【OS】Day29

Class	Operating System: Three Easy Pieces
Date	@January 31, 2022

【Ch23】Complete Virtual Memory System

23.2 The Linux Virtual Memory System

The Linus Address Space

A Linux virtual address space consists of a user portion(where user program code, stack, heap, and other parts reside) and a kernel portion(where kernel code, stacks, heap, and other parts reside).

Like those other systems, upon a context switch, the user portion of the currently-running address space changes; the kernel portion is the same across processes.

Like those other system, a program running in user mode cannot access kernel virtual pages; only by trapping into the kernel and transitioning to privileged mode can such memory be accessed.

In classic 32-bit Linux, the split between user and kernel portions of the address space takes place at address 0xC0000000, or three-quarters of the way through the address space.

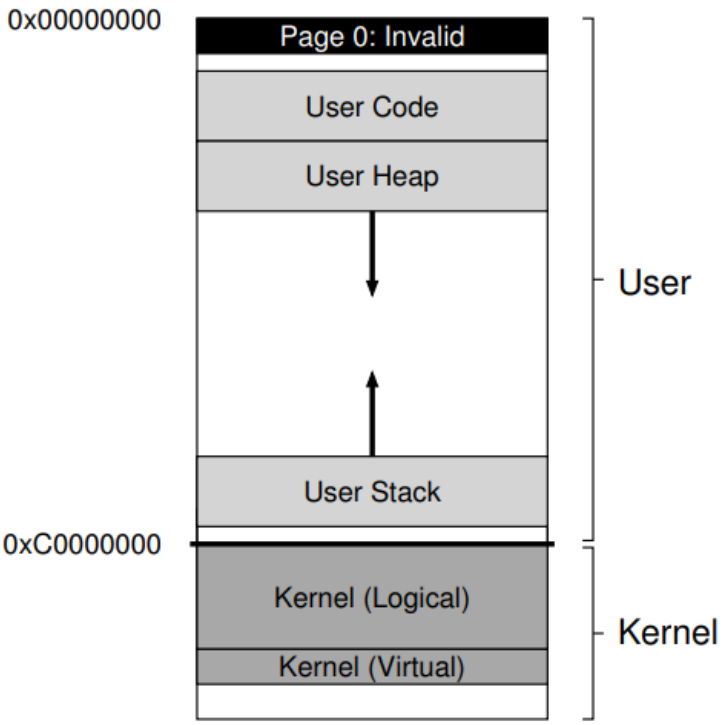

Figure 23.2: **The Linux Address Space**

One slightly interesting aspect of Linux is that it contains two types of kernel virtual addresses:

1. The first are known as kernel logical addresses. This is what we would consider the normal virtual address space of the kernel; to get more memory of this type, kernel code merely needs to call `kmalloc`.

   Most kernel data structures live here, such as page tables, per-process kernel stacks, and so forth.

   Unlike most other memory in the system, kernel logical memory cannot be swapped to disk.

2. The other type of kernel address is a kernel virtual address. To get memory of this type, kernel code calls a different allocator, `vmalloc`, which returns a pointer to a virtually contiguous region of the desired size.

   Unlike kernel logical memory, kernel virtual memory is usually not contiguous; each kernel virtual page may map to non-contiguous physical pages.