Paging and segmentation

Paging and segmentation are processes by which data is stored to, then retrieved from, a computer's storage disk.

Paging is a computer memory management function that presents storage locations to the computer's CPU as additional memory, called virtual memory. Each piece of data needs a storage address.

Segmentation is a virtual process that creates variable-sized address spaces in computer storage for related data, called segments. This process speed retrieval.

Managing computer memory is a basic operating system function — both paging and segmentation are basic functions of the OS. No system can efficiently rely on limited RAM alone. So, the computer's memory management unit (MMU) uses the storage disk, HDD or SSD, as virtual memory to supplement RAM.

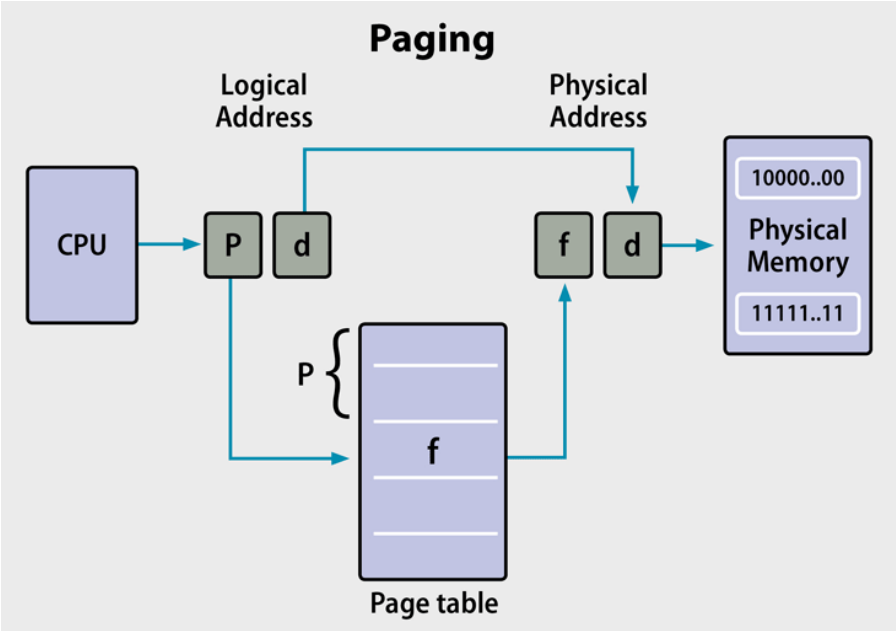
What is Paging?

As mentioned above, the memory management function called *paging* specifies storage locations to the CPU as additional memory, called virtual memory. The CPU cannot directly access storage disk, so the MMU emulates memory by mapping pages to frames that are in RAM.

Before we launch into a more detailed explanation of pages and frames, let's define some technical terms.

- **Page:** A fixed-length contiguous block of virtual memory residing on disk.
- **Frame:** A fixed-length contiguous block located in RAM; whose sizing is identical to pages.
- **Physical memory:** The computer's random access memory (RAM), typically contained in DIMM cards attached to the computer's motherboard.
- **Virtual memory:** Virtual memory is a portion of an HDD or SSD that is reserved to emulate RAM. The MMU serves up virtual memory from disk to the CPU to reduce the workload on physical memory.
- **Virtual address:** The CPU generates a virtual address for each active process. The MMU maps the virtual address to a physical location in RAM and passes the address to the bus. A virtual address space is the range of virtual addresses under CPU control.

- **Physical address:** The physical address is a location in RAM. The physical address space is the set of all physical addresses corresponding to the CPU's virtual addresses. A physical address space is the range of physical addresses under MMU control.



By assigning an address to a piece of data using a "page table" between the CPU and the computer's physical memory, a computer's MMU enables the system to retrieve that data whenever needed.

The Paging Process

A page table stores the definition of each page. When an active process requests data, the MMU retrieves corresponding pages into frames located in physical memory for faster processing. The process is called paging.

The MMU uses page tables to translate virtual addresses to physical ones. Each table entry indicates where a page is located: in RAM or on disk as virtual memory. Tables may have a single or multi-level page table such as different tables for applications and segments.

However, constant table lookups can slow down the MMU. A memory cache called the Translation Lookaside Buffer (TLB) stores recent translations of virtual to physical addresses for rapid retrieval. Many systems have multiple TLBs, which may reside at different locations, including between the CPU and RAM, or between multiple page table levels.

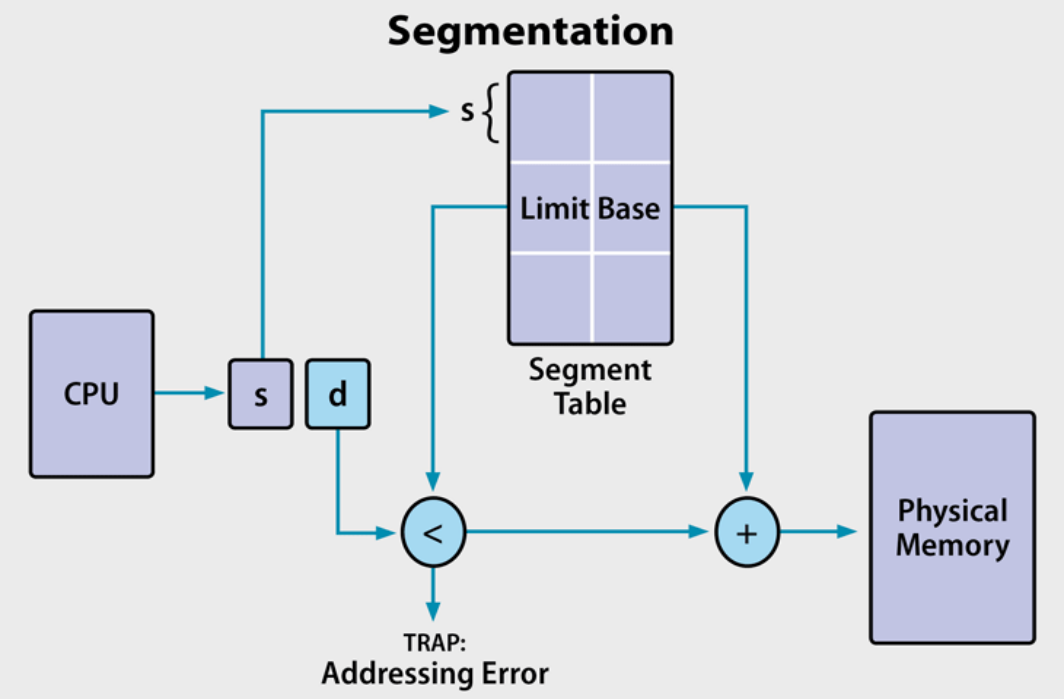
Different frame sizes are available for data sets with larger or smaller pages and matching-sized frames. 4KB to 2MB are common sizes, and GB-sized frames are available in high-performance servers.

An issue called hidden fragmentation used to be a problem in older Windows deployments (95, 98, and Me). The problem was internal (or hidden) fragmentation. Unlike the serious external fragmentation of segmenting, internal fragmentation occurred if every frame is not the exact size of the page size. However, this is not an issue in modern Windows OS.

What is Segmentation?

The process known as *segmentation* is a virtual process that creates address spaces of various sizes in a computer system, called segments. Each segment is a different virtual address space that directly corresponds to process objects.

When a process executes, segmentation assigns related data into segments for faster processing. The segmentation function maintains a segment table that includes physical addresses of the segment, size, and other data.



Segmentation speeds up a computer's information retrieval by assigning related data into a "segment table" between the CPU and the physical memory.

The Segmentation Process

Each segment stores the processes primary function, data structures, and utilities. The CPU keeps a segment map table for every process and memory blocks, along with segment identification and memory locations.

The CPU generates virtual addresses for running processes. Segmentation translates the CPU-generated virtual addresses into physical addresses that refer to a unique physical memory

location. The translation is not strictly one-to-one: different virtual addresses can map to the same physical address.

The Challenge of Fragmentation

Although segmentation is a high-speed and highly secure memory management function, external fragmentation proved to be an insurmountable challenge. Segmentation causes external fragmentation to the point that modern x86-64 servers treat it as a legacy application, and only support it for backwards compatibility.

External fragmentation occurs when unusable memory is located outside of allocated memory blocks. The issue is that the system may have enough memory to satisfy process request, but the available memory is not in a contiguous location. In time, the fragmentation worsens and significantly slows the segmentation process.

Segmented Paging

Some modern computers use a function called segmented paging. Main memory is divided into variably-sized segments, which are then divided into smaller fixed-size pages on disk. Each segment contains a page table, and there are multiple page tables per process.

Each of the tables contains information on every segment page, while the segment table has information about every segment. Segment tables are mapped to page tables, and page tables are mapped to individual pages within a segment.

Advantages include less memory usage, more flexibility on page sizes, simplified memory allocation, and an additional level of data access security over paging. The process does not cause external fragmentation.

Paging and Segmentation: Advantages and Disadvantages

Paging Advantages

- On the programmer level, paging is a transparent function and does not require intervention.
- No external fragmentation.
- No internal fragmentation on updated OS's.
- Frames do not have to be contiguous.

Paging Disadvantages

- Paging causes internal fragmentation on older systems.
- Longer memory lookup times than segmentation; remedy with TLB memory caches.

Segmentation Advantages

- No internal fragmentation.
- Segment tables consumes less space compared to page tables.

- Average segment sizes are larger than most page sizes, which allows segments to store more process data.
- Less processing overhead.
- Simpler to relocate segments than to relocate contiguous address spaces on disk.
- Segment tables are smaller than page tables, and takes up less memory.

Segmentation Disadvantages

- Linux only supports segmentation in 80×86 microprocessors: states that paging simplifies memory management by using the same set of linear addresses.
- Porting Linux to different architectures is problematic because of limited segmentation support.
- Requires programmer intervention.
- Subject to serious external fragmentation.

Key Differences: Paging and Segmentation

Size:

- **Paging:** Fixed block size for pages and frames. Computer hardware determines page/frame sizes.
- **Segmentation:** Variable size segments are user-specified.

Fragmentation:

- **Paging:** Older systems were subject to internal fragmentation by not allocating entire pages to memory. Modern OS's no longer have this problem.
- **Segmentation:** Segmentation leads to external fragmentation.

Tables:

- **Paging:** Page tables direct the MMU to page location and status. This is a slower process than segmentation tables, but TLB memory cache accelerates it.
- **Segmentation:** Segmentation tables contain segment ID and information, and are faster than direct paging table lookups.

Availability:

- **Paging:** Widely available on CPUs and as MMU chips.
- **Segmentation:** Windows servers may support backwards compatibility, while Linux has very limited support.

Reference

https://www.enterprisestorageforum.com/hardware/paging-and-segmentation/#:~:text=Segmentation%20is%20a%20virtual%20process,basic%20functions%20of%20the%20OS.