**Virtual memory**

Virtual memory is a [memory management](http://whatis.techtarget.com/definition/memory-management) capability of an OS that uses hardware and software to allow a computer to compensate for physical memory shortages by temporarily transferring data from random access memory ([RAM](http://searchstorage.techtarget.com/definition/RAM-random-access-memory)) to [disk storage](http://searchstorage.techtarget.com/definition/hard-disk). [Virtual address](http://whatis.techtarget.com/definition/virtual-address) space is increased using active memory in RAM and inactive memory in hard disk drives ([HDDs](http://searchstorage.techtarget.com/definition/hard-disk-drive)) to form [contiguous](http://searchsqlserver.techtarget.com/definition/contiguous) [addresses](http://searchstorage.techtarget.com/definition/address-space) that hold both the [application](http://searchsoftwarequality.techtarget.com/definition/application) and its data.

Functions of virtual memory

The purpose of virtual memory is to enlarge the [*address space*](https://www.webopedia.com/TERM/A/address_space.html), the set of addresses a program can utilize. For example, virtual memory might contain twice as many addresses as [main memory](https://www.webopedia.com/TERM/M/main_memory.html). A program using all of virtual memory, therefore, would not be able to fit in main memory all at once. Nevertheless, the [computer](https://www.webopedia.com/TERM/C/computer.html) could execute such a program by [copying](https://www.webopedia.com/TERM/C/copy.html)into main memory those portions of the program needed at any given point during execution.

To facilitate copying virtual memory into real memory, the operating system divides virtual memory into [*pages*](https://www.webopedia.com/TERM/P/page.html), each of which contains a fixed number of addresses. Each page is stored on a [disk](https://www.webopedia.com/TERM/D/disk.html)until it is needed. When the page is needed, the operating system copies it from disk to main memory, translating the virtual addresses into real addresses

Segmentation

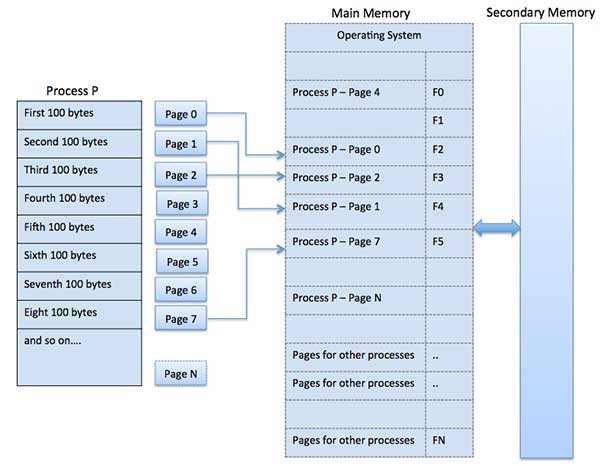
Segmentation – Computer memory is allocated in various sizes (segments) depending on the need for address space by the process. These segments may be individually protected or shared between processes. Commonly you will see what are called “Segmentation Faults” in programs, this is because the data that’s is about to be read or written is outside the permitted address space of that process.What is paging?

**paging** is a [memory management](https://en.wikipedia.org/wiki/Memory_management) scheme by which a computer stores and retrieves data from [secondary storage](https://en.wikipedia.org/wiki/Computer_data_storage#Secondary_storage)[[a]](https://en.wikipedia.org/wiki/Paging#cite_note-1) for use in [main memory](https://en.wikipedia.org/wiki/Computer_data_storage#Primary_storage).[[1]](https://en.wikipedia.org/wiki/Paging#cite_note-ostep-1-2) In this scheme, the operating system retrieves data from secondary storage in same-size [blocks](https://en.wikipedia.org/wiki/Block_(data_storage)) called [*pages*](https://en.wikipedia.org/wiki/Page_(computer_memory)). Paging is an important part of [virtual memory](https://en.wikipedia.org/wiki/Virtual_memory) implementations in modern operating systems, using secondary storage to let programs exceed the size of available physical memory.

A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard that's set up to emulate the computer's RAM. Paging technique plays an important role in implementing virtual memory.

Paging is a memory management technique in which process address space is broken into blocks of the same size called **pages** (size is power of 2, between 512 bytes and 8192 bytes). The size of the process is measured in the number of pages.

Similarly, main memory is divided into small fixed-sized blocks of (physical) memory called **frames** and the size of a frame is kept the same as that of a page to have optimum utilization of the main memory and to avoid external fragmentation.



**fragmentation**

In the default configuration of Windows, the pagefile is allowed to expand beyond its initial allocation when necessary. If this happens gradually, it can become heavily [fragmented](https://en.wikipedia.org/wiki/File_system_fragmentation) which can potentially cause performance problems.[[13]](https://en.wikipedia.org/wiki/Paging#cite_note-16) The common advice given to avoid this is to set a single "locked" pagefile size so that Windows will not expand it. However, the pagefile only expands when it has been filled, which, in its default configuration, is 150% the total amount of physical memory.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] Thus the total demand for pagefile-backed virtual memory must exceed 250% of the computer's physical memory before the pagefile will expand.

The fragmentation of the pagefile that occurs when it expands is temporary. As soon as the expanded regions are no longer in use (at the next reboot, if not sooner) the additional disk space allocations are freed and the pagefile is back to its original state.

Locking a pagefile size can be problematic if a Windows application requests more memory than the total size of physical memory and the pagefile, leading to failed requests to allocate memory that may cause applications and system processes to fail. Also, the pagefile is rarely read or written in sequential order, so the performance advantage of having a completely sequential page file is minimal. However, a large pagefile generally allows use of memory-heavy applications, with no penalties beside using more disk space. While a fragmented pagefile may not be an issue by itself, fragmentation of a variable size page file will over time create a number of fragmented blocks on the drive, causing other files to become fragmented. For this reason, a fixed-size contiguous pagefile is better, providing that the size allocated is large enough to accommodate the needs of all applications.

The required disk space may be easily allocated on systems with more recent specifications (i.e. a system with 3 GB of memory having a 6 GB fixed-size pagefile on a 750 GB disk drive, or a system with 6 GB of memory and a 16 GB fixed-size pagefile and 2 TB of disk space). In both examples the system is using about 0.8% of the disk space with the pagefile pre-extended to its maximum