**UNIT 3**

**MEMORY MANAGEMENT**

**PART A**

**MULTIPLE CHOICE QUESTIONS:**

1. The processes on the disk that are waiting to be brought into memory for execution form the\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (PgNo: 354) [L1]
2. Input queue
3. Output queue
4. Ready queue
5. Waiting queue

Answer: a

1. If it is not known at compile time that where the process will reside in memory, then the compiler must generate(PgNo: 355)[L1]
   1. Relocatable code
   2. Source code
   3. Absolute code
   4. Error

Answer: a

1. An address generated by the CPU is commonly referred to as (PgNo :355)[L1]
2. Logical address
3. Physical address
4. CPU address
5. Disk address

Answer:a

1. The set of all logical addresses generated by a program is (PgNo: 356)[L1]
2. Logical address space
3. Logical unit
4. Logical address set
5. Logical Space

Answer: a

1. The run-time mapping from virtual to physical addresses is done by a hardware device called(PgNo :356) [L1]
2. Memory management unit
3. Address space
4. Swap space
5. Compiler

Answer:a

1. All routines are kept on disk in a relocatable load format. The main program is loaded into memory and is executed whenever needed. This is called\_\_\_(PgNo :357)[L1]
2. Dynamic loading
3. Dynamic linking
4. Static loading
5. Static linking

Answer:a

* + 1. A process can be taken temporarily out of memory to a backing store and then brought back into memory for continued execution. The process ic called as\_\_\_(PgNo :358)[L1]

1. Swapping
2. Loading
3. Linking
4. Backing

Answer:a

* + 1. consists of all processes whose memory images are on the backing store /memory and are ready to run (Pgno:359)[L1]

1. Ready queue
2. Process queue
3. Wait queue
4. Static queue

Answer:a

* + 1. In allocation, each process is contained in a single section of memory that is continuous to the section containing the next process. (Pgno:359)[L1]

1. Contiguous memory allocation
2. Dynamic memory allocation
3. Partition
4. Fragmentation

Answer:a

* + 1. When a partition is free, a process is selected from the input queue and is loaded into the free partition. This is called (Pgno:362)[L1]

1. contiguous memory allocation
2. Dynamic memory allocation
3. Multiple partition method
4. Variable partition method

Answer:c

* + 1. In which partition scheme, the operating system keeps a table indicating which parts of memory are available and which are occupied. (Pgno:362)[L1]

1. contiguous memory allocation
2. Dynamic memory allocation
3. Multiple partition method
4. Variable partition method

Answer:d

* + 1. All memory available for user processes are considered as one large block of available memory is referred as hole.
    2. \_\_\_\_\_\_\_\_\_strategy allocates the ﬁrst hole that is large enough to accommodate the process (Pgno:363)[L1]

1. First fit
2. Best fit
3. Worst fit
4. Least fit

Answer:a,

* + 1. \_\_\_\_\_\_\_strategy searches the entire ordered list and allocates the smallest hole that is large enough to accommodate the process (Pgno:363)[L1]

1. First fit
2. Best fit
3. Worst fit
4. Least fit

Answer:b

* + 1. \_\_\_\_\_strategy produces the largest leftover hole, which may be more useful than the smaller leftover hole (Pgno:363)[L1]

1. First fit
2. Best fit
3. Worst fit
4. Least fit

Answer:c

* + 1. \_\_\_\_\_\_\_is a condition, where when there is enough total memory space to satisfy a request but the available spaces are not contiguous (Pgno:363)[L1]

1. First fit
2. External fragmentation
3. Internal fragmentation
4. Segmentation

Answer:b

* + 1. The process where the memory contents are shuffled, so as to place all free memory together in one large block is called as (Pgno:364)[L1]

1. Compaction
2. External fragmentation
3. Internal fragmentation
4. Segmentation

Answer:a

* + 1. The memory allocated to a process may be slightly larger than the requested memory. The difference between these two numbers is called (Pgno:363)[L1]

1. Compaction
2. External fragmentation
3. Internal fragmentation
4. Segmentation

Answer:c

* + 1. are the two techniques that permit the logical address space of the processes to be noncontiguous, thus allowing a process to be allocated with physical memory wherever such memory is available.(Pgno:364)[L1]

1. Segmentation and Paging
2. Fragmentation and Paging
3. Segmentation and Compaction
4. Paging and Compaction

Answer:a

* + 1. Which of the following segments are constructed by a compiler, whenever a program is compiled (Pgno:365)[L1]

1. Code
2. Global variables
3. Stacks used by each thread
4. Code ,global variables and stack used by each thread.

Answer:d

* + 1. In segmentation, each address is specified by \_\_\_\_\_\_\_\_\_\_\_\_(Pgno:364)[L1]

1. Segment number & offset
2. Offset & value
3. Value & segment number
4. Key & value

Answer:a

* + 1. In Paging, physical memory is breaked into fixed-sized blocks called (Pgno:367)[L1]

1. Frames
2. Pages
3. Segments
4. Thread

Answer:a

* + 1. In paging, logical memory is breaked into blocks of the same size called \_\_\_\_\_(Pgno:367)[L1]

1. Frames
2. Pages
3. Segments
4. Thread

Answer:b

* + 1. Every address generated by the CPU is divided into two parts. They are \_\_\_\_\_\_\_\_\_\_\_(Pgno:368)[L1]

1. frame bit & page number
2. page number & page offset
3. page offset & frame bit
4. frame offset & page offset

Answer:b

* + 1. The size of a page is typically of size \_\_\_\_\_\_\_\_\_\_\_\_(Pgno:368)[L1]

1. varied
2. power of 2
3. power of 4
4. power of 6

Answer:b

* + 1. Modern computer systems support a large logical address space of size\_\_\_\_\_\_\_\_\_(Pgno:378)[L1]

1. 232 to 264
2. 264 to 2128
3. 216
4. 268

Answer: a

* + 1. Address translation works from the outer page table inward, this scheme is also known as\_\_\_\_\_\_(Pgno:379)[L1]

1. Forward Mapped page table
2. Backward Mapped page table
3. Forward and backward page table
4. Page table

Answer: a

* + 1. \_\_\_\_\_\_Architecture supports a variation of one level paging.(Pgno:379)[L1]

1. ARM
2. VAX
3. Intel 32
4. SPARC

Answer: b

* + 1. \_\_\_\_\_\_\_ Architecture supports a variation of four level paging. (Pgno:380) [L1]

1. ARM
2. VAX
3. Intel 32
4. ULTRASPARC

Answer: d

* + 1. \_\_\_\_\_\_\_\_is an approach for handling address space larger than 32 bits.(Pgno:380)[L1]

1. Hashed page table
2. Inverted page table
3. Hierarchical page table
4. Forward mapped page table

Answer: a

* + 1. \_\_\_\_\_\_\_\_\_being the virtual page number available in hashed page table. (Pgno:380)[L1]

1. Index value
2. Hash Value
3. Location of the page
4. Pointers to the page

Answer:b

* + 1. Each entry in the hash table contains a \_\_\_\_\_\_\_of elements that hash to the same location. (Pgno:380)[L1]

1. Stack
2. Linked List
3. Arrays
4. Queue

Answer: b

* + 1. Each entry in the hash table contains a linked list of elements that hash to the same location to handle \_\_\_\_\_\_\_\_\_\_\_\_.(Pgno:380)[L1]

1. Cohesion
2. Collision
3. Coupling
4. Collaboration

Answer: b

* + 1. 64 bit address space uses \_\_\_\_\_\_\_\_\_\_\_\_\_page table. (Pgno:380)[L1]

1. Hashed
2. Inverted
3. Clustered
4. Hierarchical

Answer :c

* + 1. Clustered page table are useful for \_\_\_\_\_\_\_\_\_\_address space. (Pgno:381)[L1]

1. Sparse
2. Dense
3. Large
4. Small

Answer :a

* + 1. \_\_\_\_\_\_\_\_\_\_is the solution to keep track of how physical memory are used when page table consists of millions of entries. (Pgno:381)[L1]

1. Hashed page table
2. Inverted page table
3. Hierarchical page table
4. Forward mapped page table

Answer: b

* + 1. \_\_\_\_\_\_\_\_\_is the pair of inverted page table entry. (Pgno:382)[L1]

1. < process id, page number>
2. <page number, process id>
3. <process id, offset>
4. <offset, page number>

Answer:a

* + 1. Inverted page table is sorted by \_\_\_\_\_\_\_but look ups occur on \_\_\_\_\_\_\_\_\_\_\_.(Pgno:382)[L1]

1. Virtual address , Physical address
2. Physical address, virtual address
3. Virtual address, logical address
4. Physical address, logical address

Answer: b

* + 1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_is a memory management scheme that supports user view of memory.(Pgno:364) [L1]

1. Paging
2. Segmentation
3. Page replacement
4. Virtual memory

Answer: b

* + 1. The \_\_\_\_\_\_\_\_register points to the page directory for the current process.(Pgno:385)[L1]

1. CR3
2. CR4
3. CR
4. Address

Answer: a

* + 1. If Page\_size flag is set, the page directory points directly to the \_\_\_\_\_\_page frame, bypassing the inner page table. (Pgno: 385)[L1]

1. 4KB
2. 2KB
3. 4MB
4. 2MB

Answer: c

* + 1. As software developers began to discover the 4-GB memory limitations of 32-bit architectures, Intel adopted a\_\_\_\_\_\_\_\_\_\_\_, which allows 32-bit processors to access a physical address space larger than 4 GB. (Pgno:386)[L1]

1. Hash table
2. page address extension
3. page address
4. page size

Answer: b

* + 1. \_\_\_\_\_\_\_\_paging is used for 4-KB and 16-KB pages. (Pgno:388)[L1]

1. two-level
2. One-Level
3. Three-level
4. Four-level

Answer: a

* + 1. A \_\_\_\_\_\_\_\_\_\_\_\_address space is a collection of segments.(Pgno:364)[L1]

1. Physical
2. Logical
3. Virtual
4. Hashed

Answer: b

* + 1. Each entry in the segment table has a \_\_\_\_\_\_\_\_ and a \_\_\_\_\_\_\_\_\_\_\_\_.(Pgno:366)[L1]

1. Segment base, segment Limit
2. Segment id , offset
3. Segment base, offset
4. Offset, segment limit

Answer: a

* + 1. The segment base contains the starting \_\_\_\_\_\_\_\_\_\_\_where the segment resides in memory. (Pgno: 366)[L1]

1. Logical address
2. physical address
3. virtual address
4. hash address

Answer: b

**PART B**

**4 MARKS:**

1. Write short notes on hierarchical page table.(Pgno:378)[L1]

* Most modern computer systems support a large logical address space (232 to 264).
* Clearly, we would not want to allocate the page table contiguously in main memory.
* One simple solution to this problem is to divide the page table into smaller pieces.
* Hierarchical Paging is a paging scheme which consist of two or more levels of page tables in a hierarchical manner.
* The entries of the level 1 page table are pointers to a level 2 page table and entries of the level 2 page tables are pointers to a level 3 page table and so on.
* The entries of the last level page table are stores actual frame information. Level 1 contain single page table and address of that table is stored in PTBR (Page Table Base Register).

1. Write short notes on inverted page table.(Pgno:381)[L1]

* Inverted Page Table structure consists of one-page table entry for every frame of the main memory.
* So the number of page table entries in the Inverted Page Table reduces to the number of frames in physical memory and a single page table is used to represent the paging information of all the processes.
* Through the inverted page table, the overhead of storing an individual page table for every process gets eliminated and only a fixed portion of memory is required to store the paging information of all the processes together.
* This technique is called as inverted paging as the indexing is done with respect to the frame number instead of the logical page number.

1. Write short notes on hashed page table.(Pgno:380)[L1]

* A common approach for handling address spaces larger than 32 bits is to use a hashed page table, with the hash value being the virtual page number.
* Each entry in the hash table contains a linked list of elements that hash to the same location (to handle collisions).
* Each element consists of three ﬁelds:
  + (1) the virtual page number,
  + (2)the value of the mapped page frame, and
  + (3) a pointer to the next element in the linked list.
* The algorithm works as follows:
  + The virtual page number in the virtual address is hashed into the hash table.
  + The virtual page number is compared with ﬁeld 1 in the ﬁrst element in the linked list.
  + If there is a match, the corresponding page frame(ﬁeld2)is used to form the desired physical address.
  + If there is no match, subsequent entries in the linked list are searched for a matching virtual page number.

1. Discuss on X86-64 bit architecture.(Pgno:387)[L3]

* **x86-64** is the [64-bit](https://en.wikipedia.org/wiki/64-bit) version of the [x86](https://en.wikipedia.org/wiki/X86) [instruction set](https://en.wikipedia.org/wiki/Instruction_set).
* It introduces two new modes of operation, 64-bit mode and compatibility mode, along with a new 4-level [paging](https://en.wikipedia.org/wiki/Paging) mode.
* With 64-bit mode and the new paging mode, it supports vastly larger amounts of [virtual memory](https://en.wikipedia.org/wiki/Virtual_memory) and [physical memory](https://en.wikipedia.org/wiki/Physical_memory) than is possible on its 32-bit predecessors, allowing programs to store larger amounts of data in memory.
* In 64-bit mode, instructions are modified to support 64-bit operands and 64-bit addressing mode.
* As the full x86 16-bit and 32-bit instruction sets remain implemented in hardware without any intervening emulation, these older [executables](https://en.wikipedia.org/wiki/Executable) can run with little or no performance penalty, while newer or modified applications can take advantage of new features of the processor design to achieve performance improvements.

1. Discuss about variations in intel32-bit and intel 64-bit architecture.(Pgno:383)[L3]

The difference between 32-bit and 64-bit processors are:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **32-bit processors** | **64-bit processors** |
| Addressable space | It has 4 GB addressable space | 64-bit processors have 16 GB addressable space |
| Application support | 64-bit applications and programs won't work | 32-bit applications and programs will work |
| OS SUPPORT | Needed 32bit OS | It can run on 32 and 64 bit OS |
| Support of Multitasking | Not an ideal option for stress testing and multi-tasking. | Works best for performing multi-tasking and stress testing. |
| Memory Limits | 32-bit systems limited to 3.2 GB of RAM 32 bit Windows. It addresses limitation doesn't allow you to use full 4GB of Physical memory space. | 64-bit systems will enable you to store up to 17. Billion GB of RAM |

1. Explain briefly on Intel-32 bit Segmentation.(Pgno:384)[L3]

* The IA-32 architecture allows a segment to be as large as 4 GB, and the maximum number of segments per process is 16 K.
* The logical address space of a process is divided into two partitions. The first partition consists of up to 8 K segments that are private to that process.
* The second partition consists of up to 8 K segments that are shared among all the processes.
* Information about the first partition is kept in the local descriptor table (LDT); information about the second partition is kept in the global descriptor table (GDT).
* Each entry in the LDT and GDT consists of an 8-byte segment descriptor with detailed information about a particular segment, including the base location and limit of that segment.
* The logical address is a pair (selector, offset), where the selector is a 16-bit in which *s* designates the segment number, *g* indicates whether the segment is in the GDT or LDT, and *p* deals with protection.
* The offset is a 32-bit number specifying the location of the byte within the segment in question.

1. Explain briefly on Intel-32 bit paging.(Pgno:385)[L3]

* The IA-32 architecture allows a page size of either 4 KB or 4 MB. For 4-KB pages,IA-32 uses a two-level paging scheme in which the division of the 32-bit linear address
* The 10 high-order bits reference an entry in the outermost page table, which IA-32 terms the page directory.
* The page directory entry points to an inner page table that is indexed by the contents of the innermost 10 bits in the linear address.
* Finally, the low-order bits 0–11 refer to the offset in the 4-KB page pointed to in the page table.
* One entry in the page directory is the Page Size flag, which if set indicates that the size of the page frame is 4 MB and not the standard 4 KB.
* If this flag is set, the page directory points directly to the 4-MB page frame, by passing the inner page table; and the 22 low-order bits in the linear address refer to the offset in the 4-MB page frame.

1. Brief on the concept of page address extension(PAE).(Pgno:386)[L2]

* Software developers discovered the 4-GB memory limitations of 32-bit architectures, Intel adopted a page address extension (PAE), which allows 32-bit processors to access a physical address space larger than 4 GB.
* The fundamental difference introduced by PAE support was that paging went from a two-level scheme to a three-level scheme, where the top two bits refer to a page directory pointer table.
* PAE also increased the page-directory and page-table entries from 32 to 64 bits in size, which allowed the base address of page tables and page frames to extend from 20 to 24 bits.
* Combined with the 12-bit offset, adding PAE support to IA-32 increased the address space to 36 bits, which supports up to 64 GB of physical memory.
* It is important to note that operating system support is required to use PAE.

1. Explain compile time, load time, execution time with respect to address binding (Pgno:354)[L3]

The binding of instructions and data to memory addresses can be done at any step along the way:

* Compile time. If you know at compile time where the process will reside in memory, then absolute code can be generated. For example, if you know that a user process will reside starting at location *R,* then the generated compiler code will start at that location and extend up from there. If, at some later time, the starting location changes, then it will be necessary to recompile this code. The MS-DOS .COM-format programs are bound at compile time.
* Load time. If it is not known at compile time where the process will reside in memory, then the compiler must generate relocatable code. In this case, final binding is delayed until load time. If the starting address changes, we need only reload the user code to incorporate this changed value.
* Execution time. If the process can be moved during its execution from one memory segment to another, then binding must be delayed until run time. Special hardware must be available for this scheme to work

1. List the differences between logical and physical address (Pgno: 356)[L2]

|  |  |
| --- | --- |
| **LOGICAL ADDRESS** | **PHYSICAL ADDRESS** |
| It is the virtual address generated by CPU | The physical address is a location in a memory unit |
| et of all logical addresses generated by CPU in reference to a program is referred as Logical Address Space. | Set of all physical addresses mapped to the corresponding logical addresses is referred as Physical Address. |
| The user uses the logical address to access the physical address. | The user can not directly access physical address. |

1. Discuss the need for relocation register . (Pgno:356)[L3]

* The run-time mapping from virtual to physical addresses is done by a hardware device called the memory-management unit (MMU).
* We can choose from many different methods to accomplish such mapping. For the time being, we illustrate this mapping with a simple MMU scheme that is a generalization of the base-register scheme.
* The base register is now called a relocation register. The value in the relocation register is added to every address generated by a user process at the time the address is sent to memory.
* For example, if the base is at 14000, then an attempt by the user to address location 0 is dynamically relocated to location 14000; an access to location 346 is mapped to location 14346.



1. Write the purpose of dynamic loading (Pgno:357)[L1]

* The size of a process has been limited to the size of physical memory. To obtain better memory-space utilization, we can use dynamic loading. With dynamic loading, a routine is not loaded until it is called.
* All routines are kept on disk in a relocatable load format. The main program is loaded into memory and is executed.
* When a routine needs to call another routine, the calling routine first checks to see whether the other routine has been loaded.
* If it has not, the relocatable linking loader is called to load the desired routine into memory and to update the program’s address tables to reflect this change.
* Then control is passed to the newly loaded routine. The advantage of dynamic loading is that a routine is loaded only when it is needed.
* This method is particularly useful when large amounts of code are needed to handle infrequently occurring cases, such as error routines.

1. How are dynamic linked libraries are used (Pgno:357)[L2]

* Dynamically linked libraries are system libraries that are linked to user programs when the programs are run.
* This feature is usually used with system libraries, such as language subroutine libraries. Without this facility, each program on a system must include a copy of its language library (or at least the routines referenced by the program) in the executable image. This requirement wastes both disk space and main memory.
* With dynamic linking, a stubis included in the image for each library routine reference. The stub is a small piece of code that indicates how to locate the appropriate memory-resident library routine or how to load the library if the routine is not already present. When the stub is executed, it checks to see whether the needed routine is already in memory.
* If it is not, the program loads the routine into memory. Either way, the stub replaces itself with the address of the routine and executes the routine.

1. What is the use of swapping (Pgno:358)[L2]

* A process must be in memory to be executed. A process, however, can be swapped temporarily out of memory to a backing store and then brought back into memory for continued execution.
* Swapping makes it possible for the total physical address space of all processes to exceed the real physical memory of the system, thus increasing the degree of multiprogramming in a system.

1. Differentiate multiple partition method and variable partition method. (Pgno:362) [L2]

* In this multiple partition method, when a partition is free, a process is selected from the input queue and is loaded into the free partition.
* When the process terminates, the partition becomes available for another process.
* In the variable-partition scheme, the operating system keeps a table indicating which parts of memory are available and which are occupied.
* Initially, all memory is available for user processes and is considered one large block of available memory, a hole.
* As processes enter the system, they are put into an input queue. The operating system takes into account the memory requirements of each process and the amount of available memory space in determining which processes are allocated memory.
* When a process is allocated space, it is loaded into memory, and it can then compete for CPU time.
* When a process terminates, it releases its memory, which the operating system may then fill with another process from the input queue.

1. Write notes on internal fragmentation and external fragmentation (Pgno: 363)[L1]

* **Internal Fragmentation:**

Internal fragmentation happens when the memory is split into mounted sized blocks. Whenever a method request for the memory, the mounted sized block is allotted to the method. just in case the memory allotted to the method is somewhat larger than the memory requested, then the distinction between allotted and requested memory is that the Internal fragmentation.

* **External Fragmentation:**

External fragmentation happens when there’s a sufficient quantity of area within the memory to satisfy the memory request of a method. however the process’s memory request cannot be fulfilled because the memory offered is during a non-contiguous manner. Either you apply first-fit or best-fit memory allocation strategy it’ll cause external fragmentation.

1. Provide the 3 strategies for dynamic storage allocation problem. (Pgno:362)[L1]

* **First fit**. Allocate the first hole that is big enough. Searching can start either at the beginning of the set of holes or at the location where the previous first-fit search ended. We can stop searching as soon as we find a free hole that is large enough.
* **Best fit**. Allocate the smallest hole that is big enough. We must search the entire list, unless the list is ordered by size. This strategy produces the smallest leftover hole.
* **Worst fit**. Allocate the largest hole. Again, we must search the entire list, unless it is sorted by size. This strategy produces the largest leftover hole, which may be more useful than the smaller leftover hole from a best-fit approach.

1. Explain the fifty percent rule in fragmentation. (Pgno:363)[L3].

* Depending on the total amount of memory storage and the average process size, external fragmentation may be a minor or a major problem.
* Statistical analysis of first fit, for instance, reveals that, even with some optimization, given *N* allocated blocks, another 0.5 *N* blocks will be lost to fragmentation.
* That is, one-third of memory may be unusable! This property is known as the 50-percent rule.

1. Suggest the best solution for external fragmentation.(Pgno:364)[L3]

* External fragmentation exists when there is enough total memory to satisfy a request (from a process usually), but the total required memory is not available at a contiguous location i.e, its fragmented.

Solution to external fragmentation :

1) Compaction : shuffling the fragmented memory into one contiguous location.

2) Virtual memory addressing by using paging and segmentation.

1. What is the difference between segmentation and paging.(Pgno:365)[L2]

* The basic difference between paging and segmentation is that a page is always of fixed block size whereas, a segment is of variable size.
* Paging may lead to internal fragmentation as the page is of fixed block size, but it may happen that the process does not acquire the entire block size which will generate the internal fragment in memory.
* The segmentation may lead to external fragmentation as the memory is filled with the variable sized blocks.
* In paging the user only provides a single integer as the address which is divided by the hardware into a page number and Offset. On the other hands, in segmentation the user specifies the address in two quantities i.e. segment number and offset.
* The size of the page is decided or specified by the hardware. On the other hands, the size of the segment is specified by the user.
* In paging, the page table maps the logical address to the physical address, and it contains base address of each page stored in the frames of physical memory space. However, in segmentation, the segment table maps the logical address to the physical address, and it contains segment number and offset (segment limit).

1. Explain the structure of page table. (Pgno:368)[L3]

* Every address generated by the CPU is divided into two parts: a page number(p) and a page offset(d).The page number is used as an index into a page table.
* The page table contains the base address of each page in physical memory. This base address is combined with the page offset to deﬁne the physical memory address that is sent to the memory unit.

**PART C**

**12 MARKS:**

1. Draw the diagram of segmentation memory management scheme and explain its principle. (Pgno: 364)[L3]
2. Describe in detail about the common techniques for structuring the page table. (Pgno:378 )[L2]
3. Elaborate on INTEL 32-bit architecture. (Pgno:383 )[L1]
4. Discuss about ARM architecture. (Pgno:388 )[L3]
5. Describe in detail on Inverted page table with neat diagram. (Pgno:381) [L3]
6. Explain segmentation in detail. (Pgno:364 )[L3]
7. Explain paging in detail. (Pgno:366 )[L3]
8. Elucidate fragmentation and propose the solutions to avoid fragmentation.

(PgNo:363) [L3]

1. Illustrate memory allocation schemes in detail. (Pgno: 362)[L3]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*