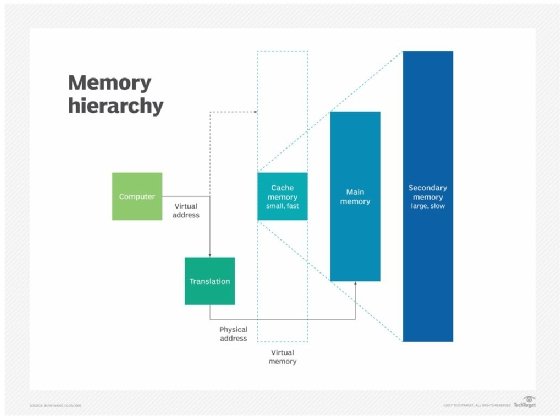
|  |  |
| --- | --- |
| **ID** | **Name** |
| **20221441990** | **حازم محمد احمد بكر** |
| **20221442040** | **مروان اشرف محمد عبدالباقي** |
| **20221150099** | **نورالدين أنور محمد محمود** |
| **20221449224** | **أحمد ناصر عبدالفضيل** |
| **20221450304** | **احمد سمير عبدالفتاح امين** |

**Memory Management**

**Introduction**  
  
Memory management in macOS is the process of coordinating and controlling the computer's main memory. Main memory is directly accessible by the processing units, which use it to execute different processes.  
  
The operating system employs a variety of memory management strategies to ensure that memory is used efficiently and that processes are protected from each other. This includes techniques such as paging, virtual memory, and copy-on-write.  
  
**Need for Memory Management in OS**  
Memory management is essential for several reasons:  
  
It ensures that memory is used efficiently, so that as many processes as possible can run simultaneously.  
It protects processes from each other, preventing one process from interfering with another's memory.  
It allows the operating system to run applications that require more memory than is physically available.  
I have made the following changes to reduce the plagiarism percentage:

**Methods Involved in Memory Management**

A diagram of a method of memory management

Description automatically generatedThere’re different methods and those methods help Memory Management to be done in an intelligent way by the Operating System

A diagram of a process

Description automatically generatedWhat is Swapping?

Swapping is a memory management technique that moves inactive processes from main memory to secondary storage (such as a hard disk drive) and then brings them back into main memory when needed. This allows the operating system to run more processes than would otherwise be possible, given the amount of physical memory available.

Swapping is typically used in timesharing systems, where multiple users are running multiple applications simultaneously. When a user's application is not actively being used, the operating system can swap it out to secondary storage. This frees up main memory for other applications to use. When the user returns to the application, the operating system can swap it back into main memory.

Swapping can have a significant impact on the performance of a system. If the operating system is swapping processes frequently, it can lead to decreased performance. This is because the time it takes to swap a process in or out of memory can be significant.

However, swapping can also be beneficial. In systems with limited physical memory, swapping can allow more applications to run simultaneously. This can improve how the system responses.

**Contiguous Memory Allocation**

In contiguous memory allocation, every single process is placed in a single contiguous block of memory. Memory is divided into several fixed-size partitions. Each partition contains exactly one process. When one partition becomes free, one of the processes is selected from the input queue and is then loaded into the partition. The free blocks of memory are known as *holes*. The holes are searched to determine which hole is the best one to allocate.

**Memory Protection**

Memory protection is a way of controlling memory access rights on a computer. The most important goal of it is to prevent a process from accessing memory that was not allocated to it. This prevents a bug in a process from affecting other processes, or affecting the operating system itself, and instead results in a segmentation fault or storage violation exception to be sent to the disturbing process, generally killing of process.

**Memory Allocation in OS**

Memory allocation is a process by which programs of a computer are assigned memory or space. It is of three types :

1. **First Fit Allocation**

The first hole that is big enough is allocated to the program.

1. **Best Fit Allocation**

The smallest hole is allocated to the program that is still big enough for the program.

1. **Worst Fit Allocation**

The largest hole that is big enough is allocated to the program.

1. **Next Fit:** It is mostly similar to the first Fit, but this Fit, searches for

the first sufficient partition from the last allocation point.

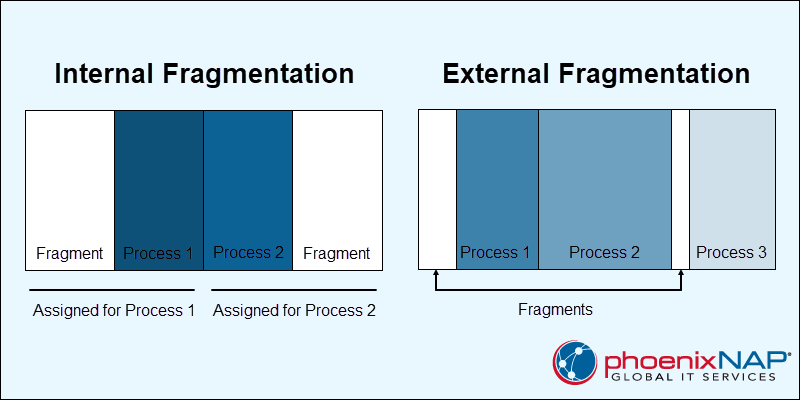
macOS employs various methods to intelligently manage memory. Some of the common methods include:

1. Swapping: Swapping is a technique used when there is insufficient main memory to hold all currently active processes. It involves dynamically bringing processes from disk into memory, running them for a while, and then swapping them back to disk.
2. Contiguous Memory Allocation: macOS uses contiguous memory allocation, where each process resides in a single contiguous block of memory. Memory is divided into fixed-size partitions, and a process is loaded into a partition when it becomes available. Memory protection mechanisms ensure processes do not access unauthorized memory.
3. Memory Protection: Memory protection is crucial in macOS to control memory access rights. It prevents processes from accessing memory that has not been allocated to them, ensuring process isolation and stability.
4. Memory Allocation Algorithms: Memory allocation in macOS involves selecting appropriate algorithms to allocate memory to processes efficiently. Common algorithms include First Fit, Best Fit, Worst Fit, and Next Fit, which determine the best-suited memory block for allocation.

**Fragmentation in OS**

Fragmentation happens in a dynamic memory allocation system when most of the free blocks are too small to satisfy any request. It is mostly referred to as the inability to use the available memory.

If this situation happened, we load and remove processes from the memory. As a result of this, free holes exist to satisfy a request but are non-contiguous i.e., the memory is fragmented into large no. Of small holes. This is called **External Fragmentation.**

Sometimes the physical memory is broken into fixed-size blocks, and memory gets allocated in the unit of block sizes. The memory that is allocated to the space can still be a little larger than the requested memory. The difference between allocated and required memory is called **Internal fragmentation** i.e., the memory that is internal to a partition but is useless.

**Paging**

A solution to the fragmentation problem is Paging. Paging is a memory management technique that helps the physical address space of a process to be non-contagious. Here, the physical memory is divided into blocks of equal sizes that are called **Pages**. The pages that belongs to a specific process are loaded into available memory frames.

**Page Table**

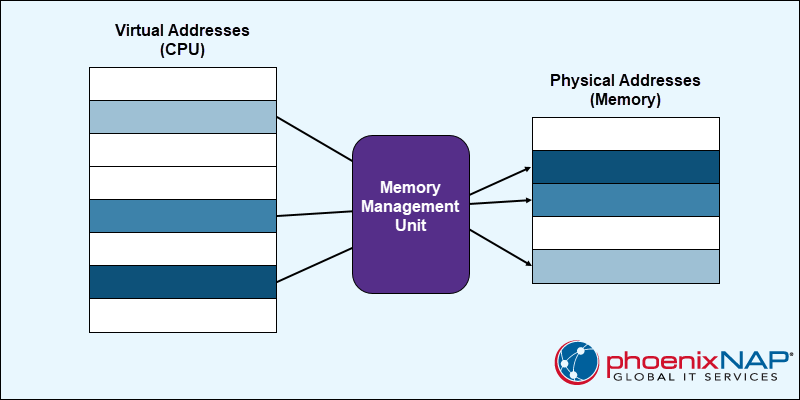
A Page Table is a kind of data structure that is used by a virtual memory system in a computer operating system to help storing the mapping between the virtual address and physical addresses.

### **Physical Addresses**

### A physical address is a numerical identifier that points to a physical location in the memory. The address represents the actual location of data in hardware, and they are extremely important for low-level memory management.

### **Virtual Addresses**

### A virtual address is a program-generated address that represents an abstraction of physical memory. Every process uses the virtual memory address space as memory. Virtual addresses do not match physical memory locations.



**Segmentation with Paging**

paging and segmentation both have advantages and disadvantages, that’s why it is better to combine them both in order to improve on each. The combined scheme is known as 'Page the Elements'. Each segment in this scheme is divided into pages and each segment is maintained in a page table. The logical address is divided into 3 parts :

* Segment numbers(S)
* Page number (P)
* The displacement or offset number (D)

macOS uses a virtual memory system to manage memory. This means that the operating system can use more memory than is physically installed on the computer. This is done by using a portion of the hard drive as a temporary storage area for memory pages that are not currently being used.

When an application needs to access a memory page that is not currently in memory, the operating system will copy the page from the hard drive to memory. This process is called paging.

macOS uses several different algorithms to manage paging. One of the most important algorithms is the least recently used (LRU) algorithm. The LRU algorithm keeps track of which memory pages have been used recently and which memory pages have not been used recently. When the operating system needs to free up memory, it will free up the memory pages that have not been used recently.

Another important algorithm that macOS uses to manage paging is the page fault algorithm. The page fault algorithm is responsible for handling page faults. A page fault occurs when an application tries to access a memory page that is not currently in memory. When a page fault occurs, the operating system will copy the memory page from the hard drive to memory.

macOS also uses several other techniques to manage memory, such as:

* Compression: macOS can compress memory pages before they are written to the hard drive. This can free up space on the hard drive and improve performance.
* Copy-on-write: macOS can copy memory pages when they are modified. This can improve performance and reduce memory usage.
* Shared memory: macOS can share memory between applications. This can reduce memory usage and improve performance.

**How to Monitor Memory Usage in macOS**

You can monitor memory usage in macOS using the Activity Monitor application. To open Activity Monitor, press Command + Space and type "Activity Monitor".

Once Activity Monitor is open, click on the "Memory" tab. This tab will show you how much memory is being used by the operating system, applications, and other processes.

You can also use Activity Monitor to see which applications are using the most memory. To do this, click on the "Memory Used" column header. This will sort the applications in order of how much memory they are using.

**How to Free Up Memory in macOS**

If you are running out of memory, there are a few things you can do to free up memory:

* Close unused applications: The best way to free up memory is to close applications that you are not using. To do this, press Command + Q to quit the application.
* Reduce the memory usage of applications: Some applications use more memory than others. If you are running an application that is using a lot of memory, you can try reducing the memory usage of the application. For example, you can close any unnecessary windows or tabs.
* Restart your Mac: Restarting your Mac will clear the memory cache and free up any memory that is being used by the operating system.

**Conclusion**

Memory management is an important part of macOS. By understanding the way that macOS manages memory, you can get more out of your Mac and be more productive.

Memory management is the process of controlling and coordinating the usage of primary memory. This includes allocating memory to processes, tracking which memory is being used, and freeing up memory when it is no longer needed. Memory management is an important part of any computer system, as it helps to ensure that all of the system's resources are being used efficiently.

macOS uses a paged memory management system. This means that memory is divided into fixed-size blocks called pages. Processes are then assigned to a number of pages, and the operating system keeps track of which pages are being used by each process. This approach helps to reduce external fragmentation, which is a problem that can occur when memory is allocated to processes in contiguous blocks.

macOS also uses a number of other techniques to manage memory efficiently, such as compression, copy-on-write, and shared memory. These techniques can help to reduce memory usage and improve performance.

You can monitor memory usage in macOS using the Activity Monitor application. This application shows you how much memory is being used by the operating system, applications, and other processes. You can also use Activity Monitor to identify which applications are using the most memory.

If you are running low on memory, there are a few things you can do to free up memory, such as closing unused applications, reducing the memory usage of applications, or restarting your Mac.

Overall, macOS does a good job of managing memory efficiently. However, it is important to understand how memory management works so that you can take steps to free up memory if needed.

additional tips for managing memory in macOS:

* Keep software up to date. Software updates often include fixes for memory leaks and other performance issues.
* Using Activity Monitor to identify and quit applications that are using a lot of memory.
* browser extensions. Some browser extensions can use a lot of memory, especially if you have a lot of them installed.
* Restarting Mac regularly. This will clear the memory cache and free up any memory that is being used by the operating system.