

# 【OS】 Day6(2)

## 【Ch2】 The Process

The definition of a **process** is quite simple: it is **a running program**.

One often wants to **run more than one program at once**. A typical system may be seemingly running tens or even hundreds of processes at the time.

The OS creates this illusion by **virtualizing the CPU**.

By **running one process, then stopping it and running another**, and so forth, the OS can promote the illusion that **many virtual CPUs exist when in fact there is only one physical CPU**.

This basic technique, known as **time sharing of the CPU**, allows users to **run as many concurrent processes as they would like**; the potential cost is performance, as each will run more slowly if the CPU(s) must be shared.

To implement virtualization of the CPU, and to implement it well, the OS will need both **some low-level machinery and some high-level intelligence**.

- We call the low-level machinery **mechanisms**: mechanisms are low-level methods or protocols that **implement a needed piece of functionality**

*For example, we will learn later how to implement a context switch, which gives the OS the ability to stop running one program and start running another on a given CPU.*

- On top of these mechanisms reside some of the intelligence in the OS, in the form of **policies**: Policies are **algorithms for making some kind of decision** with the OS.

*For example, a scheduling policy in the OS will decide which program should the OS run when there are multiple. It will likely use historical information, workload knowledge, and performance metrics.*