Lecture 23: Memory Management

Stacks – Reserved part of CPU R/W main memory that saves the program counter and the register contents of a running task, and its R/W memory addresses. Physically, it is a block a successive memory locations accessible on a LIFO basis with a pointer that keeps track of where everything is that's running. One can have almost unlimited subroutine nesting and interrupt nesting since the task status and register information is stored. The operating system accesses the stack via the stack pointer whose registers contain the next unused address in the respective stack chain. Therefore interrupt handlers can be serviced by the CPU since the running task registers are stored at its stack pointer location.

Page frames – Program segment in non-contiguous memory usually done in a fixed size block. Dynamic allocation of memory is slow due to the capacitor charging/discharging, and is unusable with real-time systems due to the large source of real-time uncertainty, especially in hard real-time applications. Page stealing quite often occurs when no free frame is available, so a page table is needed, except that it costs much CPU overhead to maintain.

Swapping – When program size exceeds main memory, processes are moved out of memory to secondary storage device. However, there can be large context switch overhead when accessing processes in the secondary storage device. Schedulers can use it only when the task execution time is very large compared to this context switch overhead.

Overlaying – When program size exceeds main memory, new code and data are brought into memory replacing old code and data. Obviously, there is much opportunity for data loss and it's still inefficient.

Partitioning – Memory is separated into fixed sized blocks avoiding the need for swapping. Tasks reside in contigous partitions, but memory usage is inefficient and can be fragmented. It is useful when the number and size of tasks is well-known and fixed.

Memory locking – Recommended in real-time applications because paging, swaping, and overlaying is diminished, but memory contention can occur due to limited memory available. Execution times are much more predictable thereby enhancing scheduling performance.

	CPU ←→	Cache ←→	Main Memory ←→	Storage Disk
		(sram)	DRAM	magnetic/optical
access time:		10 nsec	100 nsec	10 usec
typical size:		16 KB	8 MB	1 GB

The Cache is the major source of real-time uncertainty: why?