CS510 Languages and Low Level Programming: hmwkTitle

Due on April 29, 2016 at 11:59pm

Mark P. Jones Spring 2016

Konstantin Macarenco

The main benefit of implementing some of the OS features in hardware is performance and security. Specialized hardware will be in most of the cases many times faster than general software. Some tasks are simply not feasible without some level or hardware support.
Software approach reduces cost and complexity of hardware centric systems. And vice versa performance of software is increased by special purpose hardware.

Memory Management

Programming first computers without any memory management aid was a big headache. What to do when a task runs out of memory? How to feet larger applications into available RAM? How to protect one task from corrupting another's task memory? Etc.

Memory Management Infrastructure development was driven by the concept of Virtual Memory which required to support of multiple address spaces. Dynamic translation between virtual and physical addresses. This concept would be too slow and prone to security problems if implemented in pure software. Memory management hardware solves following issues related to Virtual Memory and Address spaces:

- Fast memory translation.
- Address Space Switch.
- Address Space isolation.
- Memory overflow protection (swapping).

Memory protection tries to solve problem of a process affecting other processes or the OS memory. Without hardware support this problem is impossible to solve since during execution cap current process is in total control and can do anything without proper protection mode. Protection rings, segmentation and paging enable this feature. OS creates page table and changes cr3 to point to the current page table. This is a privileged instruction and can be only done in ring0.

I see two possible solutions without hardware support (neither is as good as having hardware support)

- An OS provides virtualised environment (virtual machine that implements hardware protection). This approach raises some performance problems.
- OS can periodically scan memory for violation. This approach is more of a debugging tool, it is a lot less secure and doesn't prevent from interprocess memory access.

Context Switching

I was surprised to learn that context switch in modern OSs is a software feature. Intel task switch mechanism nowadays is used mainly to transfer from one protection level to another. Linux creates only on TSS(segmentation related feature) for this purpose and uses ESP0 and SS0 for this purpose. Hardware context switch is appeared to be slower less flexible and portable from x86 to x86_64.