Lectures 22: Operating System Memory Management Examples

I. Pentium Memory Management is a refined version of 80386. Based on $2^{32} = 4$GByte address spaces.

A. Address Spaces have four distinct views of memory
   1. Unsegmented unpaged memory has the logical address the same as the physical address. No translation slow down, but not good for multi-programming. Useful for high performance controllers, etc.
   2. Unsegmented paged memory has protection and management done via paging. Used in Berkeley UNIX.
   3. Segmented unpaged memory makes memory into a collection of logical spaces. Since segments can be any length, protection can be assigned to any size chunk, not just page size. Since its segment table is on-chip when a process is in RAM, there are no page table misses, and access time is predictable.
   4. Segmented paged memory has segments to control access of partitions, and paging to manage allocations within the partitions. Used by UNIX System V.

B. Segmentation
   1. Each virtual (logical) address consists of a 16-bit segment selector, and a 32-bit offset.
      a. Two bits of the segment selector specify the requestor privilege level, RPL.
         1) 0 = OS memory management, protection, access control, kernel; 1 = other OS system calls 2 = shared libraries, and user security programs, 3 user programs.
         2) An executing program may access only data segments with privilege levels greater than or equal to the program.
         3) Some instructions, e.g. memory management, can only be executed in level 0, and I/O can only be executed by levels 0 and 1.
      b. One bit of the segment selector is the Table Indicator that specifies whether the global segment table (GDT) that is shared by all processes, or the process’ local segment table (LDT) should be used for translating the segment selector number.
         1) Linux uses the GDT very little, just to make a big kernel code segment (PL=0), a big kernel data segment (PL=0), a user code segment(PL=3) and a user data segment(PL=3). Then Linux is done with GDT. It probably does not use LDTs, and relies on page tables.
         c. 13 bits for the index into the segment table. $2^{13} = 8192$ segments possible with each segment table.
      2. 64-bit segment table entries include 32-bit segment base address, 20-bit segment size, 1-bit granularity of size (1 byte or 4Kbytes), 1-bit segment present bit, and privilege/access bits.
      3. 32-bit base address from segment table entry + 32-bit offset from the virtual address = linear address.
      4. If no paging then linear address = physical address in RAM.

C. Paging is a one or two-level table lookup.
1. Top level table is a **page directory** consisting of 1024 32-bit entries that is located at an address pointed by a global register.

   a. MMU can have one page directory for all processes, one page directory for each process, or a combination.
   
   b. The page directory for the current process is always in RAM. Page tables may be in virtual memory.
   
   c. Page directory entries include 20-bit page frame address, page size bit (4KB or 4MB), present bit, cachable bit, write through/write back bit, dirty bit, and read/write bit.
   
   d. If page size is 4MB, then there are $2^{10}$ frames in 4GB, and the page directory serves as the page table. Only the upper 10-bits of the page frame address in the page directory entry are used to specify the frame #. A linear address would then have the format 10-bit page directory index, 22-bit (4MB) offset.
   
   e. If page size is 4KB, then there are $2^{20}$ frames in 4GB, and the page directory serves as a guide to up to 1024 page tables.

2. Second level tables are **page tables** which also contain 1024 32-bit entries that have the same format as the page directory entries.

   a. When page tables are used then linear address = 10-bit index into page directory page (that has the 20-bit frame # of the page table), 10-bit index of a page in the page table, 12-bit offset.

D. Translation Lookaside Buffer has 3 tables that map virtual addresses to physical addresses: 1) 64 entries for 4K data pages, 2) 8 entries for 4MB data pages, 3) 32 entries for instruction pages (both 4K and 4MB).

II. ARM Memory Management

A. Like Pentium the address space has four views of memory, is based on 4GB, and uses a two-level table organization (L1 and L2).

   1. **Supersections** consist of $2^{24} = 16$MB blocks of RAM. There are two VM formats: 1) 8-bit L1 table index, and 24-bit offset, or 2) 12-bit L1 table index, and 20-bit offset.

      a. **Sections** consist of 1 MB blocks of RAM. VM format is 12-bit L1 table index, and 20-bit offset.

   2. **Large pages** consist of 64KB blocks of RAM. There are two VM formats with both having a 12 bit L1 table index: 1) 4-bit L2 table index and 16-bit offset, or 2) 8-bit L2 table index and 12 bit offset.

   3. **Small pages** consist of 4KB blocks of RAM. VM format is 12-bit L1 table index, 8-bit L2 table index, and 12 bit offset.

   4. It is also possible to collect memory regions (any combination of sections, and page tables) into one of 16 **domains**. Access control can be applied to domains.

B. Like Pentium, ARM has a two-level table organization where the second level page table is optional. Both tables have 32-bit entries, each with a presence bit, and a two-bit format indicator.

   a. **First-level tables** (L1) are 4KB so they hold 1K entries. The two-bit format indicates whether the entry hold information about sections, supersections, or second-level tables.

      1) Supersection entries have an 8-bit base address with an additional 8 bits available to allow for larger physical memory. They also have bits for permissions, cachable, write buffer use, but not domains.

      8-bit entries are replicated 16 times because index portion overlaps the supersection offset by four bits.

      2) Section entries have a 12-bit base address, and bits for permissions, cachable, write buffer use, and domains.

      3) Page table entries have a 22-bit L2 base address and bits for the domain.

   2. **Second-level tables** (L2) hold translations for either large or small pages, and are 1K in size.

      1) Large table entries have a 16-bit base address, and bits for permissions, cachable, and write buffer use. Like supersections, each entry must be replicated 16 times because they overlap the offset by 4 bits.

      2) Small table entries have a 12-bit base address, and bits for permissions, cachable, write buffer use, and whether the region is executable or not.

C. Virtual Memory Translation

   1. TLB with 256 MRU virtual address to physical address mappings with 128 for data, 128 for instructions.

D. Access Control

   1. All table entries except the L1 page table entries, contain access control bits that designate no access, read only, read-write, or privileged access only.

   2. Domains allow multiple processes to use the same translation tables while maintaining some protection from each other. There are two kinds of domain accesses which allows very flexible memory protection for programs that access different memory resources.

      a. **Clients** must obey the access permissions of that domain.

      b. **Managers** control the behavior, including permissions of the domain.

      c. One program can be a client of some domains, and a manager of other domains.