Introduccion - SistOp

Lautaro Bachmann

Contents

Introduction to Operating Systems

So what happens when a program runs?

Well, a running program does one very simple thing: it executes instructions. the processor fetches an instruction from memory, decodes it and executes it **we have just described the basics of the Von Neumann model of computing2.**

the operating system

is in charge of making sure the system operates correctly and efficiently in an easy-to-use manner.

virtualization.

the OS takes a physical resource and transforms it into a more general, powerful, and easy-to-use virtual form of itself. **we sometimes refer to the operating system as a virtual machine.**

tell the OS what to do

the OS provides some interfaces (APIs) that you can call. A typical OS, in fact, exports a few hundred **system calls** that are available to applications. we also sometimes say that the OS provides a **standard library** to applications.

resource manager.

Each of the CPU, memory, and disk is a **resource** of the system; it is the operating system's role to **manage** those resources, doing so efficiently or fairly

Virtualizing The CPU

Turning a single CPU (or a small set of them) into a seemingly infinite number of CPUs and thus allowing many programs to seemingly run at once is what we call **virtualizing the CPU,** the ability to run multiple programs at once raises all sorts of new questions. if two programs want to run at a particular time, which should run? **This question is answered by a policy of the OS;** policies are used in many different places within an OS to answer these types of questions,

Virtualizing Memory

model of physical memory

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

virtualizing memory.

Each process accesses its own private virtual address space (sometimes just called its address space), which the OS somehow maps onto the physical memory of the machine. A memory reference within one running program does not affect the address space of other processes

Concurrency

We use this conceptual term to refer to a host of problems that arise, and must be addressed, when working on many things at once (i.e., concurrently) in the same program.

2.4 Persistence

In system memory, data can be easily lost, Thus, we need hardware and software to be able to store data persistently; such storage is thus critical to any system

file system;

it is responsible for storing any files the user creates in a reliable and efficient manner on the disks of the system.

2.5 Design Goals

One of the most basic goals is to build up some abstractions in order to make the system convenient and easy to use. One goal in designing and implementing an operating system is to provide high performance;

Another goal will be to provide protection between applications, as well as between the OS and applications. isolating processes from one another is the key to protection and thus underlies much of what an OS must do. operating systems often strive to provide a high degree of reliability.