

1) Need of protection:

- Problem may occur in multitasking operating systems or multi-user systems when two or more users attempt to read and change the content of a memory location at the same time.

The section of program where the value of a variable is being read and changed must be protected from access by other tasks until the operation is completed.

- Another region that requires protection is Operating System code. The incorrect address in a user program may cause program to write over the critical sections of operating system corrupting the OS codes and data areas.

The system then 'locks-up' and the only way to get control again is to reboot the system. In a multitasking system this is intolerable, so several methods are used to protect the OS.

Heg-Border

① Explain need of protection mechanism in 80386?

- - Problem may occur when in multitasking operating system or multi user systems when two or more user attempts to read and change content of memory location at some time.
- The main purpose of protection feature of 80386 is to have multitasking with each task having its protected memory. Protection ensure that multiple task cannot access the code and data of other tasks.
- In case of multitasking there is chance or possibility that a task can accessed code and data of another task hence to avoid this protection is implemented.

2) Explore five aspects of protection applied in segmentation.



- The protection is implemented internally in memory management system, both in segment as well as page translation mechanism.
- Protection in 80386DX has five aspects:-
- (i) Type checking:
 - Segmentation can incorporate type checking mechanisms to ensure that data accessed within a segment is of correct type.
 - As mentioned, W (writeable), R (Read), A (accessed), etc bits from type field specifies the usage of segment and restrict segment for particular use only.
- (ii) Limit checking:
 - 80386DX uses limit field of segment descriptor to prevent programs from addressing outside the segments.
 - for all type of segments, the value of the limit is one less than size (expressed in bytes) of the segments.
 - It ensures that accessed to data within segment do not exceed predefined boundaries thereby preventing buffer / stack overflows.

(iii) Restrictions of Addressable Domain:

- Segmentation can restrict the addressable domain of each segment, limiting the memory location that can be accessed within segment
- This helps prevent unauthorized access to sensitive data or system resources by ensuring that segments can access only allotted memory location.

(iv) Restrictions of Procedure Entry points:

- Segmentation can restrict the entry points to procedure or functions within each segment.
- By controlling which procedures can be invoked from specific segments, ensuring that only authorized code execution paths are allowed within each segment.

(v) Restrictions of Instruction Set:

- Segmentation can also restrict the set of instructions that can be executed within each segment.
- This involves specifying which CPU instructions are allowed to be executed within a segment based on intended functionality and security requirements of segment.

* Rules of protection :-

1] Access control :-

- Each ring has its own set of permission that access to system resource such as memory, IO ports.

2] privilege instruction

- A certain instruction can only be executed to high level (Ring 0 & sometimes Ring 1) while lower privilege level from executing them.

3] Memory protection

- Each process allocated its own virtual space and it enforces memory protection to prevent unauthorized access to other processes memory.

4] Exception handling :

- processor raises exception for illegal or privileged instruction exception in lower privilege levels ensuring that the system remains stable & secure.

3) Write a short note on CPL, DPL, RPL, EPL,
→ IOPL

Descriptor privilege Level (DPL):

- Descriptor contains field called the Descriptor Privilege Level (DPL). It is least privilege level at which a task may access that descriptor and the segment associated with that descriptor.
- It is contained in access right byte of descriptor of the segment.

Current Privilege Level (CPL):

- The 80386DX stores the descriptors in the internal cache for currently executing segments. Privilege levels of such descriptors are referred to as Current Privilege Level (CPL).
- This privilege level is also called as Task privilege level.
- It specifies privilege level of currently executing task.

Request Privilege Level (RPL):

- Selector contains field called the Requester's Privilege level (RPL). The RPL is intended to represent privilege level of procedure that originates a selector.
- RPL is the two least significant bits of selector.

EFFECTIVE PRIVILEGE LEVEL (EPL):

- When access to a new memory is desired, an Effective Privilege Level (EPL) is computed. This is the least privilege of CPL and RPL.
- EPL is defined as

$$EPL = \max \{ RPL, CPL \} \text{ (numerically)}$$

\therefore thus task become least privilege.

- for example,

$$\text{if } RPL = 2$$

$$CPL = 1$$

then $EPL = 2 \rightarrow$ task become least privilege.

INPUT/OUTPUT PRIVILEGE LEVEL (IOPL):

- In IOPL mechanism, for execution of JN, JNS, OUT, OUTS, CLI and STI instructions, the CPL of procedure or task must be the same or a lower number than the number represented by the IOPL bits. ($CPL \leq IOPL$)
- Any attempt by less privileged procedure to use these instructions results in general protection exception. These instruction are called sensitive instructions because they are sensitive to IOPL.
- Thus each task can have different IOPL, task can change IOPL but only procedure executing at PLO can change it otherwise it remains unaltered and do not result in any exception.

4) Explain how segment and page level protection function in combination.



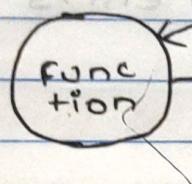
- When paging is enabled, the 80386 first evaluates the segment protection, then evaluates the page protection. If processor detects a protection violation at either the segment or page level, the requested operation cannot proceed; a protection exception occurs instead.

- It is possible to define a large data segments which has some sub-units that are read-only and other subunits that are read-write. In this case, the page directory (or page table) entries for the read-only subunits would have the U1S and R1W bits set to X0, indicating no write rights for all pages described by that directory entry.

Q. No.								Q. No.			
--------	--	--	--	--	--	--	--	--------	--	--	--

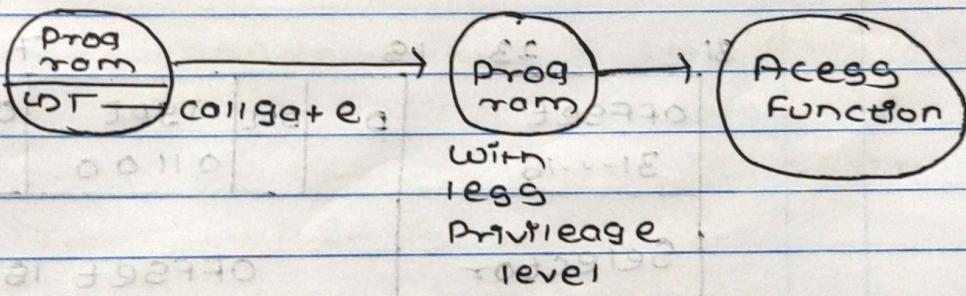
प. क.
Q. No.

Q4] a] What is call gate? Explain how it is used in calling function with higher privilege level.



and privilege level not sufficient we used

call gate



① Privilege level check

② Code Segment loading

③ Stack Setup

④ Control transfer

* Call Gate :-

- A call gate is a special type of descriptor that allows program running at certain level to call a procedure or function located in a different privilege level.

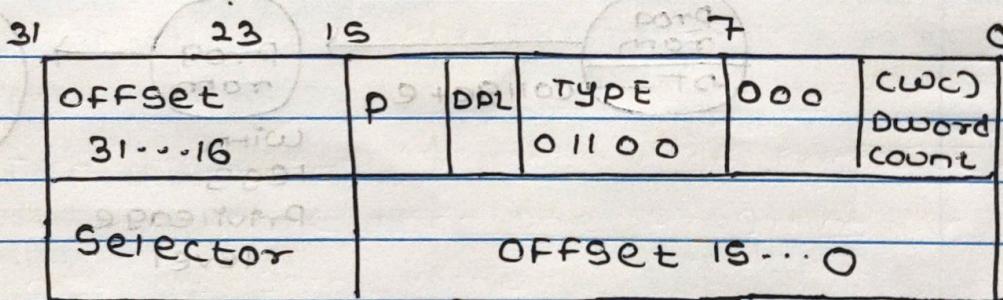
Do not write your name or seat no. below this line

Q. No.						Q. No.				
--------	--	--	--	--	--	--------	--	--	--	--

प्र. क्र.
Q. No.

- Call gate acts as an interface between code segment at different privilege level.
- Call has mainly two primary functions:
 - ① To specify privilege level of entry
 - ② To define an entry point of procedure.

*Call gate:-



Call gate contains two things :-

- 1] Selector:-
 - Points to descriptor for segment where Procedure is actually loaded

2] Offset:-

Offset of called Procedure on Segment

- 3] DPL:-
 - It specifies the privilege level to access code segment.

Q. No.						Q. No.				
--------	--	--	--	--	--	--------	--	--	--	--

प्र. क.
Q. No.

4] Descriptor type:-

It specifies the type of descriptor whether the gate is call gate or task gate.

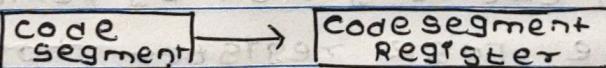
* Working of call gate:-

① Privilege level check:-

- The processor checks if the privilege level of calling program or task is sufficient to execute code segment specified in call gate.

② Code segment loading:-

- After checking privilege level the processor loads the code segment specified in call gate into code segment register (CS).



③ Stack setup:-

The processor sets up the stack segment (SS) and stack pointer (SP) values in call gate.

Q. No.					Q. No.			
--------	--	--	--	--	--------	--	--	--

प्र. क्र.
Q. No.

- ④ Control transfer - Finally the processor transfers control to ~~other~~ offset specified in call gate.

- 2] Explain role of Task register in multitasking and instructions used to

- Multitasking is ability of computer to run more than one program or task at same time.

*Role of Task Register in multitasking

- The Task Register (TR) identifies current executing task by pointing to TSS.
- The Task register has both visible portion which can read and change by instruction and invisible portion which cannot read by instruction.
- The Selector in visible portion is used to ^{SPECIFY} select TSS descriptor in GDT.

- while the invisible portion is used as a cache which hold base and limit value from TSS descriptor.
- Holding the base and limit in invisible portion of Task Register make execution of task faster, because processor does not need to repeatedly fetch this value from memory.

* Instruction used to modify and read

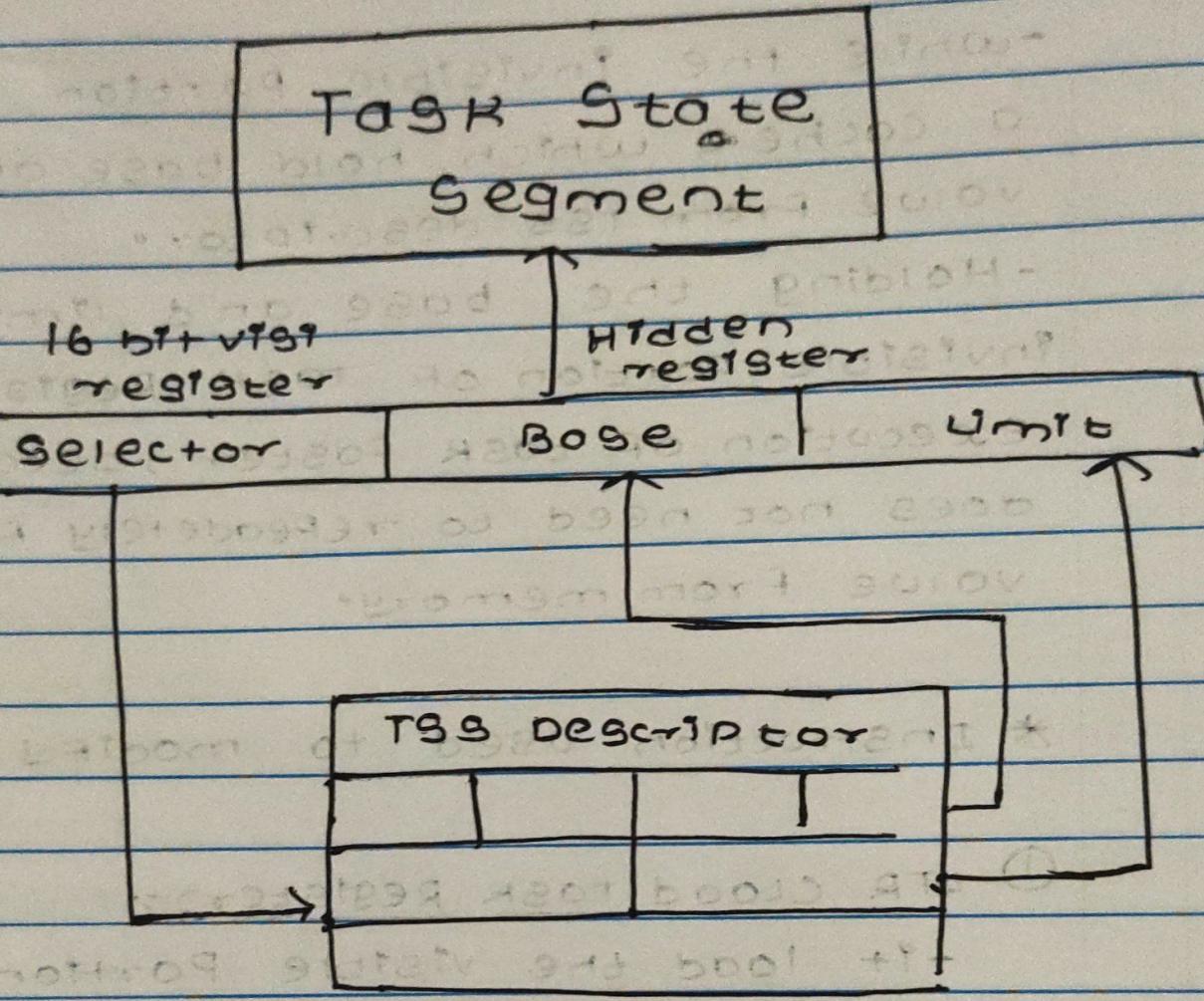
- ① LTR (Load Task Register) :-
 - It load the visible portion of task register with selector and invisible portion with information from TSS descriptor selected by selector. LTR is privilege instruction.
- ② STR (Store Task Register) :-
 - It store the visible portion of task register in general register or memory word.
 - STR is not privilege instruction.

Do not write your name or seat no. below this line

Q. No.

Q. No.

I. No.
2. No.



Task register

- * Explore the role of various field's in page level protection.

- Page-level protection :-
 when the processor is executing at supervisor level all pages are both readable and writable. When the processor is executing at user level only page's that belong to user level and are marked for read/write access are writable. Page that belong to supervisor level are neither readable nor writable from user level.

31	20	12	11	7	0
Page Frame Address	AVAIL	0	0	D A O O / / P	U R
31--12	S	W			

protection field's of page table entries.

- Two kind's of protection are related to pages :
 1. Restriction of addressable domain
 2. Type checking.

- 1] Restriction Addressable domain
- The concept of privilege for pages is implemented by assigning each page to one of two level's.

Heg-Border

① Explain need of protection mechanisms in 80386?

- Problem may occur when in multitasking operating system or multi user systems when two or more user attempts to read and change content of memory location at same time.
- The main purpose of protection feature of 80386 is to have multitasking with each task having its protected memory. Protection ensure that multiple task cannot access the code and data of other task.
- In case of multitasking, there is chance or possibility that a task can accessed code and data of another task hence to avoid this protection is implemented.

② Define function of type checking and limit checking.



* Type checking :-

- Whenever a selector is loaded into Segment Register, the corresponding target descriptor is copied from Descriptor table and loaded into descriptor cache.

1. Supervisor level ($U/S = 0$):

- for the operating system and other systems software and related data.

2. User level ($U/S = 1$):

- for applications procedures and data.

- The current level (U or S) is related to CPL if CPL is 0, 1 or 2 the processor is executing at supervisor level

- If CPL is 3 the processor is executing at user level.

- when if processor is executing at supervisor level all page's are addressable

- but when the processor is executing at user level only pages that belong to the user level are addressable.

2] Type checking :-

- At the level of page addressing two type's are defined.

1. Read - only access ($R/W = 0$)

2. Read / write access ($R/W = 1$)

- when the processor is execution at supervisor level all page's are both readable and writable.

- when the processor is executing at user level only page's that belong to user level are marked for

Page No.			
Date			

read / write access are writable.

- pages that belong to supervisor level are neither readable nor writable from user level.

- The descriptor indicate type of segment that is compared with Segment Register in which Selector is stored.
- Access is granted if type match or not interrupt 13 occurs.

TYPE Field of descriptor specifies

① Type of descriptor

② Intended usage of segment.

- The W, R, C (Confirming), A (Accessed), E (Expanded Down) bits from type field specify the usage of segment.

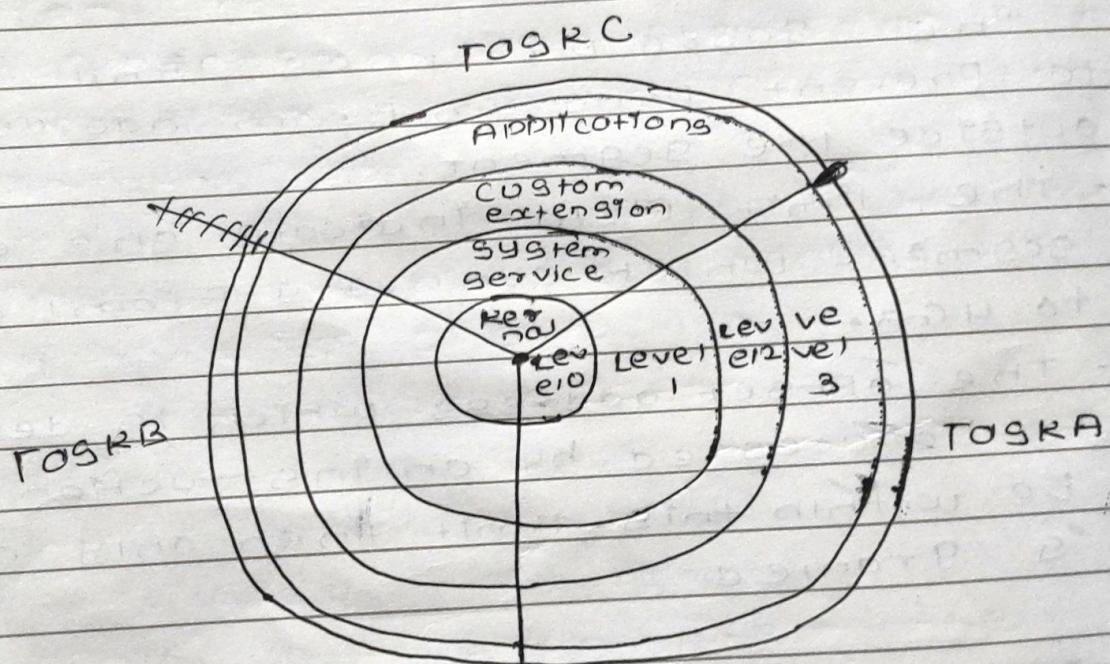
* Limit Checking :-

- The 80386 DX uses limit field to prevent program from addressing outside the segment.
- The limit field indicate size of segment which can vary from 1 byte to 4 GB.
- The offset address which is desired to be accessed by an instruction should be within this limit then only access is granted.
- If not general protection exception may occur-
 - ① For byte access (8 bit), offset address must be $L = \text{limit}$
 - ② For word access (16 bit) offset address must be $L = \text{limit} - 1$

(3) For doubleword (16 bits) the offset address must be $\Sigma = (\text{limit} - 8)$

(3) Explain different privilege level?

- The 80386 has four levels of protection which are optimized to support of a multitasking OS to isolate and protect user program from each other.
- Four protection levels are numbered from 0 to 3, whereas 0 is high privilege level and 3 is least privilege level.



① DPL :- Descriptor Privilege level)

- Descriptor contain field called Descriptor privilege level.
- It is least privileged level at which task may access that descriptor and segment associated with that descriptor.
- It is contain in accessed Right byte of descriptor of Segment.

② CPL :- Current Privilege levels :-

- the 80386DX stores the descriptor in internal cache for currently executing segment privilege level for such descriptor is called current privilege level.
- This is also known as task privilege level.
- It specifies privilege level for currently executing task.
- A task CPL can only be changed by control transfer through gate descriptor.

③ Requestor Privilege level CRPL :-

- Selector contain the field called Requestor privilege level. The
- RPL is intended to represent privilege level of procedure that originates a selector.
- RPL is two least significant bit of Selector.

④ Effective Privilege level (EPL):

- when accessed to new memory segment is desired the EPL is computed.

- EPL is defined as:

$$EPL = \max(CRPL, CPL)$$

- For example

$$\text{IF } RPL=2 \text{ and } CPL=1 \Rightarrow EPL=2$$

④ What is privileged instruction? Explain its significance with examples?

→ There are 19 privileged instruction supported by 80386. Privileged instruction are those that affect the segmentation and protection mechanism, alter interrupt flag, or perform peripheral I/O.

These are divided into two groups:-

① Privileged Instruction

② IOPL-sensitive Instruction

* Privileged Instruction:-

- This instruction affect the system data structure are come under 1st group.

- The instruction under this group must be executed when CPL is 0; otherwise 80386 generate exception.

Q. ④ what is privileged instruction? Explain its significance with example?

→ There are

- the privileged level instruction are type of instruction use to give the privilege to particular item.
- there are, 19 privileged instruction supported by 80386 privileged instruction which are those that affect the segmentation and protection mechanism after interrupt flag or perform peripheral I/O.
- these are divided into two group :-

① Privileged Instruction

② IOPL - sensitive instruction.

* Privileged Instruction

- This instruction affect the system data structure are come under 1st group
- The instruction under this grouped must be executed when CPL is 0; otherwise 80386 generated exception.

* Instruction :-

- ① HLT → Halt the processor
 - ② CLTS → clear task switched flag
 - ③ LGTS, LIDT, LLDT → Load GDT, IDT, LDT
 - ④ LTR → Load task Register
 - ⑤ LMSW → Load machine status word
 - ⑥ mov CRn, REG / mov REG, CRn
 - move to / from control register
 - ⑦ mov CRn, REG / mov REG, DRn
 - move to / from debug register
 - ⑧ mov TRn, REG / mov REG, TRn
 - move to / from test register.
- * IOPL ~~sensitive~~ instruction :-
- Here IOPL field in flag register defines right to use I/O related instruction
 - Hence instruction from this group is called sensitive instruction.

* Instruction :-

- | | Action |
|-------------|--------------------------|
| - IN, INS | Input data from I/O port |
| - OUT, OUTS | Output data to I/O port |
| - STI | Enable interrupt |
| - CLI | Disable interrupt. |

* privileged level instruction

1. HLT (Halt)

- The 'HLT' instruction halts the CPU until next external input is received
- e.g : HLT

2. CLTS

- clear Task-switched Flag
- The 'CLTS' instruction clears the Task-switched (TS) flag in the Control Register CR0
- e.g : CLTS

3. LDTR

- Local Descriptor Table Register.
- The LGDT

3. LGDT

- Load Global Descriptor Table
- The 'LGDT' instruction loads the base address and limit of the Global Descriptor Table (GDT) from memory into the GDTR register.

4. LIDT

- Load Interrupt Descriptor Table Register.
- The 'LIDT' instruction loads the base address & limit of the interrupt Descriptor Table (IDT) from memory into the IDTR register.
- e.g : LIDT [address]

5. LTR (Load Task Register)

- The 'LTR' instruction loads the Task Register (TR) with a segment selector that point to a task segment (TSS).

6. LLDT

- Load Local Descriptor Table Register.
- the LLDT instruction load Descriptor Table Register (CDTR) with a segment selector that point to a local Descriptor Table (LDT).

7. LMSW

- Load Machine status word
- the LMSW instruction loads the Machine status word modify certain bits of the CR0 Register.

8. IN

- (INPUT from port)
- The 'IN' instruction read a byte, word or double word from an I/O port into a CPU register.

9. OUT

- (Output to port)
- The 'OUT' instruction writes a byte word or double word from a CPU register to an I/O port.