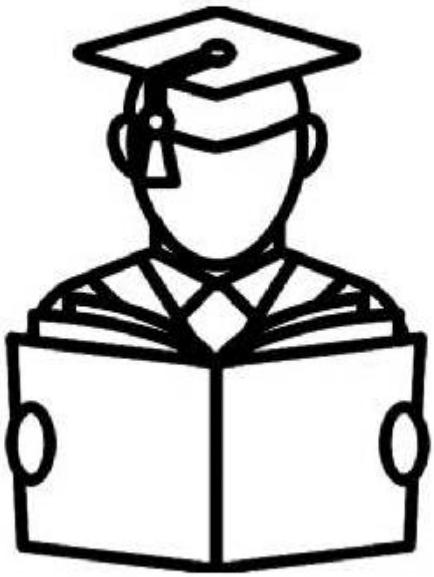
चौधरी PHOTOSTAT

"I don't love studying. I hate studying. I like learning. Learning is beautiful."



"An investment in knowledge pays the best interest."

Hi, My Name is

Computer Science & IT Engineering for GATE/IES

(MADE EASY)

Operating System

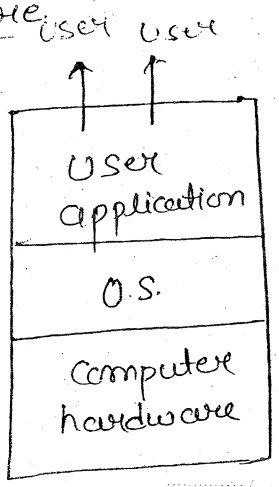
Teaching Schedule

1. Introduction & Background
2. Process Management
   - → process concept
   - → CPU Scheduling] most important
   - → Synchronization] very very important] 40%
   - → Concurrent programming
   - → Deadlocks
   - → Threads
3. Memory Management
   - → RAM chip Implementation
   - → loading, linking & Address Binding
   - → Techniques (40%)
     - → paging
     - → multilevel paging
     - → Inverted paging
     - → Segmentation
     - ⇒ Segmented paging
   - → virtual memory
4. File & Device Management
5. Protection & Security (no question will come from this section)

Introduction:

Definition of Operating System :-

→ Operating system is an interface b/w the user and the computer hardware.



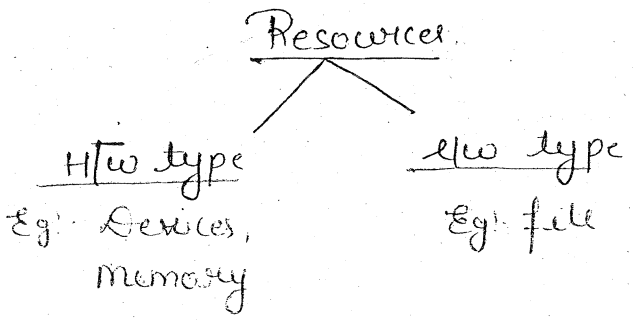

```
main()
{
int x;
printf("Hello world"); (interact with monitor)
scanf("%d", x); (interact with keyboard)
}
```

printf → Internally calls the write() system call to communicate with the monitor.

System call: System call is, it is a request made by the user program to the operating system in order to get any kind of service.

→ The operating system can also be called as Resource allocator.

Operating system is responsible for allocating resources of your computer.



Goals of Operating System:

1. primary goal of O.S. is convinience. (Easy to use)
2. Secondary goal of O.S. is efficiency.

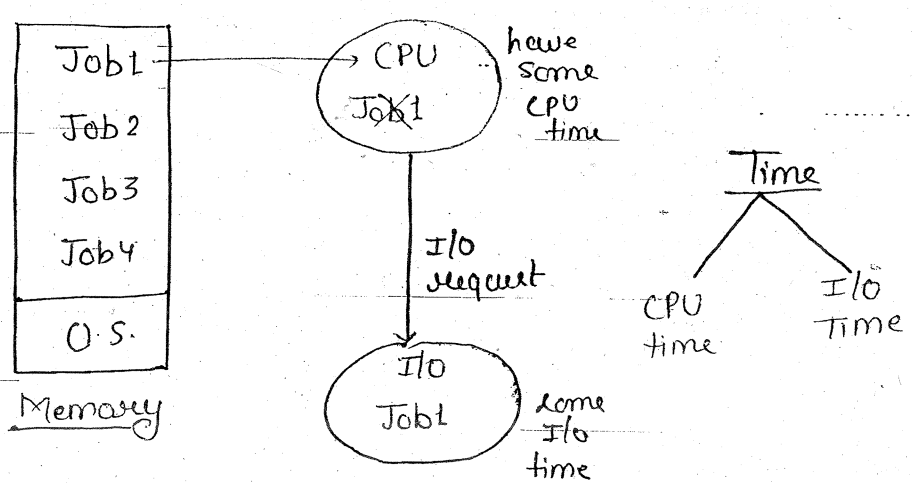
> windows O.S is more convinient than UNIX O.S.

> UNIX O.S. is more efficient than window O.S.

Generally, for large database, which are used throughout the year without rebooting the computer, we should use UNIX O.S.

Types of Operating System:

1. The Batch OS



→ If the job is Completed comptely, then only other another job will be scheduled on to the CPU.

(means all (CPU + I/O) time is to be completed)

→ Increased CPU idleness

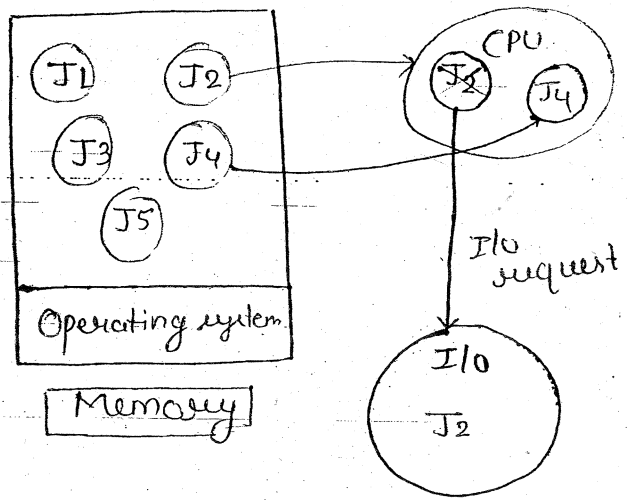
→ Decreased throughput of the system.

(the no of jobs completed for unit time is called as the throughput of the system)

CPU become idle- when job left CPU after I/O request & go to I/o device CPU become idle

→ IBM OS/2

2. Multiprogramming Operating System



* After epending some CPU time on CPU, Job J_2 left CPU & go for some I/o operation & epend I/o time on I/O device.

* When J_2 left CPU, CPU will not become idle.

Note:

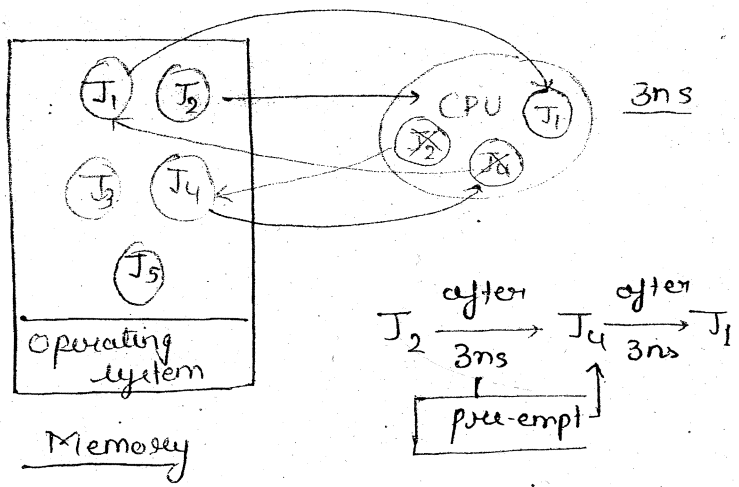
If the job is leaving the CPU to perform I/O operation then another job, which is ready for execution will be scheduled on to the CPU.

→ Increased CPU Utilization.

→ Increased throughput of system.

Eg:- Windows, UNIX etc.

3. Multitasking Operating System:



→ The multitasking OS is an extension to multiprogramming OS.

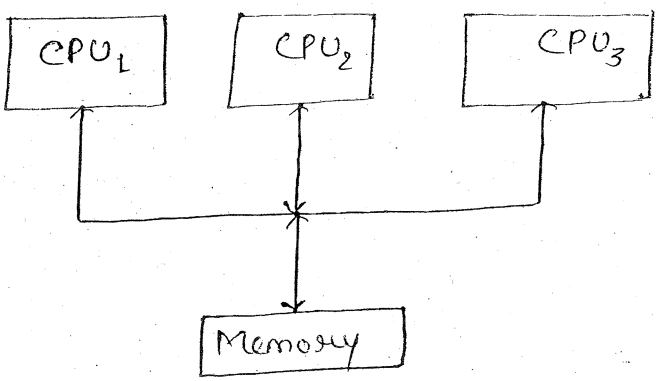
→ The jobs will be executed on the CPU in the time-sharing mode.

Eg:- Windows, UNIX etc.

→ Only one job is scheduled on CPU at a time.

→ For user point of view all happens at a time.

4. Multiprocessor Operating System (Parallel system)
(more than one cpu connected to the system)



Advantages:

→ The increased throughput of the system.
(no. of jobs completed in unit time is increased).

→ Reliability (if one CPU is fail, then other system will still run with the help of remaining CPUs)
(fault tolerant system)
fault will be accepted

→ economical (cost will be less) — three system with three CPU - more costly than - one system with three CPU.

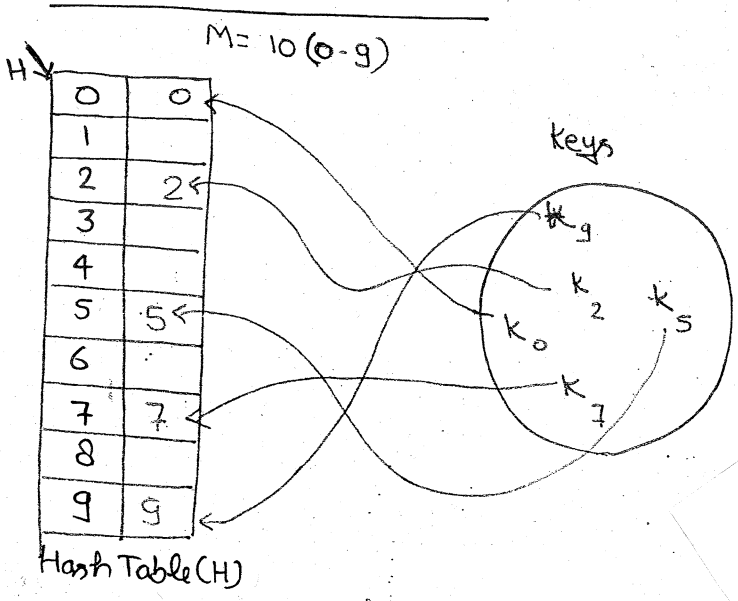
→ You can't make one CPU is busy always & make other CPU's idle always. Jobs are distributed over the no. of CPU's.

→ Eg:- UNIX can be used as Multiprocessor OS.

Hashing

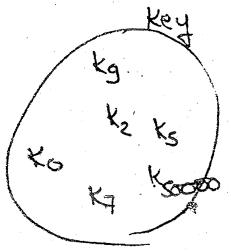
1) Searching Technic
2) Goal of Hashing is O(1)

Direct Address Table (DAT)



		Searching Time
If we want to find 7	then just go to H[7]	O(1)
" " 0	" " H[0]	

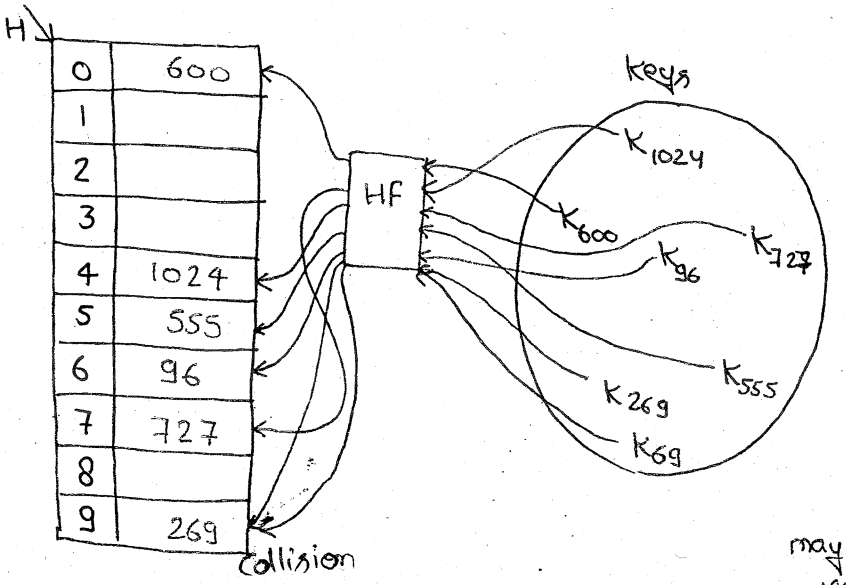
i) Order (1) searching time
ii) key is the address without any calculation
iii) Even though no. of keys are very less but one of the key may contain 64 bits then Hash table of size 2^{64} required



to store only 6 keys we need approx. 50000 size of Hash Table
there is a huge gap b/w the keys.

Hash Function

M = 10 (0-9)



may be two bigger keys will go to same place.

Two different keys has to same slot is called collision.

if we want to find 269 then 269 mod 10 = 9, go to H[9] Searching time O(1)

if we " — " 96 " — 96 mod 10 = 6, go to H[6]

Types of Hash Functions

1.) Division - Modulo Method
2.) Digit Extraction Method
3.) Mid- Square Method
4.) Folding Method
   i.) Fold Shifting Method
   ii.) Fold - Boundry Method

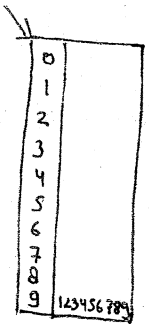
ii) Division - Modulo Method -

M = 10 (0 - 9)

HF (key) = key mod M

key = 123456789

HF(123456789) = 9



ex. M = 8 (2^3)

10101101010101 mod 2^3 = 101

000000010111 101 mod 2^3 = 101

111111111110 101 mod 2^3 = 101

Collision!

ex. $M = 2^k$

11001100 10101010101010 mod 2^k = 101010 1

k Bits — LSB - k bits

Note → Don't choose M values exactly power of 2 (2^k) becoz if $M = 2^k$ Hash func^n. of any key will contain LSB - k bits.

→ Choose M value which is a prime number and which is not too close to power of 2.

~~M = 17~~ M = 711 ✓

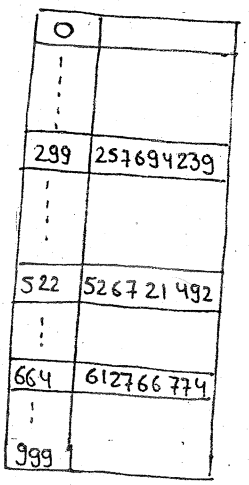
iii) Digit Extraction Method [Truncation Method]

M = 1000 (0 - 999)

keys = ①②③④⑤⑥⑦⑧⑨
5 2 6 7 2 1 4 9 2 = 522

2 5 7 6 9 4 2 3 9 = 299

6 1 2 7 6 6 7 7 4 = 664



we are extracting 3 digits only
∴ max 3 digit addresses we can handle ∴ max slot no is 99

Here we are taking 1st, 5th & 9th dig but we can take any three digit but at the time of retrieving we have to take same digits that we have taken at the time of insertion.

If 1st, 5th & 9th bits are same then collision occur ☹

∴ Only these digits are participating so chances of collision are more.

iii) Mid - Square Method

M = 1000 (0 - 999)

key = 8452

First Square it then take the mid

$(key)^2 = (8452)^2 = 71436304 = 71(436)304$

= 436 (or) 363

Here all digits are taking participation
But here also collision may occur but chances of collision are very less as compare to other.

iv) Folding Method

a.) Fold Shifting Method

M = 1000 (0 - 999).

key = (123)(456)(789)

max 3 digit address possible so take 3-3 digit from key & fold it and add it

```
  123
  456
  789
 ----
 1368
```

only in few cases overflow will come then further fold & add it untill we get the required answer

```
  136
    8
 ----
  144
```

Still collision are possible

Swapping (123)(456)(789) ⇒ 144

(456)(123)(789) ⇒ 144

(456)(789)(123) ⇒ 144

b) Fold-Boundry Method

M = 1000 (0-999)

Key = (123)456(789) take only boundry

123
789
912

→ Hash Function should take constant time.

The weakness of Hashing is, whatever may be the Hash Function there may exist a bad set of keys which may hash to same slot

Collision-Resolution Technique →

i) Chaining (outside)
ii) Open addressing (inside)
  a) Linear Probing
  b) Quadratic Probing
  c) Double Hashing

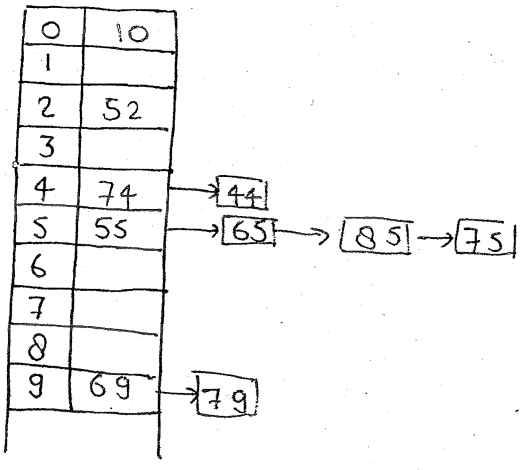
i) Chaining →

M = 10 (0-9)

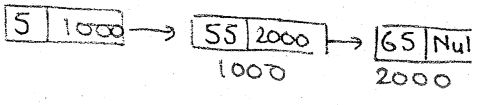
H(key) = key mod M

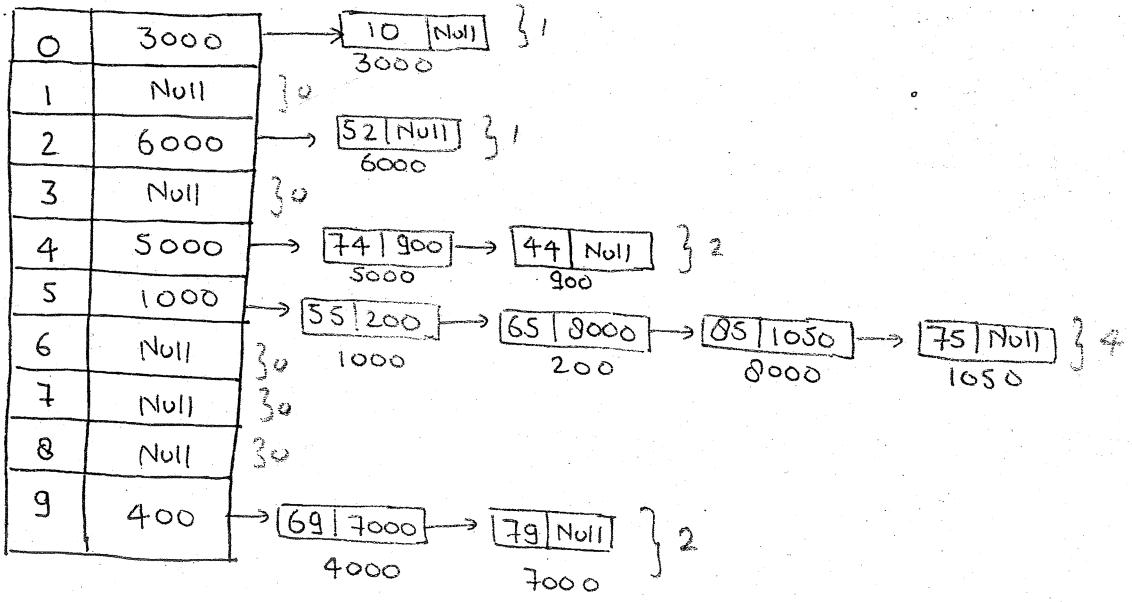
Keys → 10, 55, 69, 74, 52, 65, 79, 85, 44, 75

CRT = Chaining
(Collision Resolution Technique)



Actually linking is done by





Here longest chain length = 4
minimum chain length = 0

⇒ Length of the longest chain with n element is n (all elements hash to same slot).

⇒ Worst case time complexity to find an element is O(n)

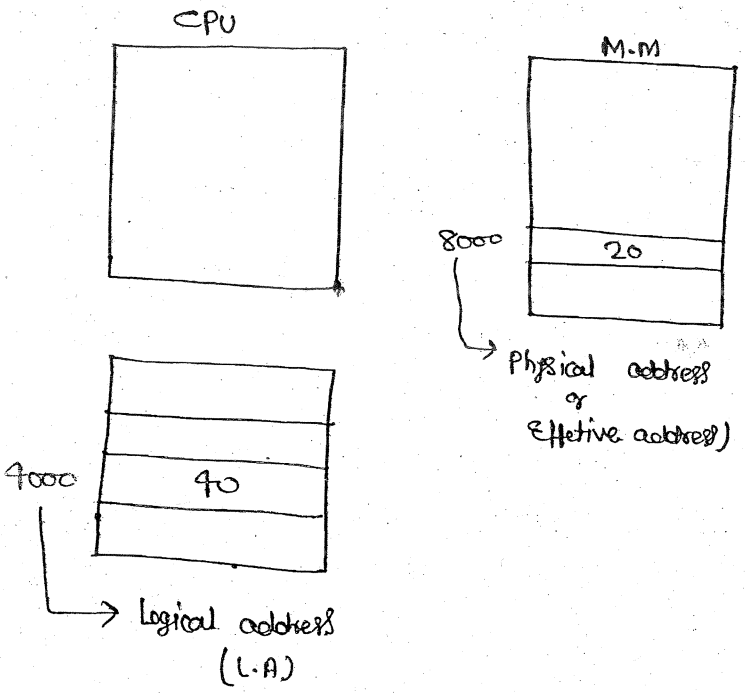
⇒ We are wasting lot of space in the form of link list even though space is available inside.

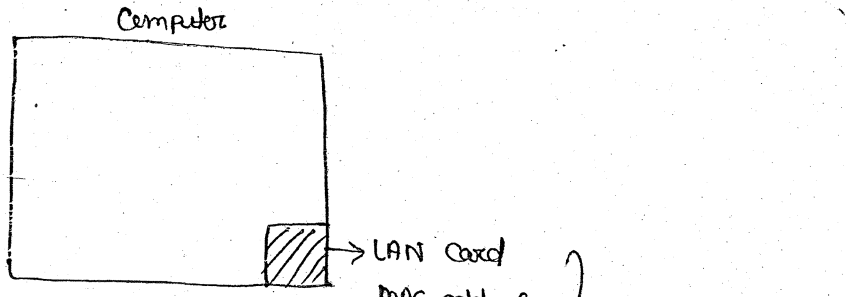
→ The advantange with this method is, it can handle unlimited no. of collision.

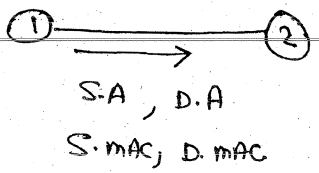
we can keep any no. of keys

→ Insertion, Deletion can be done at any where without affecting others.

Network







→ MAC address can't be used as an identification unit, in transmitting the data via Internet biz every company has their own way of representing the MAC address.

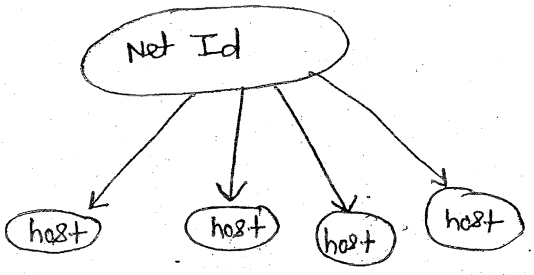
IANA → Internet Assigned Number Authority

↓

logical addressing system
(Classful addressing system)

IPv4 ⇒ 32 bit address.

A, B, C, D, E



Whenever the Computer is given an Ip Address it is treated as host.
Entire Network is will be represented by network ID (net ID).
The IP address can be represented by 2 notation:-

① Binary notation:-

01010111 10101111 10101010 11111000 → first few bits.

② Dotted Decimal Notation:-

63.73.126.120 → first octet.

Class A

(2^7-2) — 0 — Net ID, 8 bits

$(2^{24}-2)$ — Host ID, 24 bits

0 0000000 = 0
!
0 1111111 = 127
= (0-127)
= (1-126)

Unicasting →
0.0.0.0 ⇒ DHCP client
127.x.y.z ⇒ loop back address.

Class B

10 — 2^{14} — Net bits, 16 bits

$(2^{16}-2)$ — Host bit, 16-bits

10 000000 = 128
10 111111 = 191
(128 – 191)

Class C

110 — 2^{21} — Net bits, 24 bits

(2^8-2) — 8 bits, Host bits

110 00000 = 192
110 11111 = 223
(192-223)

- In class A Each Network will have $(2^{24}-2)$ Hosts.
- In class B Each Network will have $(2^{16}-2)$ Hosts.
- In class C Each network will have $(2^{8}-2)$ hosts

multi casting → Class D → 1110

1110 0000 = 224

1110 1111 = 239

(224 - 239)

Class E → 1111

1111 0000 = 240

1111 1111 = 255

(240 - 255)

- Class D & class E not support IP address so its not support Host address.

In binary notation first few bits decide the class type. in dotted decimal notation the first octet decide the class type

$IP_1 = 201.16.139.149$

net Id =

direct broadcast =

address of network =

Network mask
(Default mask)

- Network mask is a mathmatical tool which is used to solving some of the networking problem.

Class A ⇒ 11111111 00000000 00000000 00000000

255 . 0 . 0 . 0 — mask of class A

Class B ⇒ 255.255.0.0

class C ⇒ 255.255.255.0

class D ⇒ X

For a N/w mask n/w bits will be 1, host bits will be 0.

Q) IP_1 = 201.16.139.149
↳ Class C

mask = 255.255.255.0

NetId = 201.16.139.0

Performing bitwise ~~AND~~ b/w IP_1 & mask will give Network ID.

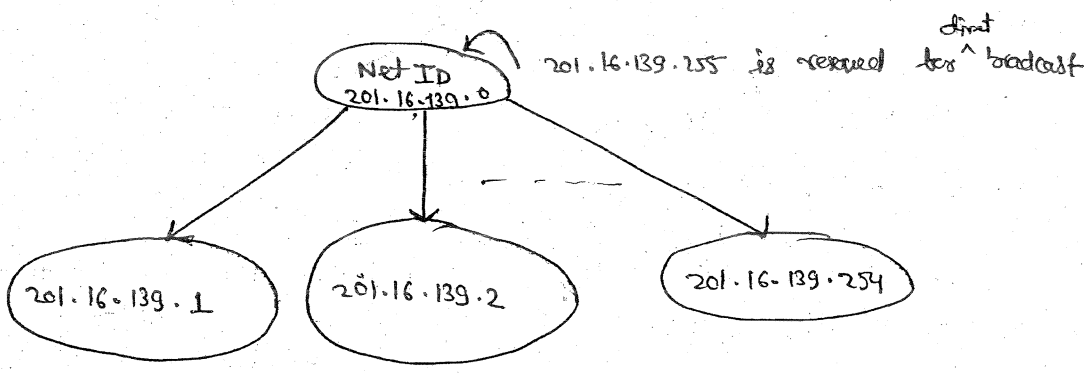
AND	
0	0
0	1
1	0
1	1

149 ⇒ 10010101
0 ⇒ 00000000
10010101
00000000

149 → 10010101
0 → 00000000 ~ bitwise AND
00000000

139 → 10001011
255 → 11111111
10001011

Net ID = 201.16.139.0



Direct broadcast :- 201.16.139.255

→ we have subtracted 2 in the host Portion bcz 1 is used as N/w ID and other is used as direct broadcast address of network.

IP = 144.16.19.159 → class B

Net ID = 144.16.0.0

→ For a nw ID the host bits will always be zero.

Direct bradcast = 144.16.255.255

→ For a Direct broadcast address the host bits will always be All one(1).

IP = 37.159.87.120

netId = 37.0.0.0

IP address

Private IP address

(i) Connecting the LAN (In LAN system are given)

(ii) Free of cost

(iii) Private IP address Range.

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

(iv) Scope is Local

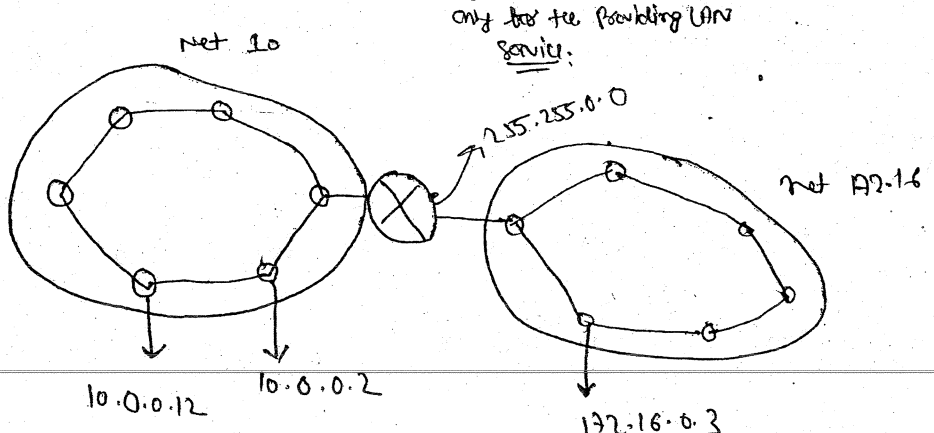
(v) Can't get Internet service only for the Building LAN service.

Public IP Address.

(i) Internet Service

All Public IP Add are the control of IANA.

they provide the service ISP = ISP1, ISP2 – –



I

	S.I.P	D.I.P
D	10.0.0.2	172.16.0.3

It's a unicast packet from one n/w to another n/w

II

	S.I.P	D.I.P
D	10.0.0.2	172.16.255.255

→ direct broadcast address

	S.IP	D.I.P
D	172.16.255.255	10.0.0.2

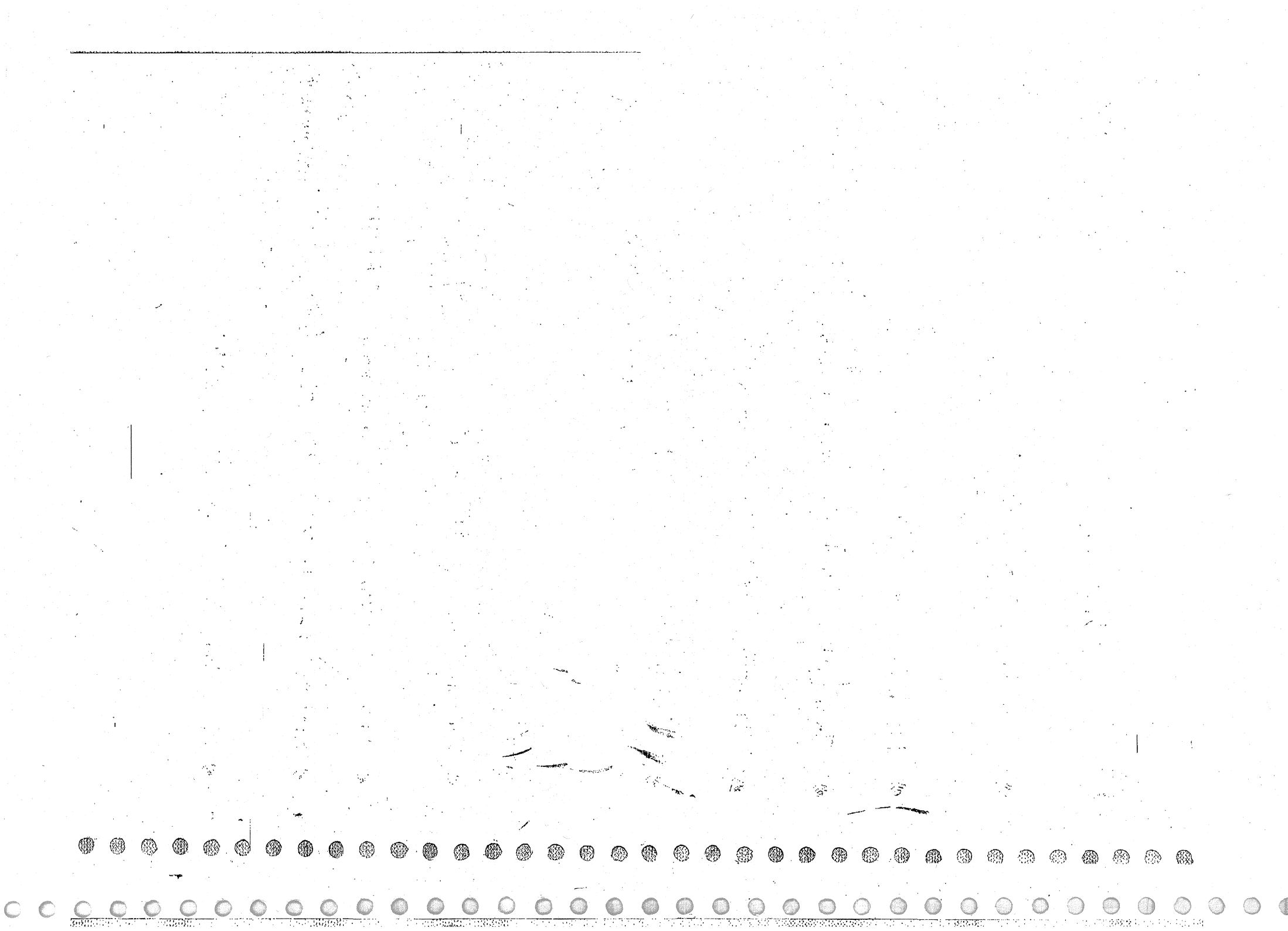
All to 1 so not possible

CONTENTS:

- Introduction and ER Model, Integrity Constraints — 2
- Normalization (Schema refinement) (10-12) — 2+2
- Transaction and Concurrency Control (12-15) — 2
- Indexing and File Organization (6-8) — 2
- Query language → SQL (10), Relational Algebra, Tuple Relational Calculus — 2+2

Textbooks:

- Fundamentals of DB Concepts — KORTH
- DBMS: NAVATHE (Transaction + Indexing)
- * DBMS: RAGHURAMA KRISHISHNAN (Questions → Query lang*, Normalization) ⇓ (Ex-soln-net)
- * The Complete DB Concepts: ULLMAN



INTRODUCTION →

DATABASE :- Collection of interrelated data.

DBMS :- (S/W) used to manage the database and access the DB efficiently. (File System + Operating System)

efficiency decrease as size of DB increase

Limitations of File System:

(1)

(University Database) → F1, F2, F3 ... FN (500 GB)

?? Retrieve data from DB i.e students who scored more than 80%.

⇓

Alternatives

- Program
- Manually

- an appl^n progrmr should have knowledge about

Physical details (low-level details)
- Fname
- Location of file
- Format
- Permissions

→

» Accessing data using physical level details too complex.

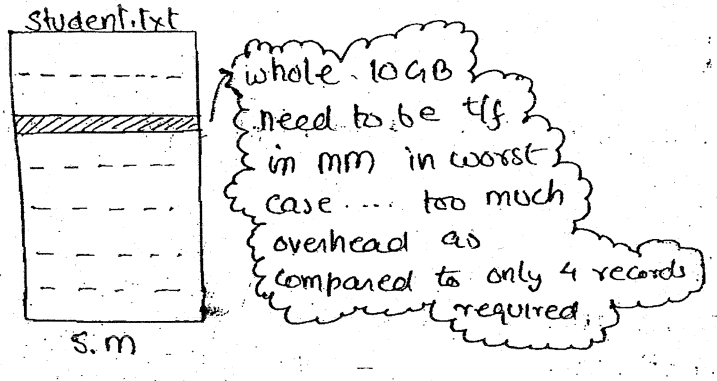
⇒ Database MS (DBMS) provide "Data Independency" i.e user can use the data without knowing any physical details.

- It is achieved by using querying language.

```
Select * from Student where marks > 80%
```

(2)

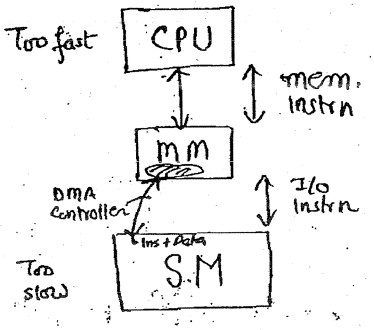
500 GB | Student.txt (10 GB) (Disk) ?? (Student whose roll no (21 to 25)) (Just a small access in 10 GB DB.)



(1 Human Architecture)

Stored Program

Ins + Data that is required to execute by CPU should be stored in Highspeed MM



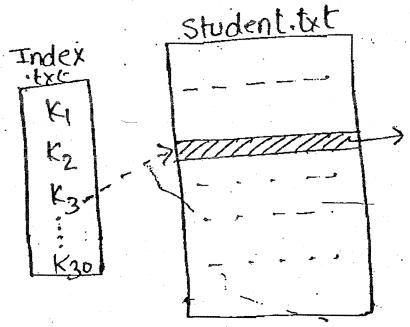
» If DB file is too huge, it take more IO cost to retrieve the data.

⇒ DBMS solution to it :- "Index to database"

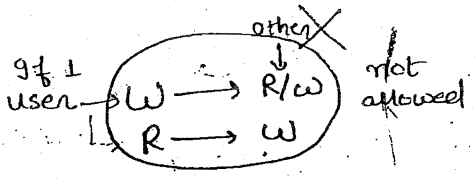
Eg 1000 pg book .... worst case 1000 pg turn.

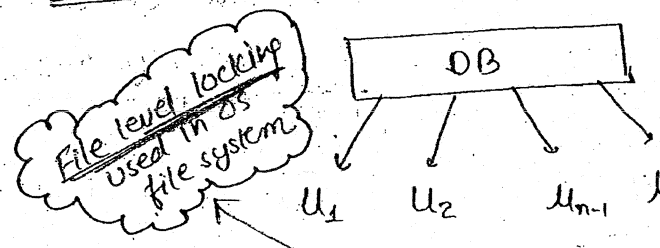
now 30 - pg - index file, then to retrieve only (30+1) = 31 access required.

(Access a page in 1000 page book)



③ Concurrency control :





no conflict even then
U_1: Update R_1
U_4: Update R_4

while R_1 accessed by U_1, it acquire lock on student.txt thereby preventing U_4 from accessing the file. Thus

(less concurrency level)

DBMS Solution

U_1: R_1 *L
U_4: R_4 *L

» Allow record level locking. i.e does not allow multiple users to update same record but provide access (read) to different records.

(High concurrency level)

④ Security :

student.txt

Sid	Sname	DOB	Marks	Ph no	Address

outsiden (sname, sid)

Admin (sid, sname, address)

Faculty (sid, sname, marks)

File System security

PASSWORD Security only

(multi-level security)

→ This type of security not there in OS. To forcefully provide that file. (REDUNDANCY)

» Problems if we maintain different files.

- Redundancy
- Too much space consumed (initially file=500GB) now it will be » 500 GB ∵ of redundancy.
- To update any column, too much overhead involved as updation need to be done in each file else inconsistency may arise.

» Different levels of security too difficult.

DBMS Solution

(VIEW) (Virtual Table)

[V(F1)] , [V(F2)] [V(F3)] } Views of stud, faculty stud, admi std, outside

(V(F3) → view (virtual table) of file 3)

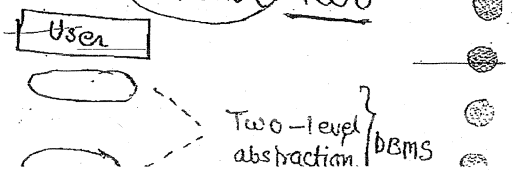
⇒ Only one physical table is there from which different virtual table (view) are created.

DATA INDEPENDENCY : [DATA MODEL] ← given by CODD

↓

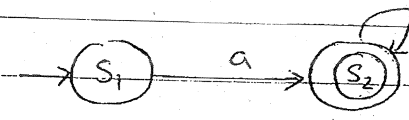
Hiding the low level (physical) details from the external user.

Codd [Relational DBMS] : To provide data independency there should be (atleast) two levels of abstraction.



Date / /

Basics of TOC

$L_1 = \{a^n / n \geq 1\}$ 

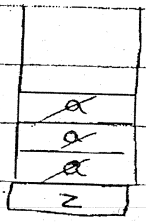
Reg.

$L_2 = \{a^n b^n / n \geq 1\}$ Not Reg

finite automata fails to recognize those languages where memory required.

F.A + DS
F.A + 1-stack
↓
P.D.A.

aaabbbb∈



$L_3 = \{a^n b^m , m, n \geq 1\}$ Regular.

$L_4 = \{a^n b^m , m \neq n , m, n \geq 1\}$ Not Regular.

$L_5 = \{a^m b^n / \underset{m,n \geq 1}{m == n+10}\}$ Not Reg

$m == n$
$m > n$
$m < n$
} Not reg.

$L_6 = \{a^m b^n / \underset{m,n \leq 10}{m == n}\}$ Regular.

$L_7 = \{a^n b^n c^m / m, n \geq 1\}$ Not Reg. CFL.

$L_8 = \{a^m b^n c^n / m, n \geq 1\}$ C.F.L

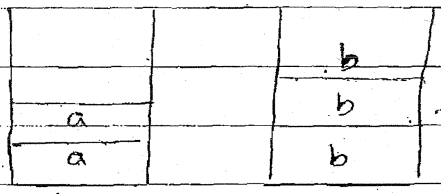
Chitra

Date / /

$L_9 = \{a^n b^{m+n} c^m\}$ C.F.L.
$m, n \geq 1$

$L_{10} = \{a^n b^m a^n b^m \mid m, n \geq 0\}$ C.S.L.

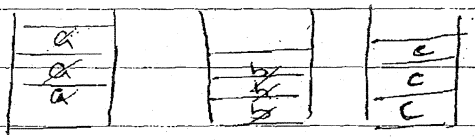
aabbbaabbb



PDA + 1 stack ⇒ Turing M/c.

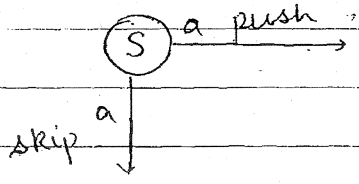
L_{11} $\{a^n b^n c^n / n \geq 1\}$ C.S.L.

aaabbbccc



L_{12} $\{a^i b^j c^k / i = j$ or $j = k$ where $i, j, k \geq 1\}$

C.F.L. (NPDA)

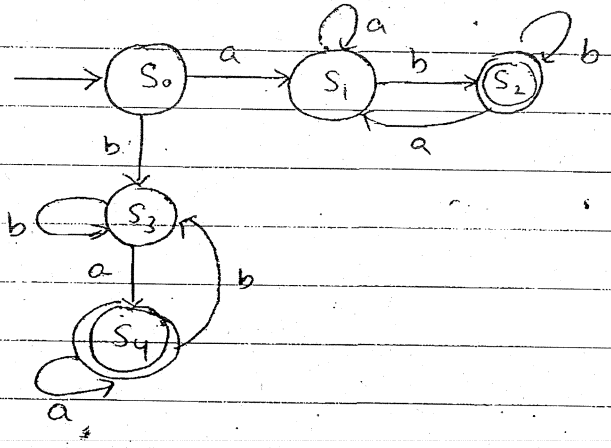


$L_{13} = \{a^n b^n c^n d^n\}$

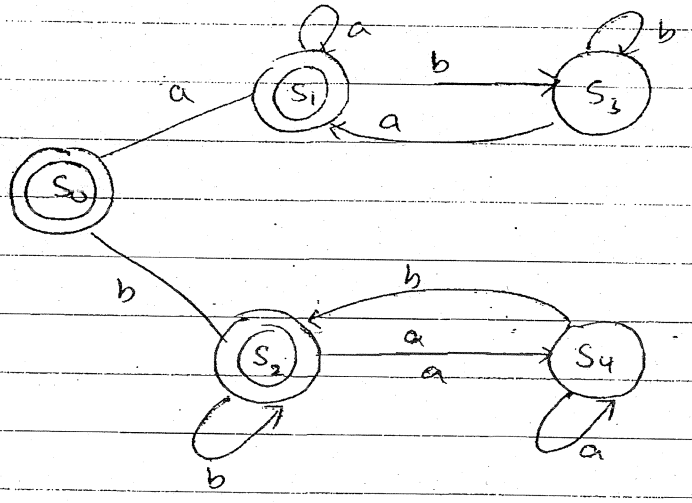
Chitra

Date

Construct DFA that accept all string of a's & b's where each string contain starting & ending symbols are different



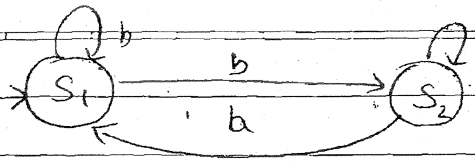
Construct minimal DFA that accept all strings of a's & b's where each string starting & ending symbol are same.



Chitra

Date / /

(1)

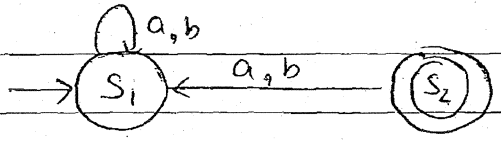


$L = \{\ \} = \phi$ empty lang^n (no string is there)

(2)

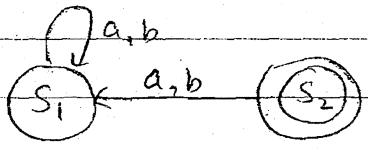
$L = \{\lambda\}$ empty string

(3)



$L = \{\ \}$ empty lang^n

Minimization: If two are behaving same then make it one



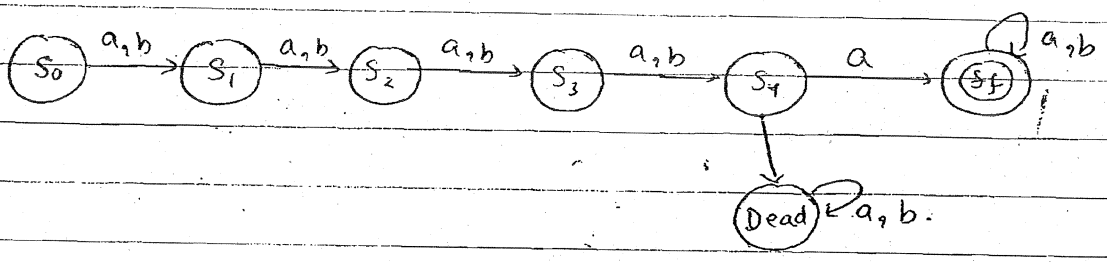
after minimization ⇒ 2 states.

bcoz minimization doesn't mean to remove unreachable state. It is a procedure to remove unreachable state before applying minimization bcoz it make problem easier.

Chitra

Date / /

Construct minimal DFA that accept all strings of a's & b's where in every string 5 symbol from the left hand side is "a"?



Temporary Non-final means in future it may be accepted.

5 LHS, a

⇓

6 + 1 = 7

$5 \Rightarrow 5 + 1 + 1$

$\Rightarrow 7$

$99 \Rightarrow 99 + 1$

$+1$

101

Note

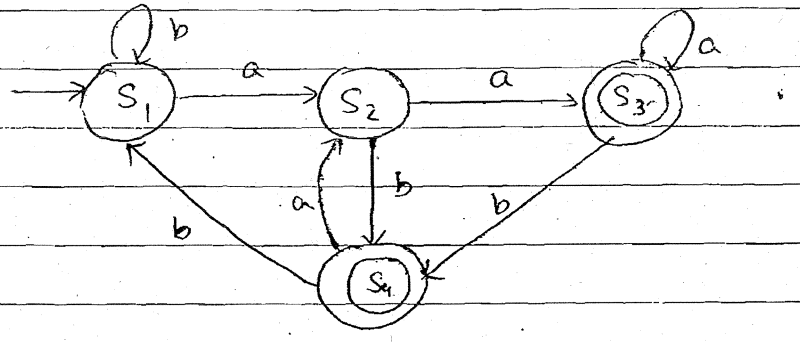
The minimal DFA that accepts all strings of a's & b's where the n'th symbol from the L.H.S contain "$n+2^4$" states.

Chitra

Date / /

Construct minimal DFA that accept all string of a's & b's where the second symbol from the R.H.S is a.

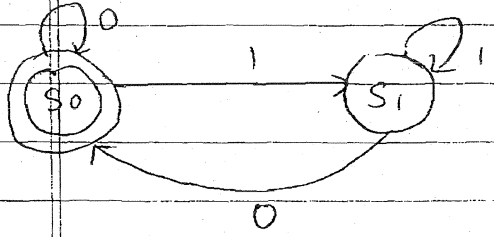
$(a+b)^* \overset{3}{a} \overset{2}{a} \overset{1}{(a/b)}$



Note: The minimal DFA that accepts all string of a's & b's where each ~~symbo~~ string contain n symbols from R.H.S sequires 2^n states

Construct minimal DFA that accept all string of 0's & 1's where every string is divisible by 2.

(every string ending with 0)



0000 ✓
1000 ✓
1100 ✓
1110 ✓
0001 ×
1001 ×
1101 ×
1111 ×

Chitra