

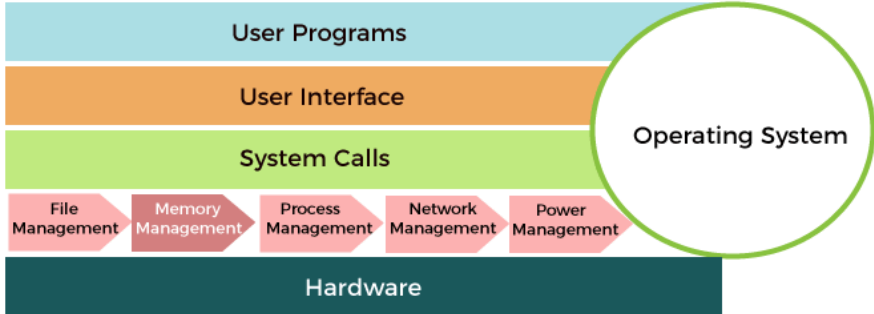


**Computer Engineering Department &**  
**Information Technology Engineering Department**

Academic Year: 2021-2022

Class: S.Y.B.Tech Sem.: 4 Course: OS

<b>Name</b>	<b>Pratik Pujari</b>		
<b>UID no.</b>	<b>2020300054</b>	<b>Class:</b>	<b>Comps C Batch</b>
<b>Experiment No.</b>	6		

<b>AIM:</b>	To perform an experiment on Memory Management
<b>THEORY:</b>	<p><b>What is memory management?</b></p> <p>Memory is the important part of the computer that is used to store the data. Its management is critical to the computer system because the amount of main memory available in a computer system is very limited. At any time, many processes are competing for it. Moreover, to increase performance, several processes are executed simultaneously. For this, we must keep several processes in the main memory, so it is even more important to manage them effectively.</p> <p style="text-align: center;"><b>Memory Management in OS</b></p>  <p>Memory management plays several roles in a computer system.</p> <p>Following are the important roles in a computer system:</p> <ul style="list-style-type: none"><li>○ Memory manager is used to keep track of the status of memory locations, whether it is free or allocated. It addresses primary memory by providing abstractions so that software perceives a large memory is allocated to it.</li></ul>



**Computer Engineering Department &**  
**Information Technology Engineering Department**

**Academic Year: 2021-2022**

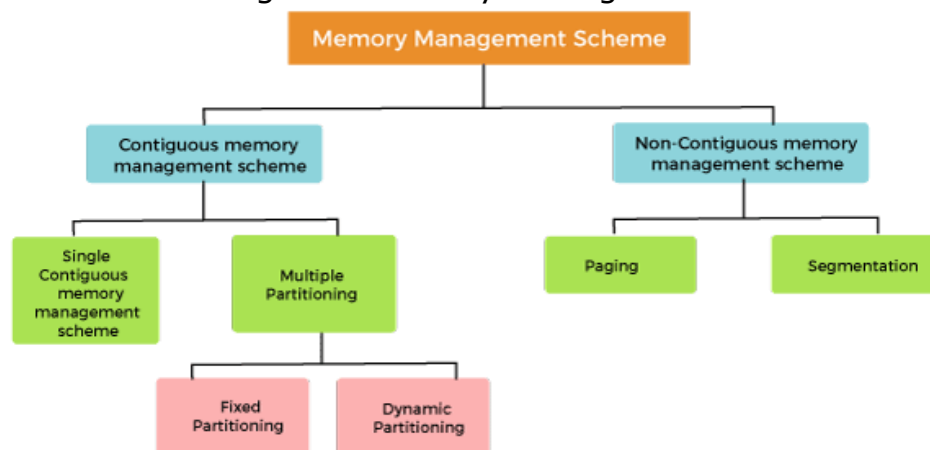
**Class: S.Y.B.Tech Sem.: 4 Course: OS**

- Memory manager permits computers with a small amount of main memory to execute programs larger than the size or amount of available memory. It does this by moving information back and forth between primary memory and secondary memory by using the concept of swapping.
- The memory manager is responsible for protecting the memory allocated to each process from being corrupted by another process. If this is not ensured, then the system may exhibit unpredictable behavior.
- Memory managers should enable sharing of memory space between processes. Thus, two programs can reside at the same memory location although at different times.

**Memory management Techniques:**

**The Memory management Techniques can be classified into following main categories:**

- Contiguous memory management schemes
- Non-Contiguous memory management schemes



**Classification of memory management schemes**

**Contiguous memory management schemes:**

In a Contiguous memory management scheme, each program occupies a single contiguous block of storage



**Computer Engineering Department &**  
**Information Technology Engineering Department**

**Academic Year: 2021-2022**

**Class: S.Y.B.Tech Sem.: 4 Course: OS**

	<p>locations, i.e., a set of memory locations with consecutive addresses.</p> <p>Single contiguous memory management schemes:</p> <p>The Single contiguous memory management scheme is the simplest memory management scheme used in the earliest generation of computer systems. In this scheme, the main memory is divided into two contiguous areas or partitions. The operating systems reside permanently in one partition, generally at the lower memory, and the user process is loaded into the other partition.</p> <p><b>Advantages of Single contiguous memory management schemes:</b></p> <ul style="list-style-type: none"><li>○ Simple to implement.</li><li>○ Easy to manage and design.</li><li>○ In a Single contiguous memory management scheme, once a process is loaded, it is given full processor's time, and no other processor will interrupt it.</li></ul> <p><b>Disadvantages of Single contiguous memory management schemes:</b></p> <ul style="list-style-type: none"><li>○ Wastage of memory space due to unused memory as the process is unlikely to use all the available memory space.</li><li>○ The CPU remains idle, waiting for the disk to load the binary image into the main memory.</li><li>○ It can not be executed if the program is too large to fit the entire available main memory space.</li><li>○ It does not support multiprogramming, i.e., it cannot handle multiple programs simultaneously.</li></ul> <p><b>Multiple Partitioning:</b></p> <p>The single Contiguous memory management scheme is inefficient as it limits computers to execute only one program at a time resulting in wastage in memory space and CPU time. The problem of inefficient CPU use can be overcome using multiprogramming that allows more than one program to run concurrently</p>
--	---



**Computer Engineering Department &**  
**Information Technology Engineering Department**

**Academic Year: 2021-2022**

**Class: S.Y.B.Tech Sem.: 4 Course: OS**

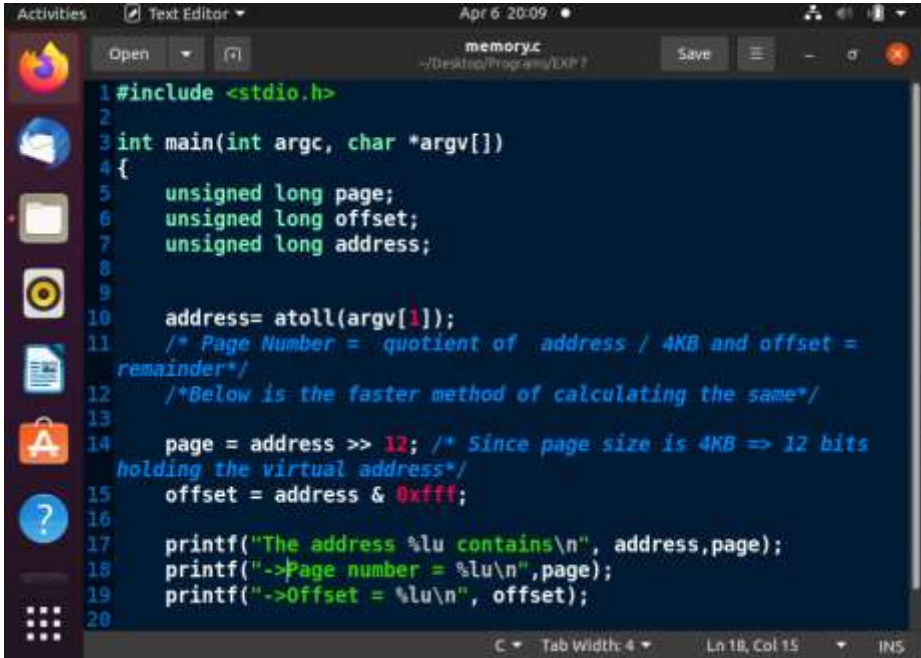
	<p><b>The multiple partitioning schemes can be of two types:</b></p> <ul style="list-style-type: none"><li>○ Fixed Partitioning</li><li>○ Dynamic Partitioning</li></ul> <p>Fixed Partitioning</p> <p>The main memory is divided into several fixed-sized partitions in a fixed partition memory management scheme or static partitioning. These partitions can be of the same size or different sizes. Each partition can hold a single process. The number of partitions determines the degree of multiprogramming, i.e., the maximum number of processes in memory. These partitions are made at the time of system generation and remain fixed after that.</p>
<b>EXPERIMENT</b>	
<b>PROBLEM STATEMENT:</b>	<p>Assume that a system has a 32-bit virtual address with 4-KB page size. Write a C program that is passed a virtual address (in decimal) on the command line and have it output the page number and offset for the given address. As an example, your program would run as follows:</p> <p><code>./a.out 19986</code></p> <p>Your program would output: The address 19986 contains: page number = 4 offsets = 3602</p>
<b>EXECUTION :</b>	<ol style="list-style-type: none"><li>1. Take the virtual address from the user as input on command line</li><li>2. Apply the logic to find page number and offset</li><li>3. Print the page number and offset</li></ol>



**Computer Engineering Department &**  
**Information Technology Engineering Department**

**Academic Year: 2021-2022**

**Class: S.Y.B.Tech Sem.: 4 Course: OS**

<b>CODE:</b>	<pre>#include &lt;stdio.h&gt; int main(int argc, char *argv[]) {     unsigned long page;     unsigned long offset;     unsigned long address;     address= atoll(argv[1]);     /* Page Number = quotient of address / 4KB and offset =     remainder*/     /*Below is the faster method of calculating the same*/     page = address &gt;&gt; 12; /* Since page size is 4KB =&gt; 12     bits holding the virtual address*/     offset = address &amp; 0xfff;     printf("The address %lu contains: \n", address);     printf("Page number = %lu\n",page);     printf("Offset = %lu\n", offset);     return 0; }</pre>
<b>OUTPUT:</b>	<p>Writing the Given above code</p> 



**Bharatiya Vidya Bhavan's**  
**Sardar Patel Institute of Technology**  
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India  
(Autonomous College Affiliated to University of Mumbai)

**Computer Engineering Department &**  
**Information Technology Engineering Department**

**Academic Year: 2021-2022**

**Class: S.Y.B.Tech Sem.: 4 Course: OS**

	<p>Compilation of the program:</p> <pre>gedit memory.c gcc -o memory memory.c</pre> <p>Output:</p> <pre>pratik@pratik-VirtualBox: ~/Desktop/Programs/EXP 7\$ ./memory 19986 The address 19986 contains -&gt;Page number = 4 -&gt;Offset = 3602</pre>
<p><b>CONCLUSION:</b> Learnt about the memory management and its topics. Learnt that the memory management provides ways to dynamically allocate portions of memory to programs at their request, and free it for reuse when no longer needed. Wrote a program about memory management that shows the page number and its offset.</p>	