

The operating system is (usually) that portion of the software that runs in kernel mode or supervisor mode. It is protected from user tampering by the hardware (ignoring for the moment some older or low-end microprocessors that do not have hardware protection at all). Compilers and editors run in user mode. If a user does not like a particular compiler, he[ ] is free to write his own if he so chooses; he is not free to write his own clock interrupt handler, which is part of the operating system and is normally protected by hardware against attempts by users to modify it.

Introduction

1.1.1. The Operating System as an Extended Machine

In this view, the function of the operating system is to present the user with the equivalent of an extended machine or virtual machine that is easier to program than the underlying hardware. How the operating system achieves this goal is a long story, which we will study in detail throughout this book. To summarize it in a nutshell, the operating system provides a variety of services that programs can obtain using special instructions called system calls. We will examine some of the more common system calls later in this chapter.

1.1.2. The Operating System as a Resource Manager

An alternative, bottom-up, view holds that the operating system is there to manage all the pieces of a complex system. Modern computers consist of processors, memories, timers, disks, mice, network interfaces, printers, and a wide variety of other devices. In the alternative view, the job of the operating system is to provide for an orderly and controlled allocation of the processors, memories, and I/O devices among the various programs competing for them.

(比如几个程序同时运行，操作系统要合理分配每个程序运行的时间。事实上应该是每个程序在1s内都运行几毫秒？)

An overview on MINIX 3

1.3.1. Processes

A process is basically a program in execution. Associated with each process is its address space, a list of memory locations from some minimum (usually 0) to some maximum, which the process can read and write. The address space contains the executable program, the program's data, and its stack. Also associated with each process is some set of registers, including the program counter, stack pointer, and other hardware registers, and all the other information needed to run the program.



Process的信息应该储存在一个 stack里面，以便下次恢复

*\*要不要创建管理员和多用户？*

1.3.2. Files

To provide a place to keep files, MINIX 3 has the concept of a directory as a way of grouping files together.



Every file within the directory hierarchy can be specified by giving its path name from the top of the directory hierarchy, the root directory.

At every instant, each process has a current working directory, in which path names not beginning with a slash are looked for. (例如打开文件)

file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

a project in Github:

<https://github.com/ybakos/cse7343-courseproject>



Tutorial Point on Operating System:

<https://www.tutorialspoint.com/operating_system/os_overview.htm>

分工：

|  |  |
| --- | --- |
| 许高远 | Others |
| 张津 | Error Handling |
| 刘杰 | Memory Allocation |
| 纪灵希 | Task Scheduling |
| 余思婧 | File System |
| 王思衡 | UI Design |