Page Faults

When access is attempted to a virtual page with a PTE validity bit set to zero, it's called a page fault. The requested page isn't in physical memory, so something else is going to have to happen.

If all is well, then the missing page should be in the page file, and its location stored within the PTE. In that case, it's just a simple matter of loading the page from the page file and allocating it to a page frame in physical memory.

Another example might be the first time a reserved address is accessed; in this situation, there's naturally no physical memory associated with it, and so a page fault occurs. The OS responds to this by allocating a new empty page which, in most cases, doesn't involve reading anything from disk.

With plenty of free space available, the solutions to both of these situations are easy jobs. However, it's more difficult under low memory conditions, when the memory manager has to choose a resident page to remove in order to make space.

To do this, Windows looks at physical pages from all user processes to determine the best candidates to be replaced. Pages which are selected for replacement are written to the page file and then overwritten with the requested page data. Each of the PTEs are then adjusted, and address translation completes with the correct page in memory.

When discussing data being moved between physical memory and page files, it's worth talking about a process's **working set**, which is the set of its pages currently resident in physical memory, not counting any shared resources like DLLs. Windows Task Manager shows memory usage in terms of **working set**, which means that when a process's pages are swapped out to disk, its working set goes down, but its allocated virtual memory is not decreasing, which can lead to confusion.