Database - Exam 2016

1 SQL Queries

Student
sid sname address syear

Courses
cid cname duration cyear

enrolled
sid cid mark

① Name of Students who don't have any marks <10.

```
Select sname from Students
natural join Enrolled
where sid not in (select sid from Enrolled where mark < 10);
```

② Pair of students (names) who have same mark in same course

```
Select S1.sname as sname1, S2.sname as snam2 from Studnts S1.
natural join Enrolled e1
join (Students S2 natural join Enrolled e2)
where e1.sid < e2.sid
and e1.cid = e2.cid and e1.mark = e2.mark
```

③ Name of Students who are not enrolled in all courses of their year

```
select Sname From Students
where sid not in
( select sid Room Enrolled
  n.t Courses
  where cyear = syear)
```

4 For each courses, cname and student name with best mark

```
select cname, sname from Courses c1
natural join Enrolled e1
natural join Students s1
where mark = (select max(mark) from Enrolled where c1.cid=e1.cid)
```

5 For each course, its name, nb student enrolled, best mark

```
select cname, count(distinct(sid)), max(mark) from Courses
natural join Enrolled
group by cid;
```

6 For each year, course name with longest duration

```
Select cyear, cname from Courses1
where duration =
    (select max(duration) from Courses c2
        where c1. cyear = c2. cyear);
```

Database Exam 2016

Creating DB

```
create table Students(
      sid    int,          PRIMARY KEY
      Sname  varchar (20) not null,
      address varchar (50) not null,
      Syear   int  not null,
   constraint  Students_PK  primary key (sid));
```

```
create table Courses(
   cid int,
   Cname varchar (20) not null,
   duration  int      not null,
   Cyear   int   not null,
 constraint Students_PK primary
          key  (cid) );
```

```
Create table Enrolled(
      sid   int
      cid   int
      mark  int not null,
constraint  Enrolled_PK  primary key (sid),
constraint  Enrolled_PK  primary key (cid),
constraint  Enrolled_FK  foreign key (sid)  references  Students (sid),
constraint  Enrolled_FK  foreign key (cid)  references  Courses (cid));
```

Inserting Values

```
Insert into Students values (1,'Kar', 'Chatou', 4);
```

Delete table

```
drop table Students;
delete from Students (where <condition>;)    optional where clause
```

Update

```
Update Students
  Set    syear = 1
  where   sid < 10;
```

Constraints

- keys (primary or unique key)
- foreign key
- value-based particolar attribute
- tuple-based relationship among component.
- assertions any SQL boolean expression

Setting Policy

```
create table Sells(
   ...
   Foreign Key (Beer) references Beers(name),
   ON delete Set NULL
   ON UPDATE CASCADE);
```

Tuple Based Check

```
check ((cyear <=2 and duration <=24) or cyear >2)
```

↳ check on insert or update only

↳ refer to attribute of the relationship

Assertions

```
create assertion FewBar check(
  (select count(*) from Bars) <= (select count(*) from Drinker));
```

↳ per year avg

↳ check assertion after every modificat° to DB relation.

```
Create assertion CourseDuration check(
  Not Exists(
    select cid from Courses group by cid
    having 20 <= avg(duration)
));
```

— courses with avg durat° above 20h.

Database Exam 2016

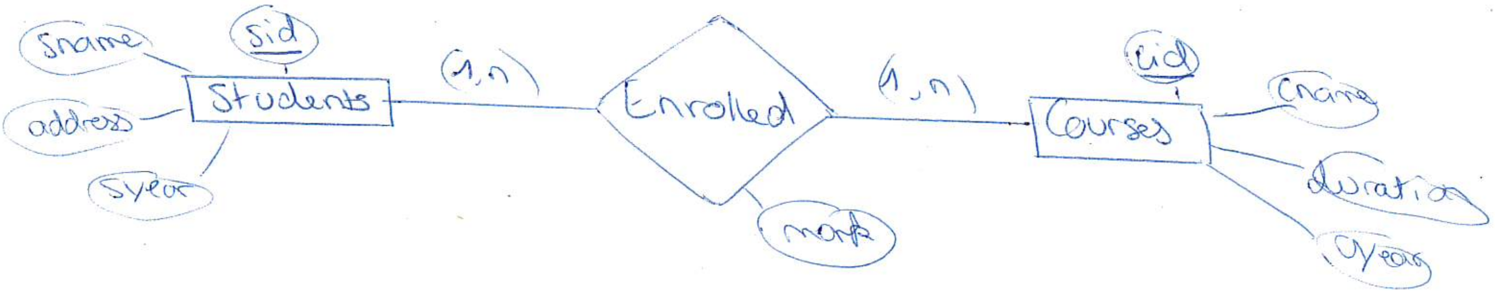
Triggers

- Event   type of DB modification "insert"
- Condition   any boolean-value exp.
- Action   any SQL statemt

1) Create trigger Upper_Trig
Before Insert, Update ON Courses
Referencing New Row as New
For each Row Set New.Cname = upper (New.Cname)

2) Create trigger Student_History_trig
After update, insert on Students
referencing new Row as N
for each row ~~then~~
Insert into StudentHistory values (N.sid, Curdate(), N.address, N.year)

Conceptual Modeling:



⑥ Java DB Connectivity

```
public List <Float> getMarks (int year)
    throws SQLException {
  // decloration variable
  List <Float> marks = new ArrayList <Float>();

  String query = " select mark from Enrolled naturral join Students" +
                 " where syear = ? ";
```

prepare stmt
setFloat()
resultset

```
   PreparedStatement stmt = connection.prepareStatement (query);
    stmt. setFloat (1, year);
    ResultSet rs = stmt. executeQuery ();
    while ( rs. next ()) {
         marks. add ( rs. getFloat (" mark "));
    }
  return marks;
  }
```

Mongo DB

① Total salary of clerks

```
db.employees.aggregate([ {$match : { job : "clerk"}},
    {$group : {_id : null, total_sal : { $sum : "$salary"}}} ]);
```

② Cities where there is a department

```
db.employees.distinct("department.location");
```

③ Name of employees who work in Chicago

```
db.employees.aggregate([ { $match : { department.location : "Chicago"}
    { $group : {_id : "name"}} ]);
db.employees.find("department.location" : "Chicago"), {_id : 0, name : 1}
```

④ Employee whose name begins with "M' and contains 't'.

```
db.employees.find("name": /^M.*t/} {name:1});
```

⑤ For each job : the job and nb of employees

```
                                                      = Count.
db.employees.aggregate([{$group : {_id : "$job", nbEmp : { $sum : 1} }}]);
```

⑥ Name of depmt with at least 5 employees

```
db.employees.aggregate([{$group : {_id : "$department.name", nbEmp : { $sum : 1}}}
    { $match : { nbEmp : {$gt 5}}} ]);
```

⑦ The highest of per-department average sal.

```
db.employees.aggregate([{$group : {_id : "$department.name", avgSal : { $avg : "$salary"
    }}}, {$group : {_id : null, maxAvgSal : { $max : "$avgSal"}}} ]);
```

⑧ The avg salary of managers.

```
db.employees.aggregate([ {$match : {job : "manager"}}, { $group :
    {_id : null, avgSal : { $avg : "$salary"}}} ]);
```

⑨ the cities where at least 2 missions took place

```
db. employees. aggregate ([ {$unwind: "missions"}, {$group: {_id: "missions. location",
    nbMission: {$sum: 1}}}, {$match: {nbMission: {$gt 1}}} ]);
```

⑩ Name of employees who did a mission in the city they work in.

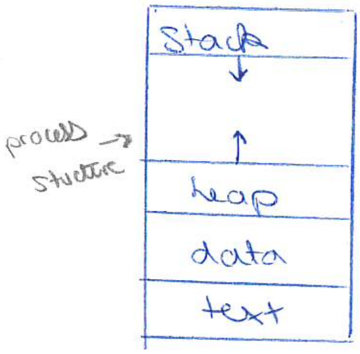
Fiche : Operating Systems

program : static, source code / binary code

process : executed program (dynamic!)

CPU → Central Processing Unit

Process context



File descriptor table

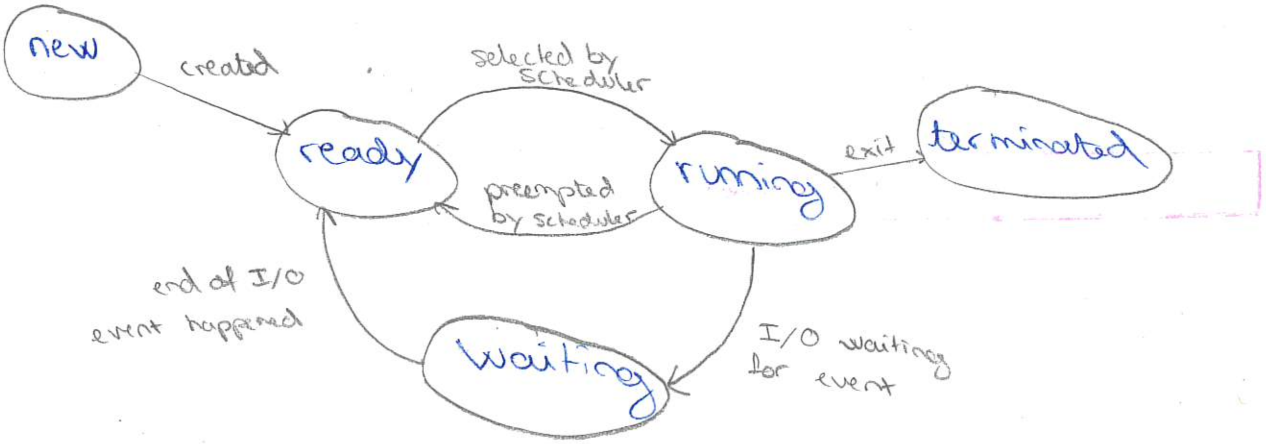
0	std in
1	std out
2	std err

Registers
shared memory
semaphores ...

← all used resources

Process control block : Contains all the info the system has about a process (process state, process nb, program counter, registers, memory limit, list open files)

Process state



- → Process created, enters the READY state
- → READY to RUNNING : process chosen for execut° by scheduler
- → RUNNING to BLOCKED : waiting I/O , event to happen (semaphore)
- → RUNNING to READY : preempted by scheduler (or time-out)
- → BLOCK to READY : I/O or event completed

Any process is an alternating sequence of CPU & I/O **bursts**.

Scheduling Goals : benefit from available resources as much as possible

→ max CPU utilisat°
→ max throughput

↘ response time
↘ turnaround time
↘ waiting time

Different Schedulers

- Short-term (CPU) : selects new process for CPU
- Medium-term : swapping scheduler, ↓ degree of multiprogr.
- Long-term : controls nb of process in memory

Context Switch

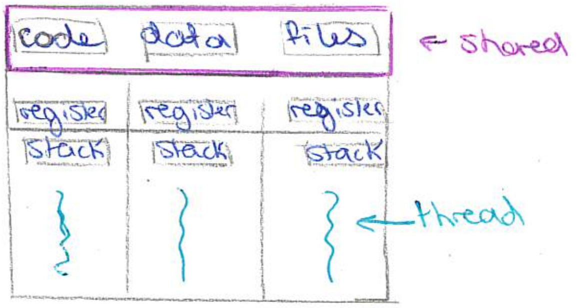
It happens when there are transitions in the process lifecycle

→ Scheduler decides to run another process
→ A process is terminated
→ A process waits for an event (I/O)

The context of current process is saved and replace by context of new running process.

Threads → share data within the same context
(whereas process are independent)

⊕ faster than process b/c smaller context to save & replace during ctxt switch
⊕ threads use less memory than process (code, data, heap, other resources shared)



→ Threads in a process share code, data, files (File Desc. for table)

File Descriptor $\xrightarrow{\text{uses}}$ File Table $\xrightarrow{\text{uses}}$ Inode table

Inode : data struct. that describes filesystem object (file or directory)
It contains file size, timestamp (last modificat°), file permiss° NOT File Name

Fiche: Operating Systems

2

Scheduling Algorithms

	creating time	exec time	Priority	I/O
p1	2	6	4	2/4
p2	0	2	2	1/3
p3	1	5	4	3/1
p4	0	3	3	/
p5	5	1	5	/

← {input à 1s
revient après 3 secs

← {input à 3s
revient après 1 sec

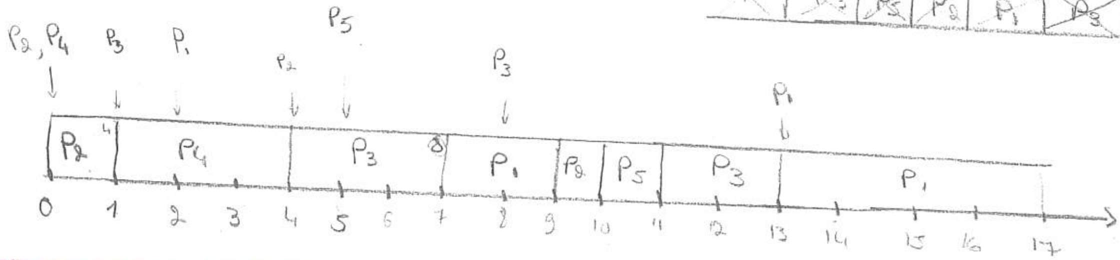
1 First Come First Served (FCFS)

Gont Chart

not preemptive

ready queue

~~P1~~	~~P3~~	~~P5~~	~~P2~~	P1	~~P3~~	~~P4~~	~~P2~~



Turnaround-time = end of process − creating time

Waiting time = Turnaround time − exec time

P1 = 17 − 2 → P1 (15)
P2 = 10 − 0 → P2 (10)
P3 = 13 − 1 → P3 (12)
P4 = 4 − 0 → P4 (4)
P5 = 11 − 5 → P5 (6)

P1 = 15 − 6 = P1 (9)
P2 = 10 − 2 = P2 (8)
P3 = 12 − 5 = P3 (7)
P4 = 4 − 3 = P4 (1)
P5 = 6 − 1 = P5 (5)

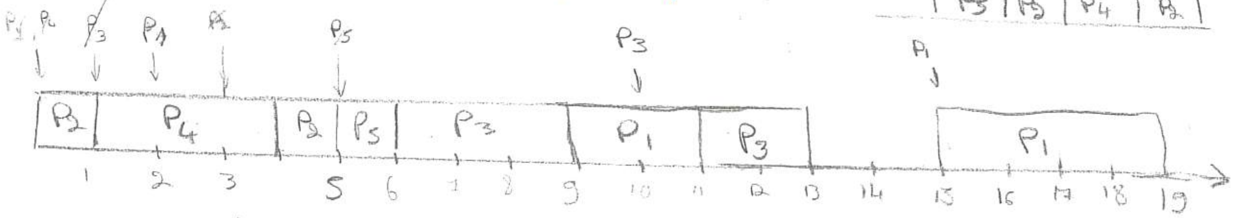
Average waiting time $= \frac{9+8+7+1+5}{5} = 6$ sec

2 Shortest Job First

→ priority based

preemptive
& non preemptive

P5	P3	P4	P2



Turnaround time $\overset{19-2}{P_1(17)}$ $\overset{5-0}{P_2(5)}$ $\overset{13-1}{P_3(12)}$ $\overset{4-0}{P_4(4)}$ $\overset{6-5}{P_5(1)}$

Waiting time $\overset{17-6}{P_1(11)}$ $\overset{5-2}{P_2(3)}$ $\overset{12-5}{P_3(7)}$ $\overset{4-3}{P_4(1)}$ $P_5(0)$

Average waiting time $= \frac{11+3+7+1+0}{5} = 4{,}4$ sec

③ Priority Algo

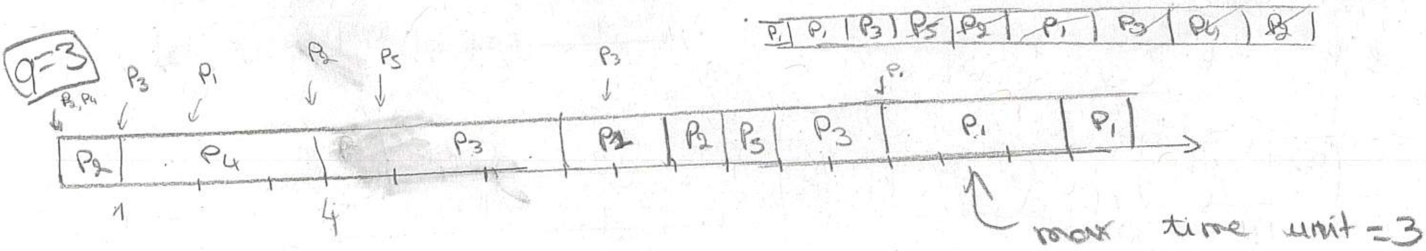
- → preemptive or non-preemptive
- → process with highest priority is selected

⚠ Starvation possible ⇒ lower priority process may never execute

↳ Solution : increase priority w/ "age" (time)

⓸ Round Robin → preemptive

- → time-sharing algo , CPU time
- → each process executes for a time quantum.

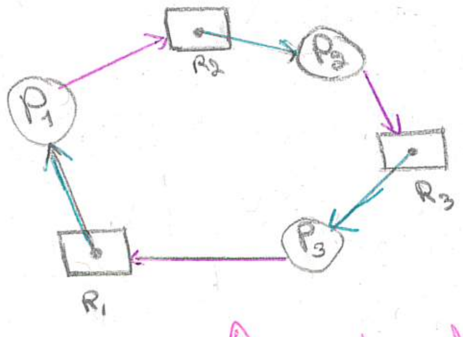


Preemptive & Non-Preemptive Scheduling

A preemptive scheduler can decide to interrupt running process (push it back on ready queue) & replace it by another.

Starvation : when process is not able to run for long time b/c other process are elected by scheduler. Only with priority based algo (priority, preempt SJF).

Resource allocat° graph



→ resource used by process

→ process waiting for resource

⚠ circular dependency = deadlock

Necessary Condit° for Deadlock

- → mutual exclus° : only 1 process at a time can use resource
- → hold & wait process holding R & waiting for another R held
- → no preemption R can only be released volontarily by procss
- → circular wait P1 waits for Resource held by P2
  P2 " P3 etc

Process Synchronizat

→ When 2 process modify same data "simultaneously" ⇒ Concurrency problem

→ "critical" code sections

Solution: mutual exclusion = one process at the time has permiss' to modify shared data.

Semaphores a structure containing

→ Counter (to count resources)

→ waiting queue (to wait for an unavailable resource).

Two operations:

P (aquire):

```
counter --;
if(counter<0) ⇔ if(resource unavailable)
    Add process to waiting queue
```

V (release)

```
counter ++;
if(counter>=0) ⇔ if (waiting not empty)
    wake up the process
```

* Semaphores used a synchro tool
* P and V operations are atomic/indivisible

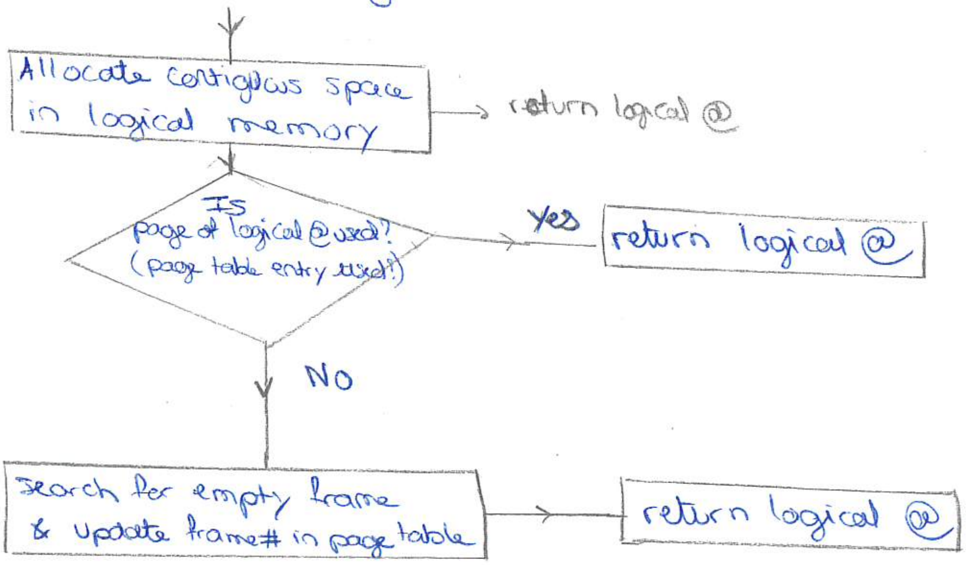
If one process should be allowed at a time ⇒ initialized sephamore mutex to 1.

```
T1 { P(mutex); i++; V(mutex)}
T2 { P(mutex); i++; V(mutex)}.
```

sm_mutex = 1;

Higher level tools ⇒ monitors.

Algorithm for one-level paging scheme



Allocate contiguous space in logical memory
return logical @
Is page of logical @ used? (page table entry used?)
Yes
return logical @
No
Search for empty frame & Update frame# in page table
return logical @

Memory Managemt

↳ Contiguous allocat°
↳ Paging
↳ Segmentation

A) Contiguous Allocation

hole: block of available mem.

Process allocates mem in a hole large enough to accommodate it.

Steps to Allocate

① Find free zone (contiguous) by looking at list of holes

② If not find → ERROR
Else Update list of holes & return base address (1st allocated slot)

3 Allocat° Strategies:

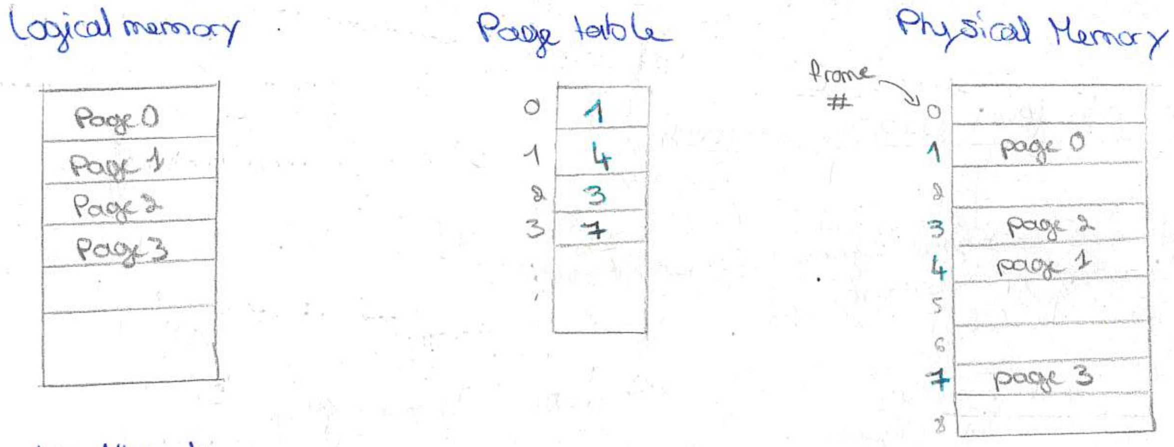
→ First Fit: first on route big enough

→ Best Fit: allocates the smallest big enough hole

→ Worst Fit: allocates the largest hole big enough

B) Paging

→ physical add. space non-contiguous / fragmented



Steps to Allocate

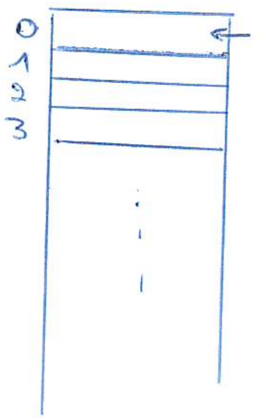
① Find nb of free contiguous slots (& update list of free slots)

② Find nb of free physical slots (non-contiguous)
if page table unused, search for free frame

③ Update mapping table (with frame #)

⚠ When freeing slots, check if whole frame empty than free it & its page

Memory Management (volatile RAM)



slot (size = data bus size)

$\# \text{address} = 2^{\text{size of @ bus}}$

Allocate (nb)

```
int main () {
  int tab [6];
  int tab [5] = 5
 printf ("%d \n", tab[5]);
```

Steps

1. Find nb of free slots (contiguous or not)
   use linked list of free slots ?
   & update the list of free slots

2. Return add. of 1st allocated slot

write (@, val)

val ← read (@)

Free (@, nb)
↳ nb of values.

```
int *p = 250;    → p is a ref to slot nb 250
 *p = 71;        → write into slot
```

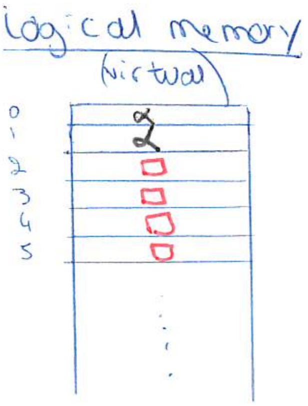
⚠ Unsafe because the memory slot n° 250 hasn't been allocated (haven't ask for permission). It could be used by other ~~memory~~ process.

→ Not contiguous memory slot is better option
but need to return the entire list of free slots.
still unsafe because we should "hide" from other process!

Virtual logical memory

Each process has its own logical space (does'nt see other process)
the process will allocate in virtual space, then syst allocates in physical memory.

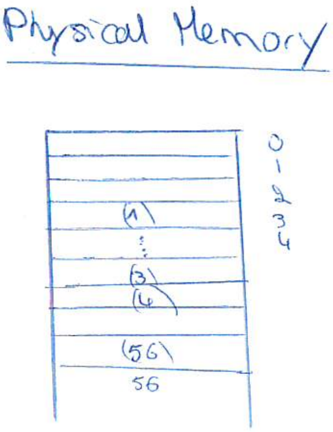
logical memory (virtual)



Mapping table

logical @	Physical @
2	4
3	72
4	56
5	57
6	1011
7	1012

Physical Memory



Steps

1- Find nb of free logical slots (contiguous)
update list of free slot

2- Find nb of free physical slots (non-contiguous)

3- Update the mapping table

My Alloc (nb)

1- Allocate nbcontiguous space in logical memory
2- Allocate nb non contiguous slots in physical memory and update the mapping table
3- return the @ of the 1st logical slot.

However, space consumming (size = $2^{32} \times 4$ bytes = 16 GB) → TOO MUCH!
→ Instead of having 1 to 1 slots, we are going to use bigger ~~blocks~~ blocks (4bytes) in the logical memory.

$\frac{2^{32}}{2^{12}} \times 4$ bytes = 4MB

Application OS example : Memory Managem^t

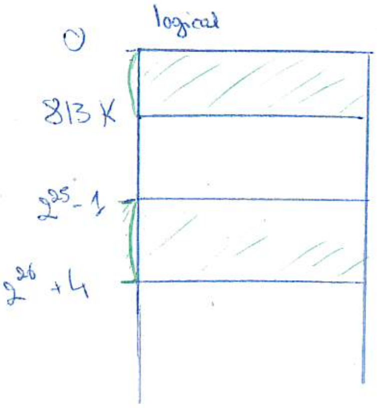
Mem : addresses over 32 bits

3 level paging scheme (7, 8, 7, 10)

block # subblock# page# offset.

Process : [0, 813K] ∪ $[2^{25}-1, 2^{26}+4]$

How many tables are used & their associated size.
Compare it to a 1 level paging scheme.



1st level → block table

2nd level → sublock table (each block has its own sublock table)

3rd level → page table (each sublock table has its own page table)

@ use 32 bits

size block table = 128 × 4 = 512 bytes

block size = $\frac{2^{\boxed{32}}}{2^7} = 2^{25}$ bytes . slots

[0 - 813K]

$813 \times 2^{10} < 2^{25}$ → inside block # 0.

Other Method : 1st block used = $\frac{\text{Start}}{\text{block size}} = \frac{0}{2^{25}} = 0$ ⚠ integer division

last block used = $\frac{\text{end}}{\text{block size}} = \frac{813 \times 2^{10}}{2^{25}} = 0$

$[0 - 2^{27}]$ → start = $\frac{0}{2^{25}} = \boxed{0}$ end = $\frac{2^{27}}{2^{24}} = \boxed{4}$

→ 1 block is being used.

size of sublock table : $\underbrace{256}_{2^8} * 4 = 1\,k$ byte

S.block size = 2^{17} words (slots) = $\dfrac{2^{32}}{\underset{\text{nb blocks}}{2^7} \times \underset{\text{nb sublocks/blocks}}{2^8}} = 2^{32-15} = \boxed{2^{17}}$

# of sublocks used = $\dfrac{2^{20}}{2^{17}} = 2^3 = 8.$

→ We need 8 page tables

C1 : [0-813K] ⇒ 1 block table
+ 1 sublock table
+ 8 page tables

$[2^{25}-1 \,,\, 2^{26}+4]$ start : $\dfrac{2^{25}-1}{2^{25}} = \boxed{0}$ end : $\dfrac{2^{26}+4}{2^{25}} =$

Fiche Network

The OSI model has **7 layers**.

TCP/IP model does not have session and presentation layers.

Which layer is responsible for process to process delivery? **Transport layer**.

The **data link layer** is responsible for **delivery of frames** between two neighboring nodes over a link. This is called **node-to-node delivery**.
The **network layer** is responsible for **delivery of datagrams** between two hosts. This is called **host-to-host delivery**.
Real communication takes place between two processes (application programs). We need process-to-process delivery. The **transport layer** is responsible for **process-to-process delivery**-the delivery of a packet, part of a message, from one process to another.

Which address identifies a process on a host? **Port address.**

Application layer is implemented in **End system**.

Transport layer is implemented in **End system**.

Transport protocol is implemented in **End hosts.**

The physical layer concerns with **bit-by-bit delivery**.

The physical layer is responsible for **line coding, channel coding, modulation**.

The physical layer translates logical communication requests from the **data link layer** into hardware specific operations.

A single channel is shared by multiple signals by **multiplexing**.

A bridge is a layer 3 device →**false**. **A BRIDGE IS A LAYER 2 DEVICE.**
__Bridge__ works at the __data link layer__ (not the case of hub and router).

Which one(s) of the following task(s) is done by the data link layer?

a) Framing
b) Error control
c) Flow control
d) All the above

Which one of the following task is **not done** by data link layer? **Channel coding** (framing, error control and flow control are done by data link layer).

The Medium Access Control (MAC) protocol used by Ethernet (802.3) is **CSMA/CD**.

The Medium Access Control (MAC) protocol used by Wi-Fi (802.11) is **CSMA/CA**.

Ethernet Physical address is made of **6 bytes**.

The IPv4 address consists of **32 bits**. (which limits the address space to 2^{32} addresses)
The IPv6 address consists of **128 bits** (= 16 octets).

A bridge is used to interconnect multiple LANs into a larger LAN.

TCP provides logical communication between **processes.**
TCP provides a communication channel between processes on each host system.

Which of the following transport protocols is more suited for Multimedia application? **UDP**
Which of the following transport protocols is more suited for file transfer? **TCP**

UDP is a transport protocol that is **connectionless**. Use ~~handshaking~~ (there is no ACK in UDP and thus no three-way-handshake).
TCP is a transport protocol that is **connection-oriented, uses 3 way handshaking.**
TCP handles **sequence numbers, acknowledgement, retransmission.**

The technique of temporarily delaying outgoing acknowledgements so that they can be hooked onto the next outgoing data frame is called **piggybacking**.

The cutoff frequency is **the frequency above which the signal is received without attenuation**.

The bandwidth is **a physical property of the transmission medium + is dependent on the length of the medium + the width of the frequency range transmitted without being strongly attenuated.**

RF modulation is **the variation of one or more properties of an RF signal.**

A hub is a layer 2 device. → **false** (physical layer).
A bridge is a layer 3 device. → **false** (data link layer).
A switch is a layer 2 device. → **true.**
A router is a layer 3 device. → **true.**

The IP address consists of **network address, host address.**

Transport layer aggregates data from different applications into a single stream before passing it to **network layer.**

A bridge is used to **connect a LAN.**

TCP provides logical communication between **processes.**

During congestion in a network, **TCP reacts to it by decreasing its congestion window.** (not UDP!).

TCP slow start mechanism consists of **multiplying congestion window size by 2 at each successfully received ACK.**

Network

TCP congestion avoidance is initiated when **a loss is detected after a time out + a loss is detected after duplicate ACK.**

In 802.11 Wi-Fi standard **a Wi-Fi access point bridges the traffic towards the gateway.**

A VLAN **allows separation between networks + reduces broadcast storms + increases security.**

Congestion window (cwnd) is a TCP state variable that limits the amount of data the TCP can send into the network before receiving an ACK. The Receiver Window (rwnd) is a variable that advertises the amount of data that the destination side can receive. Together, the two variables are used to regulate data flow in TCP connections, minimize congestion, and improve network performance.

Three parameters are tracked for controlling congestion: CWND, Sequence numbers, ACK numbers.

MSS: Maximum Segment Size

- Which address uniquely defines a host on the Internet? → **IP.**
- When the data packet moves from the upper to the lower layers, headers are **added.**
- Which type of protocol is IP? **Unreliable and connectionless.**
- Which address is also known as link address? **Physical.**
- In OSI model, which layer is responsible for encryption and decryption? **Presentation layer.**
- Which layer is responsible for movement of individual bits from one node to another? **Physical.**
- Which layer is responsible for moving frames from one node to another? **Data Link.**
- Which layer is responsible for delivery of individual packets from source host to destination host? **Network layer.**
- Types of addresses used in internet? **Physical(Link), address, port, specific.**
- Which layer enables the users to access the network? **Application layer.**
- In which layer is Segmentation done? **Transport.**
- By which layer is routing handled? **Network layer.**
- **Bridge** works at the **data link layer** (not the case of hub and router).
- **Router** primarily functions at network layer.
- What is the main function of session layer? **Dialog control.**
- By which layer is used POP3? Application layer.

The start and stop bits are used in serial communication for **synchronization.**
Which network has connectivity range up to 10 meters? **PAN.**
Star topology does not allow direct traffic between devices. **True.**
The functionalities of presentation layer include **Data compression, Data encryption, Data description.**
In the OSI model, as a data packet moves from the lower to the upper layers, headers are **removed.**

Which layer links the network support layers and user support layers? **Transport layer.**

Explanation: Physical, data link and network layers are network support layers and session, presentation and application layers are user support layers.

Transmission data rate is decided by **physical layer.**

Which transmission media has the highest transmission speed in a network? **Optical fiber** (not coaxial cable nor twisted pair cable nor electrical cable).

Which one of the following task is **not done** by data link layer? **Channel coding** (framing, error control and flow control are done by data link layer).

Header of a frame generally contains
   a) Synchronization bytes
   b) Addresses
   c) Frame identifier
   d) All of the mentioned

CRC stands for **Cycling Redundancy Check.**

Which one of the following is a data link protocol?
   a) Ethernet
   b) Point to point protocol
   c) HDLC
   d) All of the mentioned

Which one of the following is the multiple access protocol for channel access control?
   a) CSMA/CD
   b) CSMA/CA
   c) Both (a) and (b)
   d) None of the mentioned

The network layer concerns with **packets.**

If one link fails, only that link is affected. All other links remain active. Which topology does this? **Star topology** (~~mesh topology, bus topology, physical topology~~).
Twisted pair wires, coaxial cable, optical fiber cables are the examples of **wired media** (~~wireless media~~)

Which cable used in communications is referred to as unshielded twisted-pair (UTP)? **Twisted-pair cable.**

Which is also known as a connectionless protocol for a packet-switching network that uses the Datagram approach? **IPv4** (~~IPv6~~)

Which connection provides a dedicated link between two devices? **Point-to-point** (~~multipoint~~)

Entrainemt QCM Reseau S6

Network Layer

!

(1) conains packets

(2) NL functions : routing , internetworking, Congest° control

(3) 4 byte IP address consists of network + host add

(4) In virtual circuit ntw each packet contains Short VC number

(5) routing algos: shortest path, distance vector routg , link state rtg

(6) Multidestinat° routing, dat is not sent by packet

(7) Spanning tree : subset including all routers but no loop

(8) Algo used for congest° control : { traffic aware rtg
admiss° control
load shedding

(9) NL ⇒ internet protocol

(10) ICMP used for error & diagnose

Transport layer

(1) Transport layer aggregate data into sigle stream before passing to netwk layer

(2) TCP & UDP used in internet

(3) UDP connectionless b/c all packets treated independently

(4) TCP connection oriented & uses 3 way shaking + receive data as single stream

(5) Socket : end point of inter process communicat°

(6) winsock → stocket style API for windows

(7) Datagram Congest° Control Proto

(8) A port is TCP name for transport serv. access point

(9) Transp layer : process to process communicat°

(10) Stream Control Transmiss° Protocol (SCTP) → transport protocol

Topology

① Topology : physical or logical arrangmt of ntw

② Star has central controller or hub

③ BUS → multipoint connect

④ WAN : wide ⇒ State , Cantry

⑤ LAN : local ⇒ campus

⑥ WAN wide area Netwk

⑦ TDM slots divided into Frames

⑧ FDM multiplexing techniq shift sig to diff carrier Frequency

Entrainement QCM Réseau 1

Physical layer

① Phys layer bit to bit delivery

② Optical fiber highest transmiss° speed

③ bits can be send as analog sig over digital modulat°

④ Phys. signalling sublayer : interfaces w/ media access control

⑤ Phys layer provides
- mechanical specificat°
- electrical specificat°
- specificat° for IR over fiber

⑥ In asynchronous serial comm., phys layer provides
- start & stop (control)
- flow control

⑦ Phys layer responsible for line coding, channel coding & modulat°

⑧ Phy. layer translate from data link layer

⑨ Multiplexing shares single channel for multiple sig

⑩ Wireless transmiss° done by radio, micro waves & infrared

Data link

① DLL takes packet from network layer

② DLL does framing, error & flow control.

③ DLL sublayer performs data link funct° (depending type of medium) ⇒ media access control MAC sublayer

④ Header of frame contains
- synchro bytes
- addresses
- frame identifier

⑤ Logical link control sublayer ⇒ auto. repeat request error management mechanism

⑥ burst error : 2 or ⊕ bits in data units changed in transmiss°

⑦ CRC : cyclic redundancy check

⑧ DLL protocol : ethernet, point to point, hdlc

⑨ multiple access control protocol CSMA/CD & CSMA/CA

⑩ Piggybacking technique delaying outgoing ACK to be hooked on next one

① OSI stands for Open syst interconnect

② OSI has 7 layers

③ TCP/IP does not have session & presentat° layer

④ Transport layer links network & user support layers

⑤ Physical + logical address, specific address, port add. used in an internet employing TCP/IP protocols.

⑥ TCP/IP devlp prior to OSI model

⑦ Transport layer responsible for process to process delivery

⑧ Port address identifies process on a host.

⑨ Application layer provides services to User

⑩ Transmission data rate is decided by physical layer

Reference models

① Nb of layers in Internet Protocol Stack = 5

③ Applicat° layer is implemented in End system

④ Transport layer is implemented in End System

⑤ Functionalities of presentat° layer : data compression, encrypt°, descript°

⑥ Delimiting and synchronisat° of data exchange ⇒ session layer

⑦ 5th layer to receive data session layer (OSI model)

⑧ 5th " in Internet Protocol stack ⇒ Applicat°

⑨ OSI, as data packet moves from lower to upper layer, headers are removed

⑩ OSI model, one layer may use informat° from other layer

45 - **Prévenir** - *Choisissez les 2 réponses correctes.* Il existe différentes méthodes pour identifier et analyser les risques en recherchant leurs causes, et ainsi pouvoir établir une veille efficace.

[✓ A.] Le diagramme d'Ishikawa
B. Les 5 Pourquoi
[✓ C.] Le SWOT

46 - **Prévenir** - *Sélectionnez la réponse correcte.* La stratégie qui consiste à réduire la gravité d'un risque est une technique de…

[✓ A.] Prévention
[✓ B.] Protection

47 - **Suivre** - *Sélectionnez la réponse exacte parmi les propositions.* La matrice d'Eisenhower permet de choisir l'action la plus appropriée en fonction de l'urgence et de l'importance d'un problème : déléguer, faire, éliminer ou planifier. Parmi les exemples suivants, quelle proposition nécessite de PLANIFIER ?

A. Il serait mieux de classer la liste des 10 invités à la réunion de fin de semaine par ordre alphabétique

B. Le client qui vient d'appeler a besoin d'un devis Faire

[✓ C.] Il faut prévenir le fournisseur que le client a besoin de sa livraison avant midi

D. On pourrait revoir la disposition du bureau [illegible]

48 - **Suivre** - On peut aller plus loin avec la méthode de gestion des risques en l'étudiant sous l'angle des opportunités.

[✓ A.] Vrai
B. Faux

Fiche

GESTION BUDGETAIRE

① Sales budget

Sales budget = Estimated sales (units) × Unit Price = 35 200 €

220 u · 160€

② Product° Program (in Qtes)

→ Whether stock of finished production will ↗ or ↘ of nb units ↘ 100 u

Q Product° = Estimated sales (units) ± variation of stock 220 - 100 = 120 u

③ Product° Budget

120 u

(Raw Mat + Direct labour) × Qtes Produced (units) + Depreciation.

Unitary Cost of Product (€/unit) = $\boxed{\dfrac{\text{Prod°. Budget}}{\text{Qtes}}}$ $\frac{15\,000}{120}$ = 125 K/unit

④ Final Inventory (Finished Goods)

Finished Goods (N) · 120 u × 25 €

$$\boxed{\text{TOT} = \frac{\text{Initial Inv} + \text{Product° Budg. (N+1)}}{(\text{inv units} + \text{prod N+1 units})}}$$

Ending Stock : TOTAL × (Initial inv. units - variat° of units)

⑤ Income Statement

Exp	Revenues
Raw Mat ← € × nb units	Sales (budget)
Direct Lab. ← € × nb units	Variat° Finished Goods (F - I)
Depreciat°	
Other wages	
Ext. Services	Tot Revenues ...
Financial exp	
Tot Expenses ...	

Gross Profit
↳ tax ...
↳ Net Profit ...

$\boxed{\text{Gross Profit} = \text{Tot Rev} - \text{Tot. Exp}}$

⑥ Quaterly Cash Budget

Quarter	Q_1	Q_2	Q_3	Q_4	BS
Inflows · receivables	customer receivables BS	×	×	×	
- sales	×	$\left(\frac{\text{sales}}{4}\right)$	$\left(\frac{\text{sales}}{4}\right)$	$\left(\frac{\text{sales}}{4}\right)$	$\left(\frac{\text{sales}}{4}\right)$ ← receivables N+1
Outflows					
Raw Mat	payables (BS)	$\left(\frac{1}{4} \times R.M\right)$	$\left(\frac{1}{4}\ R.M\right)$	$\left(\frac{1}{4}\ R.M\right)$	$\left(\frac{1}{4} R.M\right)$ ← payables N+1
Direct lab Other wages Ext. serv. Fin. exp. } = SUM	$\frac{\text{TOTAL}}{4}$	$\left(\frac{\text{TOTAL}}{4}\right)$	$\frac{1}{4}$ TOTAL	$\frac{1}{4}$ TOT	×
→ Investmt	investmt	×	×	×	×
→ Taxes	×	×	×	taxes	×
TOT OUTFLOWS	↑ SUM	↑ SUM	↑ SUM	↑ SUM	×
Cash Flow	Inflow - Outflow	In - Out	In - Out	IN - Out	×
Initial Cash Balance	Out drafts (- ....)	↗	↗	↗	×
Final Cash	Cash Fl - Init Cash			End Final Cash	×

Recquired Capital = Q_4 Final Cash + Outdraft

⑦ Balance Sheet N+1

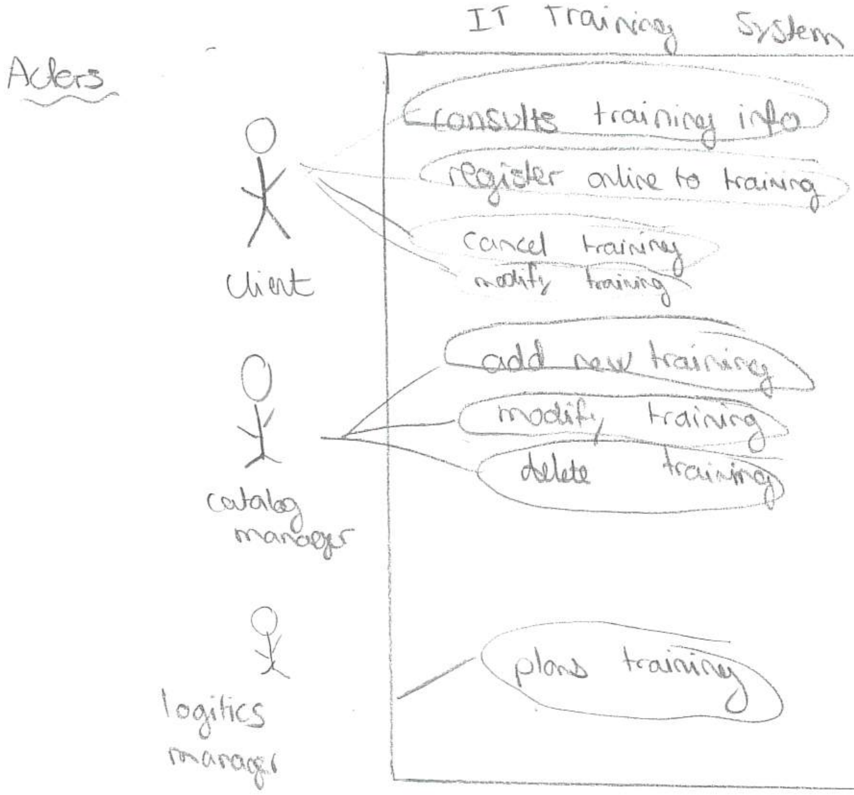
Assets		Liabilities	
Gross V.	$GV_{(N)}$ + Investmt	Eq. Capital	″
Acc. Depre	$A.D_{(N)}$ + Depreciat°	Acc. Profit	$A.P_{(N)}$ + Profit
Net V.	GV - A.D	Profit	Net Profit
Inventories R.M	$(R.M)_N$	Loans	″
Finish. Gd	Ending Stock Q4	Payables	($\frac{1}{4}$ Cash)
		Overdraft	″
→ Receivables	($\frac{1}{4}$ Cash)	TOTAL:	-----
→ Cash	0		
TOTAL :	....		

⑧ Financial table

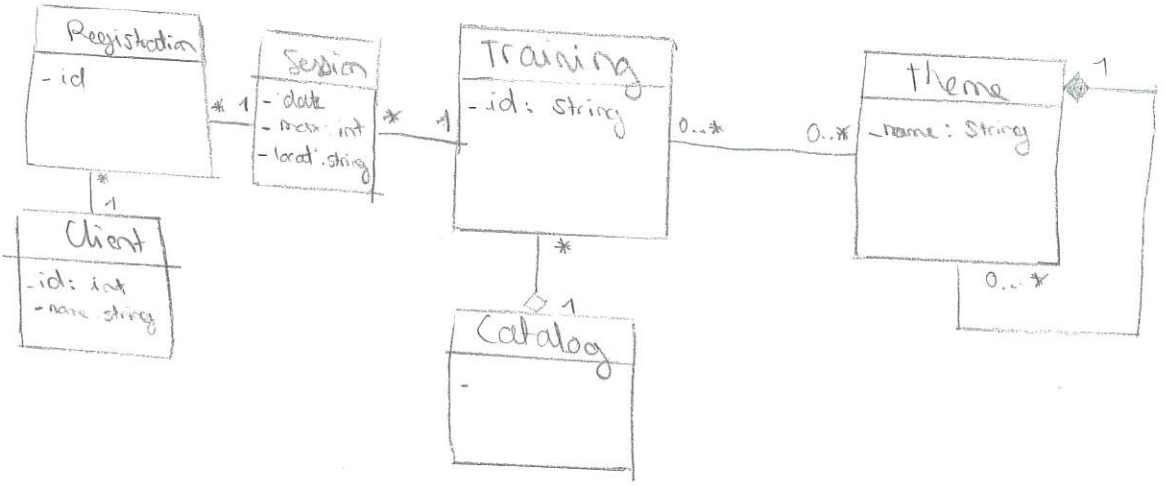
Needs	Resources
investmt	Capital
	Self Finance Capacity *
Var. receivables (F-I)	Var Payables (F-I)
	Var Inventories (F-I)
TOTAL ----	TOT ----

SFC = Net Profit + Depreciat°

UML: Ancien DS
revision

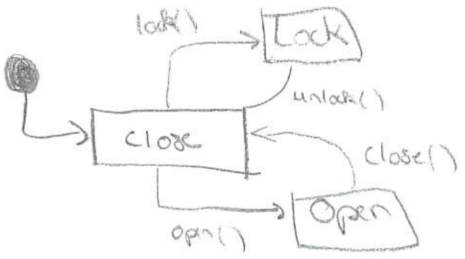


2) Class Diagram



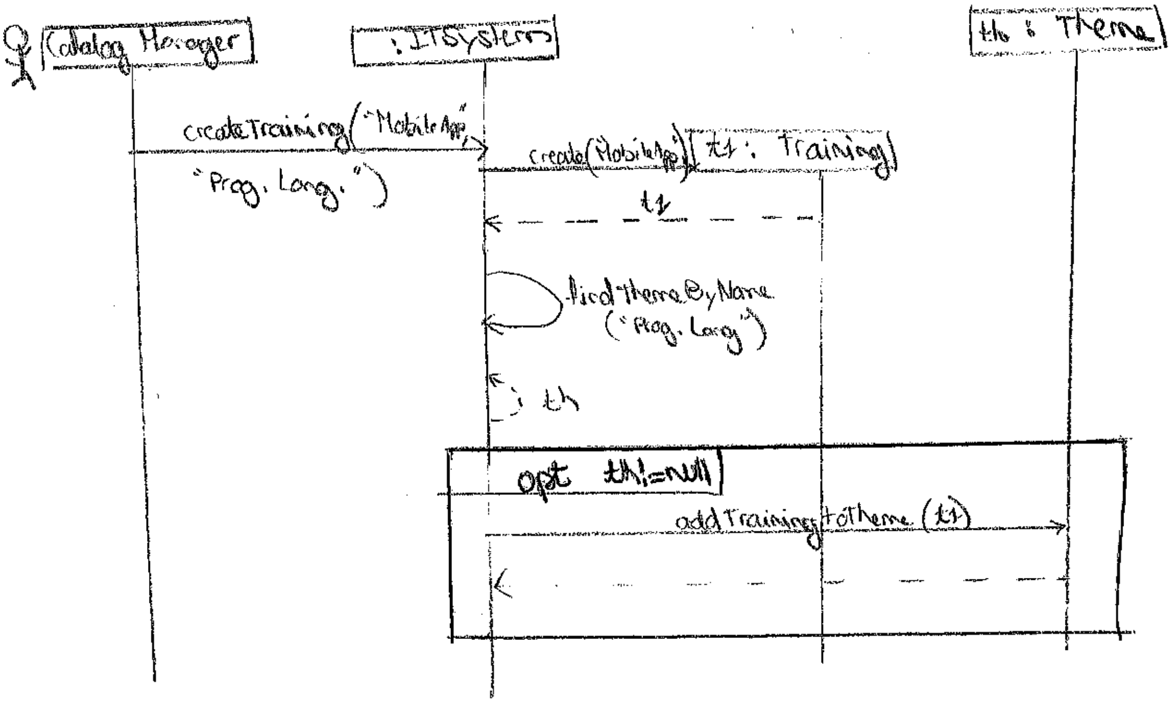
Sequence Diagrame → inter-Object view

State Machine → intra-Object View

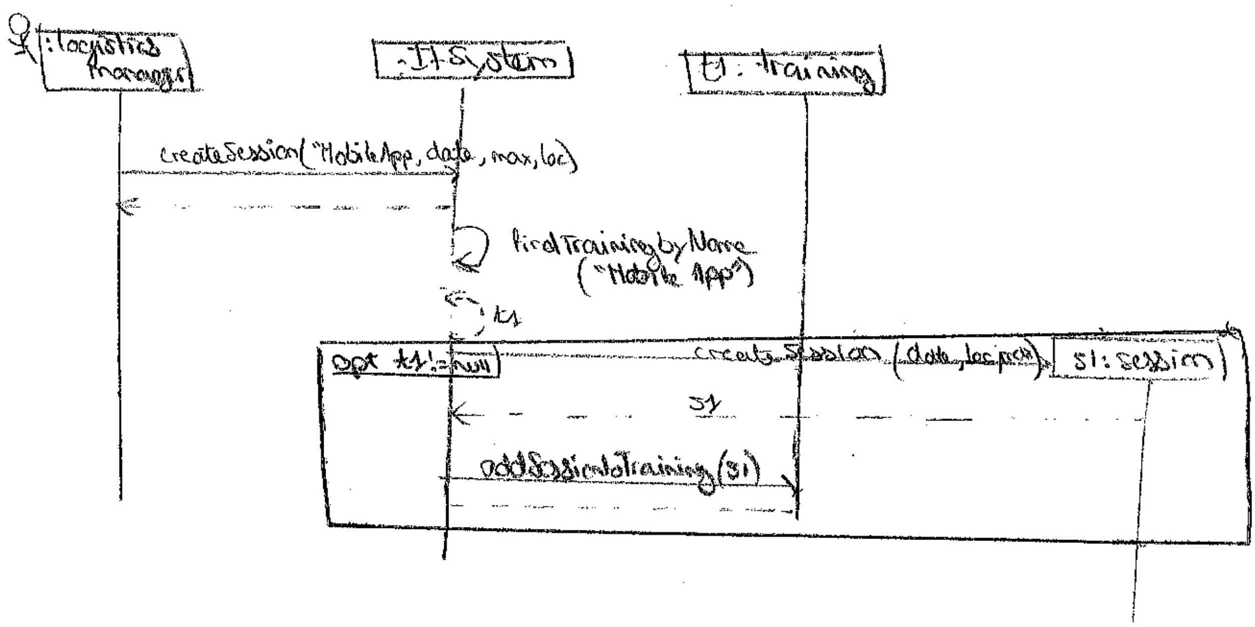


Sequence Diagram

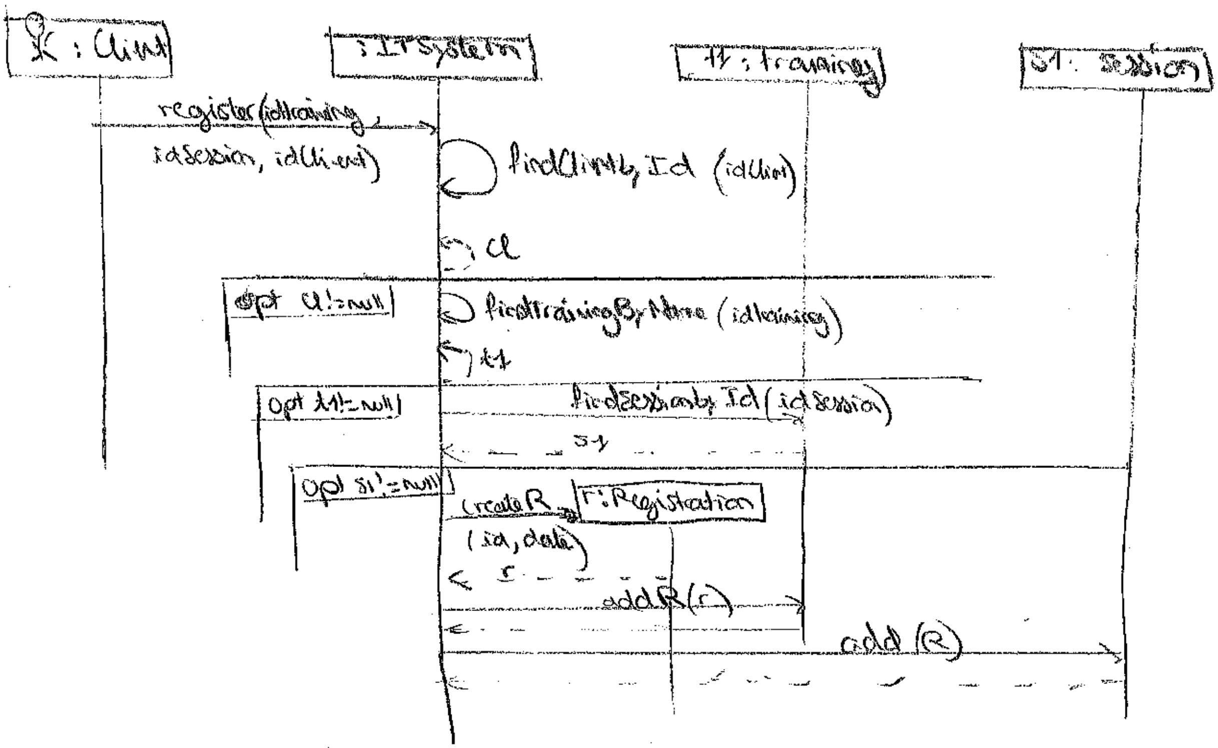
a)



b)


:logistics manager
:ITSystem
t1 : training
createSession("MobileApp, date, max, loc)
findTrainingByName ("Mobile App")
t1
opt t1!=null
createSession (date, loc, max)
s1 : session
s1
addSessionToTraining (s1)

c)


:Client
:ITSystem
t1 : training
s1 : Session
register(idtraining, idsession, idClient)
findClientById (idClient)
cl
opt cl!=null
findTrainingByName (idtraining)
t1
opt t1!=null
findSessionById (idSession)
s1
opt s1!=null
createR (id, date)
r:Registration
r
addR(r)
add (R)