Chapter 15: Mechanism: Address Translation

In developing the virtualization of the CPU, we focus on a mechanism called **limited direct execution** (or LDE). In virtualizing memory, we will pursue a similar strategy, attaining both efficiency and control while providing the desired virtualization.

The generic technique we will use, which you can consider an addition to our general approach of limited direct execution, is something that is referred to as **hardware-based address translation**, or just **address translation** for short.

**15.1 Assumptions**

We will assume for now that the user’s address space must be placed contiguously in physical memory. We will also assume that the size of the address space is not too big. Finally, we assume that the user’s address space must be placed contiguously in physical memory.

**15.2 An Example**

A picture containing diagram

Description automatically generated

OS uses the first 16KB of the physical memory for itself. It allocates the slot from 32KB to 48KB for the program to run and the other slots are free.

The problem is that how can we relocate this process in memory in a way that is transparent to the process?

**15.3 Dynamic (Hardware-based) Relocation**

We will need two hardware within each CPU. One is called the base register and the other is the bounds. This base-and-bounds pair is going to allow us to place the address space anywhere we’d like in physical memory and do so while ensuring that the process can only access its own address space.

In this setup, each program is written and compiled as if it is loaded at address zero. When a program starts running, the OS decides where in physical memory it should be loaded and sets the base register to that value. In the above example, the base register will be set at physical address 32KB.

Now, when any memory reference is generated by the process, it is translated by:

*Physical address = virtual address + base*

Each memory reference generated by the process is a **virtual address**. The hardware adds the contents of the base register to this address and the result is a physical address that can be issued to the memory system.

Transforming a virtual address into a physical address is exactly the technique we refer to as **address translation.** Since this relocation of the address happens at runtime and because we can move address spaces even after the process has started running, the technique is often referred to as **dynamic relocation**.

The bounds register is there to help with protection as the processor will first check that the memory reference is within **bounds** to make sure it is legal. In the above example, the bounds register will be set to 16KB. If process generates a virtual address that is outside of the bounds, or negative, it will raise error.

We call the part of the processor that helps with address translation the **memory management unit (MMU)**.

**15.4 Hardware Support: A summary**

CPU virtualization’s two modes: **privileged mode** (kernel mode) and **user mode**.

**processor status word** indicates which mode the CPU is currently running in.

The hardware must also provide the **base and bounds registers**.

The hardware should provide special instructions (**privileged** instructions) to modify the base and bounds registers, allowing the OS to change them when different processes run.

Table

Description automatically generated

the CPU must be able to generate **exceptions** in situations where a user program tries to access memory illegally.

**15.5 Operating System Issues**

1. the OS must take action when a process is created, finding space for its address space in memory. When a new process is created, the OS will have to search a data structure (often called a **free list**) to find room for the new address space and then mark it used.
2. The OS must do some work when a process is terminated, such as reclaiming all its memory, putting it back to free list.
3. The OS must also perform a few additional steps when a context switch occurs. When the OS decides to stop running a process, it must save the values of the base and bounds registers to memory, in some per-process structure such as the **process structure** or **process control block (PCB**).
4. The OS must provide exception handlers

Internal fragmentation means the space inside the allocated unit is not all used and thus wasted.