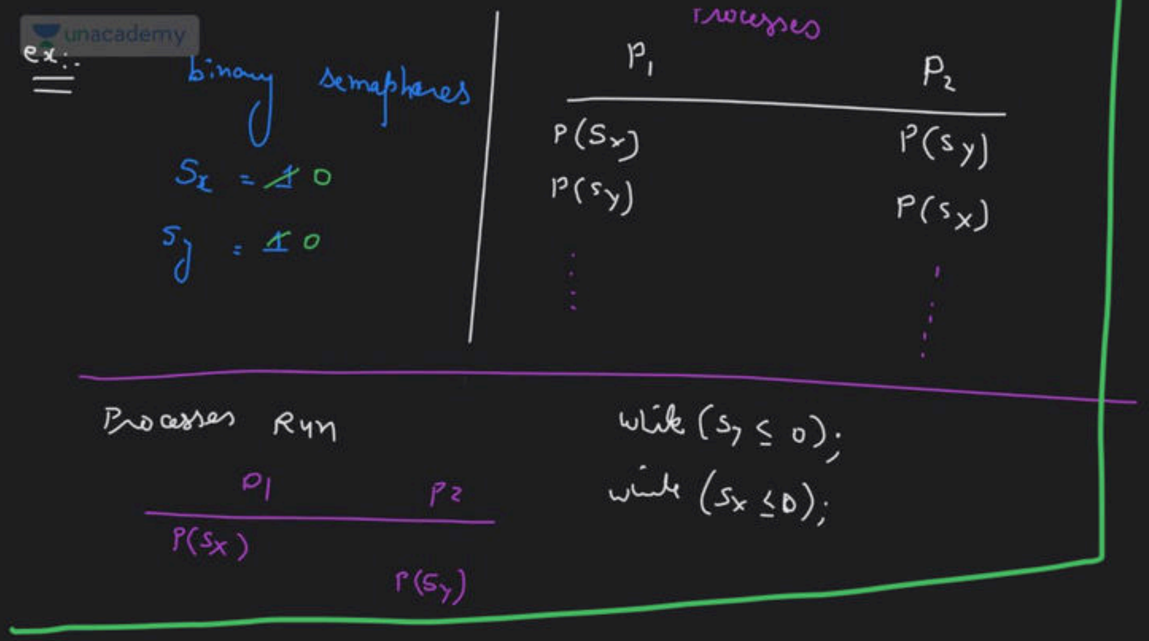
Remaining ⇒

Disk scheduling

→ M. L. Paging
→ Page Replacement
→ .

▲ 1 • Asked by Vaishnavij...

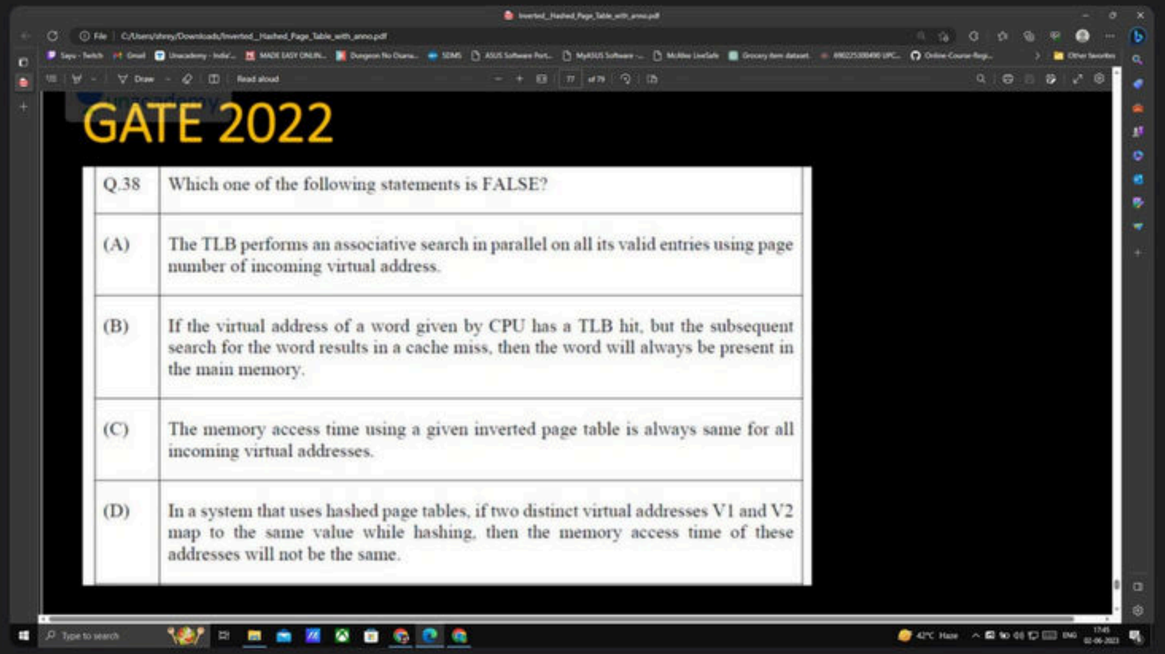
Please help me with this doubt



unacademy

▲ 1 • Asked by Shreyas

Please help me with this doubt

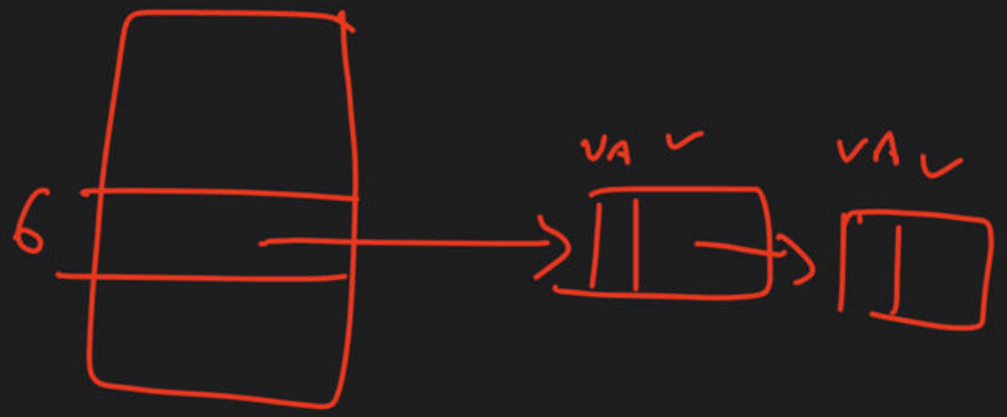


GATE 2022

Q.38	Which one of the following statements is FALSE?
(A)	The TLB performs an associative search in parallel on all its valid entries using page number of incoming virtual address.
(B)	If the virtual address of a word given by CPU has a TLB hit, but the subsequent search for the word results in a cache miss, then the word will always be present in the main memory.
(C)	The memory access time using a given inverted page table is always same for all incoming virtual addresses.
(D)	In a system that uses hashed page tables, if two distinct virtual addresses V1 and V2 map to the same value while hashing, then the memory access time of these addresses will not be the same.

H(VA1)

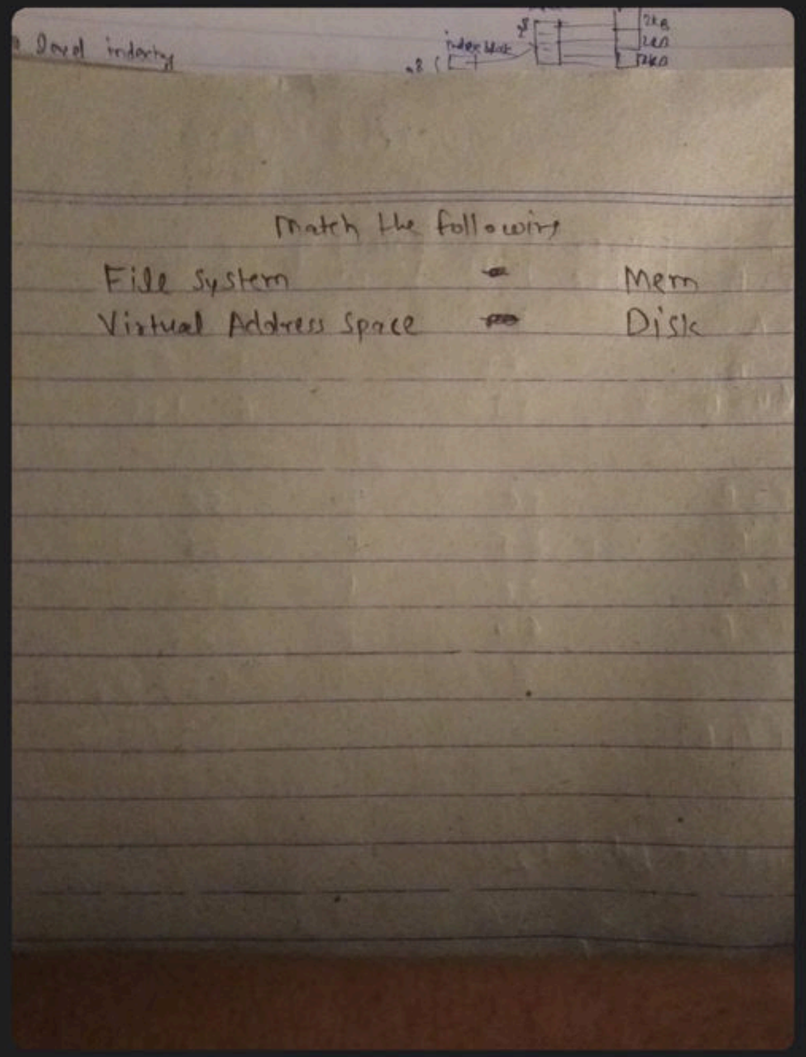
H(VA2)



unacademy

▲ 1 • Asked by Kumar

iski matching krni h



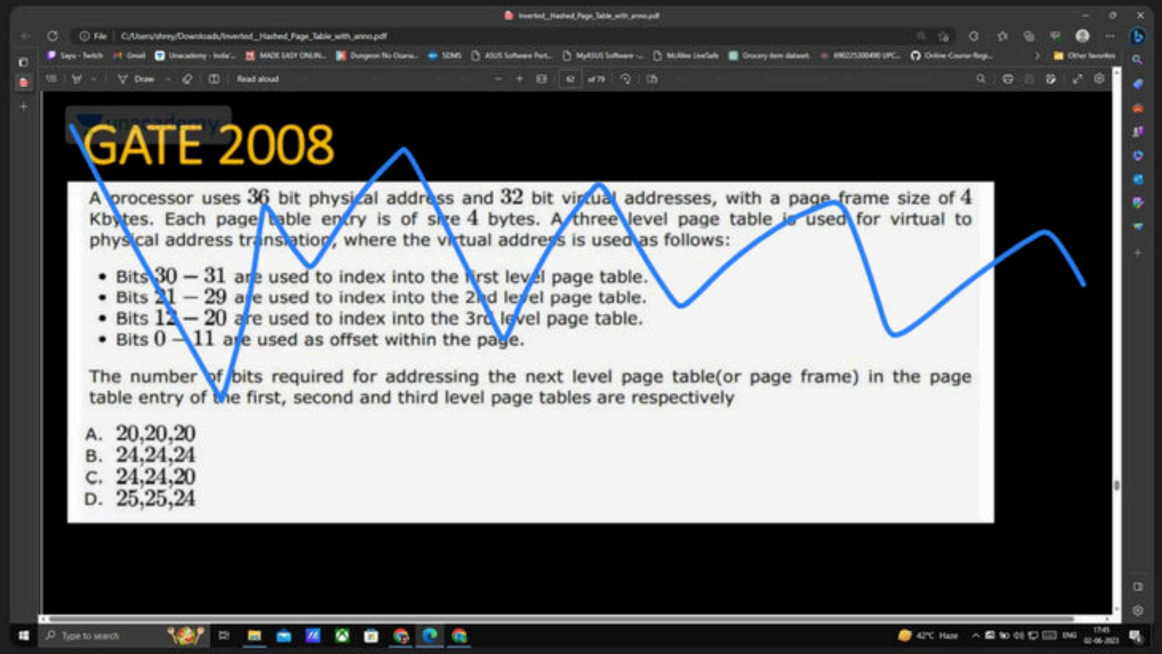
file sys. => H.D.D. (Sec. mem.)

V.A.S. => virtual

↳ Process size

▲ 1 • Asked by Shreyas

Please help me with this doubt



unacademy

▲ 1 • Asked by Spiderman

Sir please, isme cahe thoda samja dijiye

Question

A processor uses 2–level page tables for virtual to physical address translation. Page tables for both levels are stored in the main memory. Virtual and physical addresses are both 32 bits wide. The memory is byte addressable. For virtual to physical address translation, the 10 most significant bits of the virtual address are used as index into the first level page table while the next 10 bits are used as index into the second level page table. The 12 least significant bits of the virtual address are used as offset within the page. Assume that the page table entries in both levels of page tables are 4 bytes wide. Further, the processor has a translation look-aside buffer (TLB), with a hit rate of 96%. The TLB caches recently used virtual page numbers and the corresponding physical page numbers. The processor also has a physically addressed cache with a hit rate of 90%. Main memory access time is 10 ns, cache access time is 1 ns, and TLB access time is also 1 ns.
Assuming that no page faults occur, the average time taken to access a virtual address is approximately (to the nearest 0.5 ns)