Nicholas Prater

CS 475

Dr. Chiu

3 May 2023

OS Extra Credit: Windows 10 Core

Video games have been a part of my life for as long as I can remember. I have seen the progression in game speed, loading speed, and even graphics. I have never really thought about what exactly makes this possible each time a new console comes out. All I have heard is that “the processing power increased,” but what does that actually mean? Xbox has been my console of choice and currently I use the Xbox One S. It is an updated version of the 2013 Xbox One. The operating system (OS) that is used in this console is the Windows 10 core compared to the Windows 8 core that was originally released with the first Xbox One.

The Windows 10 core allows for “preemptive multitasking” (Karl-Bridge-Microsoft). Multiple threads are able to be processed at once. This tends to be really useful for an Xbox system that prides itself on efficient performance in terms of both speed and graphics. A system that isn’t allowed to run multiple processes at once will not make for an efficient gaming experience for the user. As a consumer of their product, I would prefer to not be waiting all day for my game to load or to experience gameplay latency, especially if I am playing online where it takes even more process time in order to synchronize all of the game participants. The Windows 10 core has something that is called fibers. These allow the application to handle the scheduling itself. Each thread has the ability to handle numerous fibers, which makes it more accessible for self-scheduling applications (Karl-Bridge-Microsoft). The OS’s treatment of processes and threads relates very similarly to what was discussed in class.

The Windows 10 core scheduler relies on user-mode scheduling (UMS) in order to handle scheduling on the OS. Each application has the ability to schedule on its own and also to switch between each thread without having to rely on the system scheduler (Karl-Bridge-Microsoft). There is a sense of fairness involved since each application has control over when they get scheduled. The UMS allows for a quicker, more efficient way of scheduling than a typical thread pool. Thread pooling is used to minimize the number of threads that are necessary and also aids in managing them subsequentially, but it can also result in starvation (Karl-Bridge-Microsoft). There are also some potential negative consequences of using fibers. Since the thread is there before the Dynamic Link Library (DLL) is loaded then a scenario could happen whereby the DLL ignores a "DLL_THREAD_DETACH" notification. This causes memory to be lost rather than being freed. Another problem might arise when there's no communication to the "DllMain" function and the DLL has no idea when a thread is destroyed or even created (Chen). Despite these pitfalls, the Windows 10 core does a much better job than its predecessor in managing scheduling.

The Windows 10 core memory manager uses both paging and segmentation. The use of paging refers to "Windows 10 virtual memory" (Tittel). It encourages an efficient use of physical RAM. Also, manual adjustments can be made to the paging files in order to decrease the size of the paging files in order to help avoid system crashes (Tittel). The Windows OS also utilizes segmentation. It divides its memory into two separate segments with separate private access so that they do not interfere with one another or other potential processes that try to access it ("Memory Management – Windows (1-2)"). The virtual memory is implemented in a way that provides privacy for each process so that there is no unwanted carryover from each other. The page table is used to translate the virtual address to its corresponding physical location. The virtual space for a 32-bit Windows system takes up four gigabytes and it is split into partitions (Alvinashcraft), which demonstrates that the Windows OS implements a hybrid of both paging and segmentation.

All in all, for most of my life I have been naïve as to what goes into these operating systems that power and help my game consoles run efficiently and effectively. As such, this research has given me a better appreciation of the amount of work that goes into creating these systems in order to make them production-level products. I have a newfound appreciation for something that I have always taken for granted.

Works Cited

Alvinashcraft. “Virtual Address Space (Memory Management) - Win32 Apps.” *Win32 Apps | Microsoft Learn,* learn.microsoft.com/en-us/windows/win32/memory/virtual-address-space. Accessed 2 May 2023.

Chen, Raymond. “Fibers Aren’t Useful for Much Any More; There’s Just One Corner of It That Remains Useful for a Reason Unrelated to Fibers.” *The Old New Thing,* 11 Oct. 2019, devblogs.microsoft.com/oldnewthing/20191011-00/?p=102989.

Karl-Bridge-Microsoft. “About Processes and Threads - Win32 Apps.” *Win32 Apps | Microsoft Learn,* learn.microsoft.com/en-us/windows/win32/procthread/about-processes-and-threads. Accessed 2 May 2023.

“Memory Management – Windows (1-2).” *Ruined*, 1 Apr. 2013, ruinedsec.wordpress.com/2013/03/23/memory-management-windows-1-2/#:~:text=Thus%2C%20paging%20in%20windows%20is,access%20by%20user%20space%20processes.

Tittel, Ed. “11 Tips to Improve Windows 10 Performance: TechTarget.” *Enterprise Desktop,* 24 Nov. 2020, www.techtarget.com/searchenterprisedesktop/tip/Make-Windows-10-performance-sing-by-tweaking-RAM-service-settings#:~:text=On%20most%20Windows%2010%20systems,the%20paging%20file%20somewhat%20smaller.