PSG COLLEGE OF TECHNOLOGY - COIMBATORE - 4

July 2023 to September 2023

INDEX

List of users

1. CSE
2. AMCS
3. AMCS

K-D Tree

By Athmikha

What are K-D Trees???

A K-Dimensional Tree (also known as K-D Tree) is a space-partitioning data structure for organizing points in a K-Dimensional space. This data structure acts similar to a binary search tree with each node representing data in the multi dimensional space.

A K-D Tree is a binary tree in which each node represents a k-dimensional point. **Every non-leaf node in the tree acts as a hyperplane, dividing the space into two partitions.**

Why use a K-Dimensional Tree?

1. Nearest neighbor search.
2. Range queries.
3. Fast look-up.

How to Build a K-D Tree?

Algorithm 1: K-D Tree Construction

Function *kdtree(pointList, depth)*:

Input: a list of points, *pointList*, and an integer, *depth* ;

Output: a k-d tree rooted at the median point of *pointList* ;

```
/* Select axis based on depth so that axis cycles
   through all valid values                          */
```

let $axis := depth \mod k$;

```
/* Sort point list and choose median as pivot element
   */
```

select median by *axis* from *pointList*;

```
/* Create node and construct subtree                 */
```

let *node.location* := *median*;

let $node.leftChild :=$ kdtree(points in *pointList* before *median*, $depth + 1$);

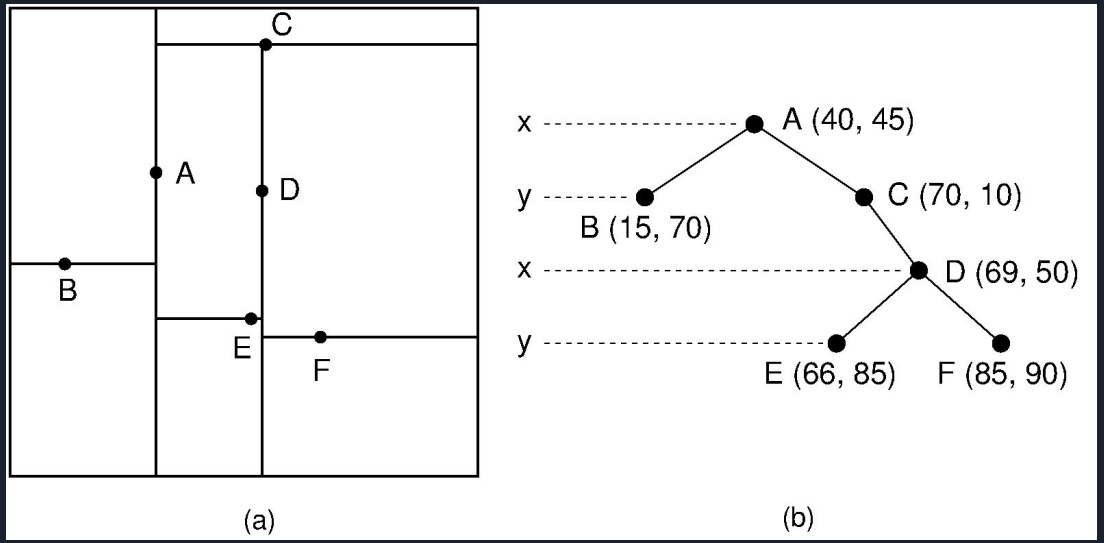
let $node.rightChild :=$ kdtree(points in *pointList* after *median*, $depth + 1$);

return *node*;

Insertion

To add a new point to a K-D tree, we can follow the same process as when adding an element to any other search tree. Start by traversing the tree from the root and move to either the left or right child depending on the location of the point being inserted in relation to the splitting plane. Once we reach the node under which the child should be located, add the new point as either the left or right child of the leaf node, depending on which side of the node’s splitting plane contains the new node

Given a list of points (40,45),(70,10),(15,70),(69,50),(85,90),(66,85)



Deletion

When deleting a node from a K-D tree, we first need to locate the node to be deleted, just like deleting from a Binary Search Tree. If the node has no children, it can be removed from the tree without affecting its structure. If the node has a descendant, we need to find the right descendant of the node that can replace it without violating the K-D tree property.

- If the node to be deleted has children to the right, we must find the node with the minimum value along the targeted axis from the right subtree.
- Otherwise, we need to find the point with the maximum value from the subtree rooted at the left child.

Nearest Neighbour Search

K-D trees are widely used for nearest-neighbor searches, where the objective is to find the point in the tree that is closest to a given query point.

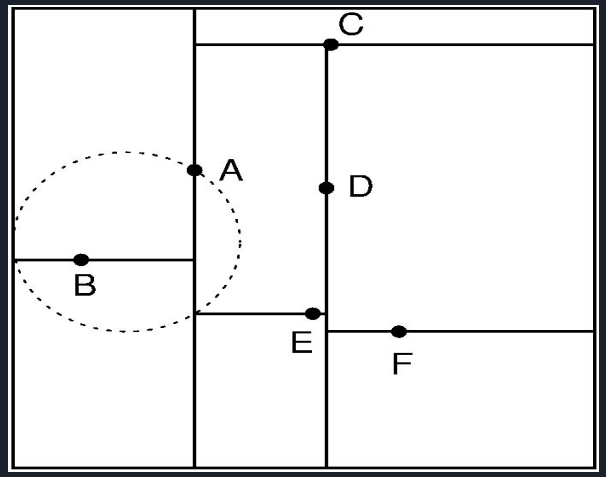
To accomplish this, we traverse the tree and compare the distance between the query point and the points in each leaf node. Starting at the root node, we recursively move down the tree until we reach a leaf node, following a similar process as when inserting a node. At each level, we decide whether to go down the left or right subtree based on which side of the splitting hyperplane the query point lies.

In first example

Find the points within 25 units of the point (25, 65)

One thing to note is that the distance metric in our example is the classical euclidian distance defined as:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$



Time complexity

The time complexity of building a K-D tree from a set of n points in k-dimensional space depends on the median finding algorithm used.

If a more efficient median-of-medians algorithm is used the total complexity is , on average. This is because, at each level of the tree, a new hyperplane is chosen to split the set of points, which takes time. Since the tree is binary, the number of levels in the tree is . Therefore, the total time complexity of building the *K*-D tree is . However, in the worst case, the construction time can be , which can occur if the tree is very unbalanced.

Application

K-D trees have a wide range of applications. As already mentioned, one of the most common uses is nearest neighbor search, where they’re used to find the closest point to a given point quickly. This makes them a useful component of many machine learning approaches like k-nearest neighbors(KNN) and a wide range of distance-based clustering algorithms.

Advantages and Disadvantages

One of the primary advantages of K-D trees is that they allow for efficient k-nearest neighbor (KNN) queries, which are useful in applications such as image recognition and recommendation systems. They can also be used for range queries, which allow for fast searches of points within a given radius or bounding box.

One of the most significant limitations is that they can become inefficient in higher dimensions, due to a phenomenon called the “curse of dimensionality.” This occurs because as the number of dimensions increases, the volume of the space between points increases exponentially, making it difficult to capture the geometric relationships between points accurately.

Resource and References

https://opendsa-server.cs.vt.edu/ODSA/Books/CS3/html/KDtree.html

https://medium.com/smucs/a-look-into-k-dimensional-trees-290ec69dffe9

https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/kdtrees.pdf

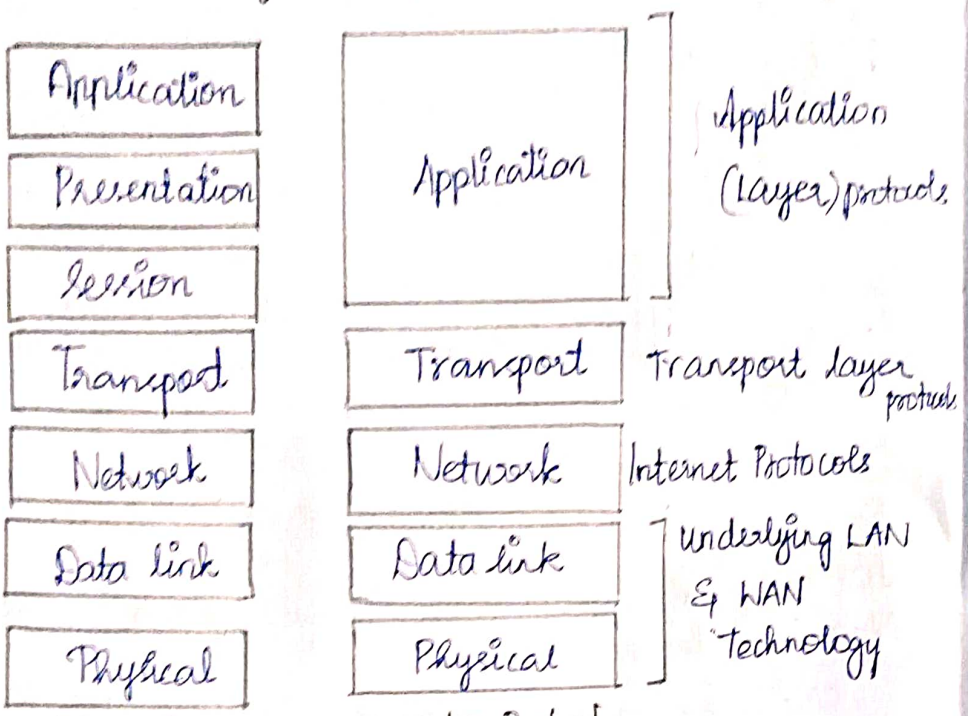
https://www.baeldung.com/cs/k-d-trees

Thank You

Tutorial

Computer Networks.

1. With a diagram TCP layers to OSI models include some of their protocols:



TCP/IP protocols:

Physical layer:

↳ TCP/IP does not have any specific protocol for physical layer.

↳ It supports all standard & proprietary protocols

Data link protocols:

↳ They also dont have any specific protocols.

It supports all standard & proprietary protocols

Network layer:

↳ TCP/IP supports Internet Protocol (IP). The IP is the transmission mechanism.

Transport layer:

↳ Transport layer was traditionally represented in TCP/IP by 2 protocols. They are User Datagram

Protocol (UDP) and Transmission Control Protocol (TCP). Stream Control Transmission Protocol (SCTP) was also introduced in last few years.

Application Layer

↳ This layer uses many protocols to provide services such as file transfer, mail transfer, etc--,

↳ Simple Mail Tranefer Protocol (SMTP), Multimedia Internet Mail Extension (MIME), File transfer Protocol (FTP), etc--.

2) Loop back Address:

A packet with this address won't reach the destination. The advantage is to check whether the system works fine.

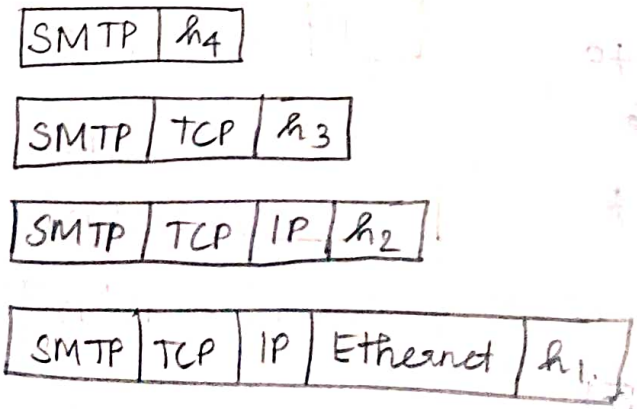
127.0.0.1?

No, the netId must be 127, hostId can be any.

3) Encapsulation:

The data part of a packet at level N is carrying the whole (part) packet from level N+1. This is called Encapsulation.

PDU's : TCP, SMTP, IP, Ethernet.



4) TIME-WAIT in 3-way Handshaking:

Significance & Time lost:

↳ When FIN segment is received, client sents an ACK segment & goes to the TIME-WAIT state.

↳ It sets a timer for a time-out value of twice the Maximum Segment Lifetime (MSL). MSL is the maximum time a segment can exist in Internet before it lost.

↳ MSL commonvalue is between 30 seconds & 1 minute.

Significance:

↳ If the last ACK segment is lost, the server which sets a timer for the last FIN, assumes that FIN is lost and resends it.

↳ If client goes to CLOSED state, before 2MSL time expires. The server cannot close the connection. The 2MSL timer makes the client wait for a duration that is enough time for an ACK to be lost & FIN to arrive.

↳ To prevent delayed packets from one connection being accepted by a later connection.

5) IP address 198.168.80.66/28.

a) Subnet Id

255.255.255.240
198.168.80.66

198.168.80.64

240 - 11110000
66 - 01000010
64 - 01000000

b) First Address

198.168.80.65.

c) Last Address

Not (Subnet Mask) (or) Actual gn address

0.0.0.15	00001111 => 15
198.168.80.66	01000010 => 66
198.168.80.79	01001111 => 79

198.168.80.79

d) Broadcast address

198.168.80.79

7) Count to Infinity problem :-

↳ It arises in Distance Vector Routing.

↳ For a routing protocol to work ~~protocol~~ properly, if a link is broken (cost becomes 'infinity'), every other router should be aware of it immediately but in distance vector routing, this takes some time. The problem is referred to as count to infinity.

Methods for avoiding count to Infinity Problem:

↳ Split Horizon

↳ Split Horizon with Poison Reverse.

8) Two forms used in MIME:

↳ Base 64 transforms bytes into printable characters which can then be sent as ASCII characters or any other type of Data to printable characters and divides the binary data into 24 bit blocks and each block divided into 4 sections each of 6 bits,

↳ Quoted printables type is used when the data consists mostly of ASCII characters with a small

non ASCII. If character is ASCII it is sent as such, if not first character is =, followed by 2 characters which are the hexadecimal rep of bytes.

If you base64 encode 60 bytes, the result will depend on the implementation of base64 encoding algorithm is used.

Approximately $(60 * 4/3) = 80$ bytes

10) Resolution in DNS

Mapping a name to an address or an address to a name.

Recursive & Iterative Resolution

↳ When client requests a local server, if it has right information it sents back the IP address, or it passes the request to its parent server & process continues until the query is resolved & sent back to client in same path is called Recursive resolution

↳ If client sents request to a local server, if it don't have result it sents the IP address of server which it think's it can resolve.

↳ And process continues until the query is resolved.

9) Replacing Hub with switch

It can significantly reduce collision.

Collision Domain:

In Hub, collision is felt in the whole network whereas in switch collision is felt,

only in that particular segment.

Broadcast Domain:

Both in Hub & switch broadcast takes place all over the network.

11) Three Different type of Addresses

Physical Address / Link Address / MAC address

Also hardware address

↳ It is the lowest level address.

↳ It is of 48 bits

↳ This can be unicast, multicast or broadcast

↳ Used in data link layer.

Network / IP address

↳ Used to identify each device connected to the Internet.

↳ It is of 32 bits

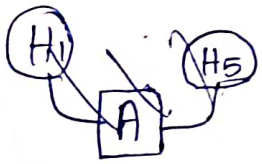
↳ They are unique & universal.

Port address:

↳ It is of 16 bit address.

↳ Used in transport layer.

↳ Service point addressing uses port address

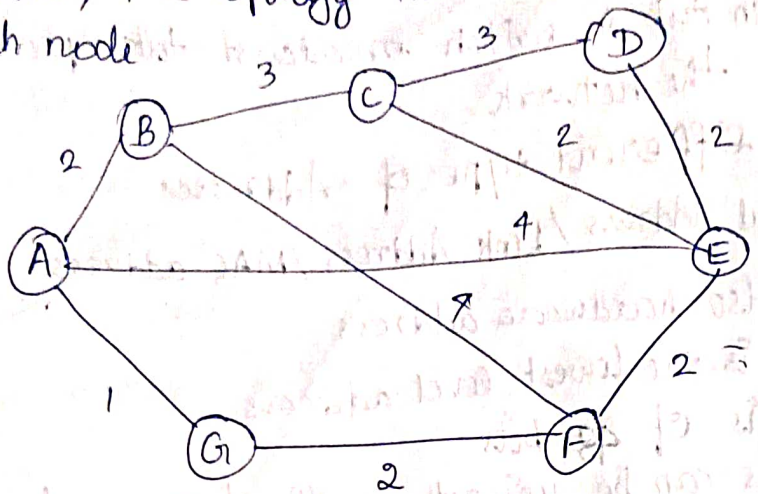


Same network from one host to another host. Through 2 routers IP & MAC address are enough. to transfer b/w networks (remote)

12) Link state routing overcomes shortco!

In Link state routing if each node in the domain has the entire topology of the domain how they are connected including type, cost & condition of the links the node can use the Dijakstra's algorithm.

If there are any changes in any point in the network, the topology must be updated for each node.



Destin

A (A,0)	(B, 2), (G, 1)
G(A,1)	(B, 2), (F, 3)
B(A,2)	
(A,0)	(B,2), (G,1)
(G,1)	(B,2), (F,3)
(B,2)	(C,5), (F,3)
(F,3)	(C,5), (E,5)
(C,5)	(E,5), (D,8)
(E,4)	(D,6)
(D,6)	

~~13) a) Routing Fragmentation.~~

14) PSEUDOHEADER:

Pseudoheader consists of the source IP, destination IP, protocol & total length fields. This is included at the TCP checksum to

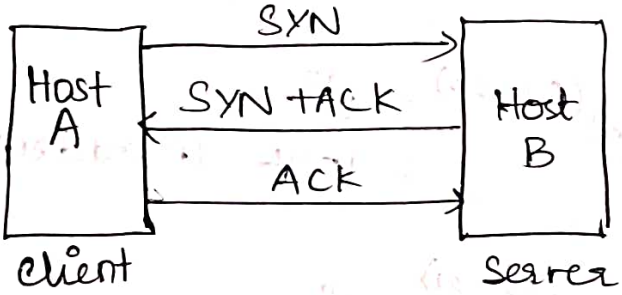
verify at the network and Transport layer the data is delivered to the correct host Pseudoheader comes from network layer. For checksum, TCP should get information from network layer but technically it violates the rule of layering. But they provides additional security.

Checksum in TCP:

→ The checksum field in TCP header is set to 0. The binary values of source IP, destination IP, protocol, length all are concatenated. The resulting binary value is treated as a large integer & bit-wise ones complement is taken. The value is converted into 16 bit binary & placed in the checksum field of TCP header.

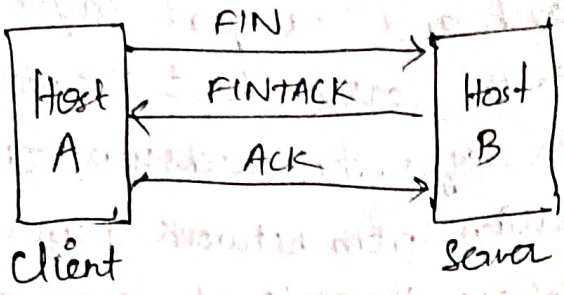
CONNECTION ESTABLISHMENT:

↳ Done using three-way handshaking
↳ Client sends an ACK to the server which in turn sends SYN+ACK & the when the client receives it, it sends back an ACK. This establishes a connection b/w client & server



CONNECTION TERMINATION:

↳ Done using 3-way handshaking
↳ Client sends FIN & in turn server sends FIN+ACK & the client sends an ACK



15) When a new Ethernet card is installed, its needs to be configured with the correct MAC address. This is done using the Address Resolution Protocol which broadcasts an ARP request on the network, asking for the MAC address of a particular IP address. If a device on the network has that IP address, it will respond with its MAC. The ARP request and response are both encapsulated in the Ethernet frames. In the given case, its possible that the ARP cache on your system or on other devices on the network still contained the old MAC address for the systems IP address. This could have made the packets to be sent to the old Ethernet card. Once the ARP cache entries expected, the ARP would be used again to resolve with new MAC address with which the request could be answered, allowing packets to be correctly delivered to the system.

15) First Network:

$202.78.32.0/20 = 2^{(32-20)} = 2^{12}$

$= 4096$ IP address

Network 2

$202.78.64.0/21 = 2^{(32-21)} = 2^{11}$

$= 2048$ IP address

To allocate a new network of atleast 512 IP address, the smallest network that gives more than 512 IP's is $32-23 = 9$; $2^9 = 512$ i.e. /23

202.78.74.0/23 which is within the range of

202.78.32.0 - 202.78.75.255

ii) There should be a minimum of 2 entries in company C's routing table which are needed to guarantee packets were correctly delivered to both the companies:

Company A: Net ID : 202.78.32.0/20
Gateway : 202.78.64.23
Netmask: 255.255.240.0
Interface: Eth1

Company B: Net ID : 202.78.48.0/22
Gateway : 202.78.48.11
Netmask : 255.255.252.0
Interface : Eth1.

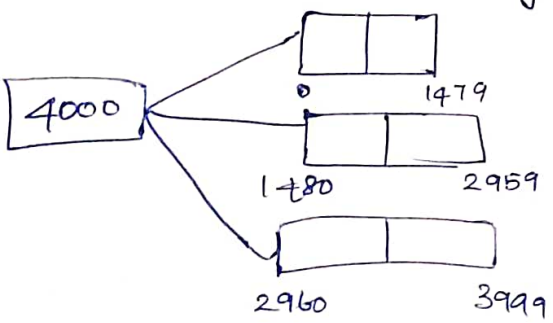
13) Datagram size : 3980 IEEE 802.3

Original datagram has no options = Header = 20.

Total size = 3980 + 20 = 4000

MTU = 1500 (1500 - 20 = 1480)

Total size > MTU ⇒ Fragmentation Necessary



	offset
	0
	185
	370

FlagId	FlagMF	Offset
100	1	0
100	1	185
100	0	370

6) If 2 routers are misconfigured, to form a routing loop for a destination X, it means that both routers believe they have the shortest path to reach X and will forward packets destined to X to each other in a loop. But when each router forwards the packet, it decreases the TTL value by 1. When the TTL value reaches 0, the packet is discarded and an message is sent back to the source indicating that the packet was discarded due to its TTL value being exceeded. So eventhough the routers are in a loop & forwarding the packet back & forth to each other the TTL value will eventually reach 0, the packet will be discarded by one of routers. The message will be sent back to source & source will stop sending packets to X. This will break the loop & prevent packets from going back & fourth forever.

OPERATING SYSTEMS TUTORIAL.

Jenisa Merlin. D
21PW08

FILE:

File is a named collection of related information that is recorded on secondary storage.

File is smallest allotment of logical secondary storage

Data cannot be written in secondary storage unless they are written in file.

FILE ATTIBUTES:

* Name: only info kept in human readable form.
* Type: needed for systems that support different types.
* Location: pointer to file location on device.
* Size: current file size.
* Protection: controls who can read, write & execute.
* Time, date and user identification: data for protection, security and usuage monitoring.

2) DATA STRUCTURES RELATED TO FILES.

OS maintains 2 data structures representing state of open files.

a) System wide open file table.

Contain a copy of FCB for every currently open file in the system.

When operation is requested, the file is specified via an index into table.

If file is no longer needed, then its entry is removed from table.

b) Per process table:

Contains a pointer to the system open file table as well as some other info.

all files that process has open.

Eg: current fp for each file is found here.

access rights and accounting info is also included.

DIFFERENT TYPES OF FILE ACCESS METHOD:

* Sequential access.

Simplest access method.

info is processed in order one record after other.

Mode of access is by the far most common.

Used in editor, compiler.

Access method is slow.

* Direct access.

Relative access method.

Fixed length logical record allows program to read and write record rapidly in no particular order.

Based on disk model of file since disk allows random access to any file block.

Decrease access time.

No need of traversing all blocks.

* Index sequential

Access using an index points to various blocks using this access file directly.

Top of sequential access

3) SINGLE LEVEL DIRECTORY:

Single directory for all users.

Naming problem

Grouping problem.

Easy implementation

TWO LEVEL DIRECTORY:

Separate directory for each user.

Better organization and reduces naming conflicts.

Become complicated when sharing files between users.

TREE STRUCTURED DIRECTORY:

Files organized in hierchial structure.

Each directory can have multiple sub-directories and files are stored.

Easy to navigate.

Efficient organization of files.

Enables access control.

ACYCLIC GRAPH DIRECTORY:

Similar to tree structured directory but allows creation of multiple links blw directories.

Improve organization and easier to navigate larger directories.

Lead to complexity and confusion if not managed properly.

GENERAL GRAPH DIRECTORY:

Most complex directory structure.

Allows multiple links between directories including loops which create more complex relationship blw directories.

Advantage to some application but challenge in manage and naviagate.

Tree structured directory is most advantageous.

4) Increasing the degree of multiprogramming can improve CPU utilization bcoz it allows multiple programs to run simultaneously on CPU.

When there are multiple programs waiting for I/o CPU can switch to another program which reduce idle time of CPU and improves uti°liza°tion.

6) Assume that head is at cylinder 500.

Total seek time,

FIFO ⇒ 548-500+123-548+459-123+156-459+
645-156+818-645+825-818+843-825+
239-843 = 2665.

SCAN ⇒ (548-500)+(825-548)+(843-825)+
(818-843)+(645-818)+(459-645)+
(239-459)+(156-239)+(123-156) = 1841.

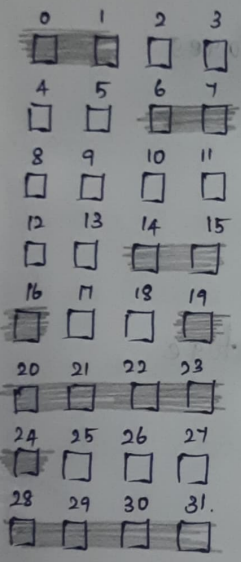
C-SCAN ⇒ (548-500)+(825-548)+(843-825)+
(1000-843)+(239-0)+(459-239)+
(645-459)+(818-645)+(156-818)+
(123-156) = 3058.

LOOK ⇒ (548-500) + (825-548) + (843-825) +
(818-843) + (645-818) + (459-645) +
(239-459) + (156-239) + (123-156) = 1886.

5) FILE ALLOCATION IN DISK BLOCKS.

* Contiguous allocation.

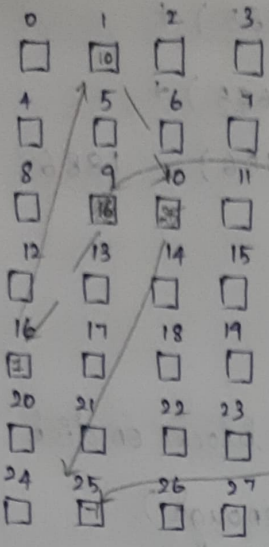
Each file occupies set of contiguous blocks on disk.
Simple (starting location & length is enough.)
Random access.
Wasteful of space.
Files cannot grow.



file	start	length
A	0	2
B	14	3
C	99	6
D	28	4.

* Linked allocation.

Files are stored as linked list of disk blocks.
file has a pointer to first block of file
and each block contains pointer to next block in file.
last block has null pointer (EOF).
Easily resized.
Result in fragmentation
Slower file access.



* Indexed allocation.

Files stored using index block that containers pointers to each block in the file.

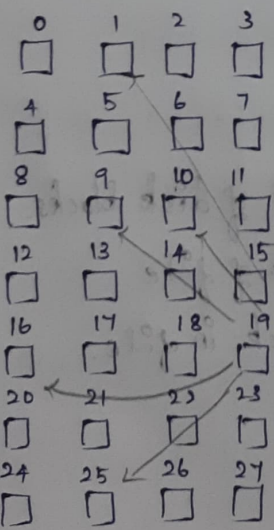
Each file has a index block that stores pointers to each block in file.

Fast file access time.

Easily resized.

Requires additional disk space to store index block.

Also result in fragmentation.



9) LOCALITY OF REFERENCE:

Tendency of computer programs to access data that is close to other data that has been accessed recently.

2 types: temporal locality
spatial locality.

Optimize performance by reducing no. of expensive operations such as cache miss & disk access.

It is achieved by storing frequently accessed data in cache or main memory or prefetching and buffering to minimize latency.

It is not possible in general to implement optimal page replacement algorithm.

Since it requires knowledge of future page ref which is not available at time of compiling or executing program.

However, it is possible to approximate behaviour of program and use info to generate list of page ref that may be close to optimal for given program & input.

10) CONTIGUOUS ALLOCATION:

Suffers internal fragmentation if initial file size allocation is too big.

Suffers from external fragmentation

Inefficient for random access.

Typically allocates a single file to single cylinder.

LINKED ALLOCATION:

Space wastage in index block.

Inefficient for random access

Suffer from unreliability bcoz single error can cause loss of many block of data.

INDEXED ALLOCATION:

Space wastage in index block.

Inefficient for random access.

Suffer from external fragmentation.

12) INTERPROCESS COMMUNICATION:

Allow processes to exchange info and coordinate activities in system.

1) Shared memory.

Fast & efficient way for processes to communicate with each other.

A region of memory is set up that can be accessed by multiple processes.

Process read from, write to allow them to share info.

It is speed, difficult to implement correctly requires synchronization mechanisms.

2) Message passing.

Process communicate by sending msg to each other.

2 types: Synchronous

Asynchronous.

3) Pipes:

simple form in which 2 process communicate by writing and reading from common buffer.

pipe - unidirectional communication channel.

parent - child or peer - peer relationship.

Used in simple communication not very efficient for large amounts of data.

4) Sockets.

More powerful form used across network.

communicate using TCP/IP or UDP protocols

Client - server or peer - peer communication.

5) Signals.

Used to notify process of an event or interrupt.

Interrupt delivered to process by OS.

Used to handle errors, communicate with other processes, manage process synchronization.

13) physical memory size = 2^{32} bytes.

Page size = 2^{16} bytes.

Logical address space = 2^{16} pages.

1) Logical address space = $2^{16} \times 2^{16} = 2^{32}$.

32 bits long.

2) Page size 2^{16} bytes. Each frame is also 2^{16} bytes long.

3) No. of frames = $\frac{\text{Physical memory size}}{\text{Page size}} = \frac{2^{32}}{2^{16}} = 2^{16}$.

Frame number specified using 16 bits.

4) No. of entries = No. of pages.

∴ No. of entries in page table = 2^{16}.

5) Each pagetable entry requires atleast 17 bits (16 → frame numbers, 1 → valid/invalid) and additional bits depending on system's design.

14)

Process	Arrival time	CPU Burst time	Memory required	Execution Start	Execution End	Turn Around time
P_1	0	5	3	11	16	16-0=16
P_2	3	3	5	17	20	20-3=17
P_3	7	9	8	21	30	30-7=23
P_4	12	10	12	31	41	41-12=29
P_5	18	16	2	42	58	58-18=40
P_6	25	2	6	59	61	61-25=36
P_7	29	8	9	62	70	70-29=41

15) SEMAPHORE:

Synchronization tool that is used to manage access to shared resource in concurrent system.

Ensure that multiple processes can access shared resource without interfering each other.

a) ~~Only one~~ one D is printed, Infinite D is printed.

b) Smallest number is zero bcoz it has to wait untill process 1 signals R, atleast once, before it prints A.

c) ~~Yes~~ No CABABDD CABCABD not a possible output sequence.

d) No. CABACDBCABDD not possible output sequence.

L=3 R=0.

Process 1.	Process 2	Process 3.
L1: wait(L)	L2: wait(R)	L3: wait(R)
putc('C')	putc('A')	putc('D')
signal(R)	putc('B')	goto L3
goto L1	signal(R)	
	goto L2	

L=2 print C R=0 print A, B R=0 print D.

R=1 R=1

L=1 print C R=0 print A, B R=0 print D.

R=1 R=1

c) No.

d) No.

8) Contiguous Allocation.

F1, 11 logical records of 112 bytes $11 \times 112 = 1232$ bytes.

Disk 30 blocks of 1024 bytes, we need to allocate 2 consecutive blocks (i.e) 2048 bytes.

F2, 890 logical records of 13 bytes $890 \times 13 = 11570$ bytes

Allocate 12 consecutive blocks (i.e) 12288 bytes.

F3, 510 binary data in single block.

F4, 4 logical blocks of 95 bytes each require 380 bytes.

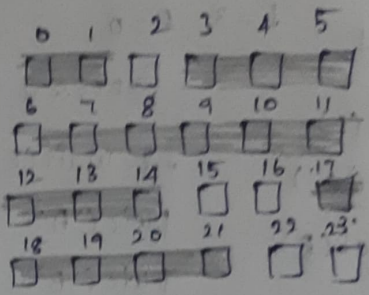
Allocate 1 block for this file.

F1 → 0, 1

F2 → 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

F3 → 14

F4 → 15.



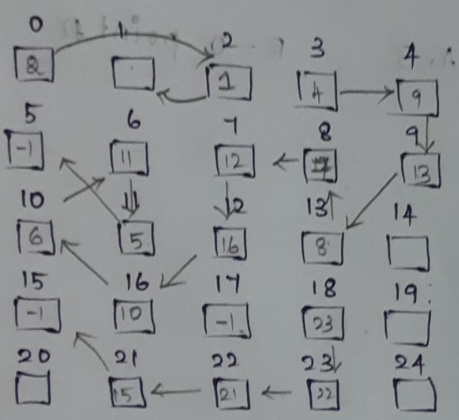
Linked allocation:

F1 → allocate 2 blocks.

F2 → " 12 blocks

F3 → " 1 block

F4 → " 4 blocks



File	Start	~~End~~ Length
F1	0	2
F2	3	12
F3	17	1
F4	18	4

Indexed allocation.

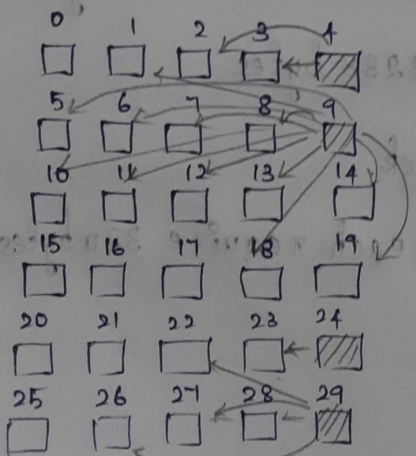
F1 → 2 blocks

F2 → 12 blocks

F3 → 1 block

F4 → 4 blocks.

File	Index Block.
F1	4
F2	9
F3	24
F4	29.



4 → 3, 2

9 → 8, 13, 18, 7, 12, 6, 11, 5, 10, 14

24 → 23

29 → 28, 27, 22

1) Contiguous allocation.

Each file occupies contiguous set of blocks on disk.

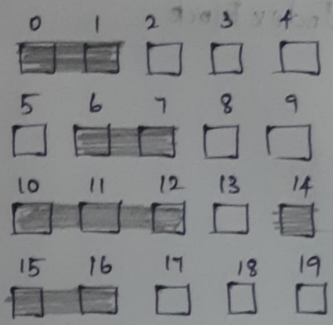
Starting address & length is required.

Support sequential and direct access.

Extremely fast.

Inefficient bcoz of internal and external fragmentation.

Difficult when file size increases.



File	start	length
A	0	2
B	14	3
C	6	2
D	10	3

Linked allocation.

Each file, linked list of disk blocks, need not be contiguous.

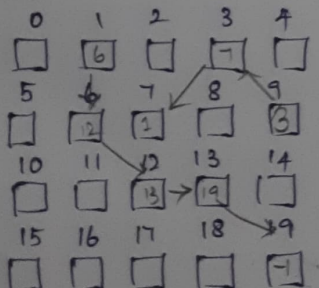
Directory entry with pointer to starting and ending file block.

Flexible in terms of file size.

No external fragmentation.

Larger no. of seeks - slower

No random/direct access support.



File	start	End
A	9	19

Indexed allocation.

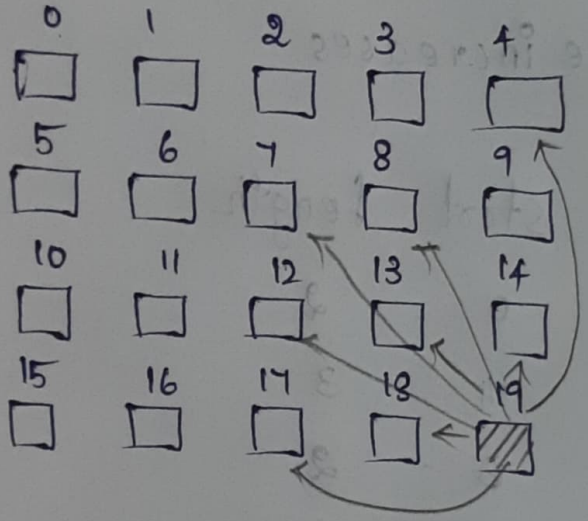
Special block, index block contains pointers to all the blocks occupied by a file.

Support direct access to blocks occupied by file

No external fragmentation.

For very small files, one entire block usuage.

For large files, it may be insufficient.



File A

Index Block

19

4
8
12
7
13
14
18
17