

第1章 Unix系统编程概述

什么是系统编程？

普通编程时，程序直接连到键盘、显示器、磁盘等设备

系统编程时，必须对系统的结构和工作方式有更深入的了解，要知道

内核提供哪些服务（系统调用）

如何使用系统调用

系统有哪些资源、设备、服务

处理器（Process）

输入输出（I/O）

进程管理（Process Manager）

内存（Memory）

系统空间（内核及其数据结构）

用户空间（用户程序）

设备（Device）

计时器（Timers）

进程间通信（Process Communication）

网络（Networking）

不同的资源和设备如何操作

什么是Unix系统编程？

用户角度：Unix能做什么

目录操作

登陆-运行程序-注销

操作指令：

ls

cd

pwd

mkdir、rmdir

待解决的问题：

目录在哪里？

用户在自己的主目录中意味着什么？

转到其它目录意味着什么？

会不会因为很多用户在同一目录中导致该目录过分拥挤？

如何编写一个改变当前目录的程序？

内核在目录树中扮演什么角色？

三个例子

网络桥牌

通信：某个用户或进程如何与其它用户或进程交换信息？

协作：同一时刻程序如何协调多个进程使他们能够没有冲突地访问共享资源？

网络访问：互相独立的计算机通过网络连接在一起，计算机中的程序如何来使用网络？

bc:Unix的计算器

预处理器bc将用户输入的表达式转化为逆波兰表达式，再通过管道（pipe）的通信程序交给dc

Unix系统编程很多场合下是为了解决好建立独立程序之间的连接和协作方式问题

从bc/dc到Web

实践：more

本书涉及哪些知识点

目标：理解系统编程

方法：

分析程序

学习系统调用

编程实现