Chapter 15 Address Translation

- In CPU Virtualization we focused on Limited Direct Execution (LDE)
- In virtualizing memory a similar strategy is used.
    - Efficiency means we must use hardware support.
    - Control means the OS ensures that no application is allowed to access any memory but its own.
    - Flexibility should allow programs to use their address space in whatever way they like.
- Hardware Base Address Translation aka address translation.
    - The hardware transforms each memory access.
        - changes the virtual address to a physical address.
    - The OS must get involved as well.
        - must set up the hardware so correct translations occur.

15.1 Assumptions

- Assume that the users address space must be placed contiguously in physical memory.
- The size of the address space is not too large.
    - specifically less than the physical memory.
- Each address space is exactly the same size.

15.2 An Example

- Consider a short code sequence that fetches a value from memory, increments it by 3, and stores it back in memory.

```
void func(){ int x = 30; x += 3;}
```

- When the function is compiled it may have corresponding assembly in the form of:

```
128: movl eax, ebx
132: addl eax, 3
135: movl ebx, eax
```

- The assembly code is laid out in the data section of the memory.
- The x variable is placed on the stack.
- From the programs perspective its address space starts at 0 and grows to a maximum of 16KB.
    - All memory referenced should be in those bounds
    - The OS has to relocate the process in memory that is transparent to the process.

15.3 Dynamic (Hardware-based) Relocation

- Base and bounds is a simple idea from the first time sharing machines of the late 1950's
    - also referred to as dynamic relocation.
- Need two hardware registers within each CPU
    - One called the base register
    - One called the bounds register.
    - The pairs allow us to place the address space anywhere in physical memory.
- The program is written as if it's going to be located at address 0.
    - When the program starts running the OS decides where in physical memory it should be loaded at.
        - Sets the base register to that value.
    - When the process is running any memory reference is translated by the processor.

- Each memory reference generated by the process is a virtual address.
    - The hardware adds the base register value to it to get the physical address.
- The bounds register is there to help with protection.
    - The processor will check that the memory reference is within bounds to make sure it is legal.
- MMU Memory Management Unit
    - The part of the CPU that deals with address translation.
- Bound registers can be defined in one of two ways.
    - Holds the size of the address space
    - Holds the physical address of the end of the address space.

15.4 Hardware Support

- Need two different CPU modes.
    - The OS runs in privileged/kernal mode.
    - Applications run in user mode.
    - A single bit indicates which mode the CPU is currently in.
- Support Base and Bounds registers
    - part of the MMU of the CPU
    - Changing should only be allowed in privileged mode.
- Exceptions
    - The cpu must be able to generate exceptions when a user program tries to access memory illegally.
    - Use an exception handler.

15.5 Operating System Issues

- There are new issues for the OS to handle with memory.
- There are points where the OS must get involved to implement

base and bounds version of virtual memory.
    - The OS must take action when a process is created.
        - finding memory space.
        - When a process is created the OS will have to search a data structure called a free list.
            - find room for the new address space and mark it as used.
    - Steps when a context switch occurs.
        - There are only one base and bound registers per CPU
        - The values for each differ for each running program
        - The OS must save and restore the values.
            - Stored in the process structure or process control block
- When a process is stopped it is possible for the OS to move an address space from one location in memory to another.
    - First the OS deschedules the process.
    - Then the OS copies from one location to another.
- The OS must provide exception handlers