【OS】Day6

【Ch1】Introduction

When a program runs: it executes instructions

- The processor fetches an instruction from memory
- Decodes the instruction
- Executes the instruction

There is a body of software that is responsible for making it easy to run programs. *Allowing programs to share memory, enabling programs to interact with devices.*

The body of software is called the operating system.

The primary way the OS does this is through a general technique called virtualization. That is, the OS takes a physical resource(the CPU, a disk, or memory) and transforms it into a more general, powerful, and easy-to-use virtual form of itself. Thus, we sometimes refer to the operating system as a virtual machine.

The OS is sometimes also known as a resource manager. Each of the CPU, memory, and disk is a resource of the system; it is thus the operating system's role to manage those resources.

2.1 Virtualizing the CPU

The OS will create an illusion that the system has a very large number of virtual CPUs when we only have one.

Turning a single CPU into a seemingly infinite number of CPUs and thus allowing many programs to seemingly run at once is what we call virtualizing the CPU.

2.2 Virtualizing Memory

The model of physical memory is just an array of bytes; to read memory, one must specify an address to be able to access the data stored; to write memory, one must also specify the data to be written to the given address.

The OS virtualizes memory so that it is as if each running program has its own private memory, instead of sharing the same physical memory with other running programs.

2.3 Concurrency

Thread: A function running within the same memory space as other functions, with more than one of them active at a time.