

<<操作系统>>

图书基本信息

书名：<<操作系统>>

13位ISBN编号：9787302274896

10位ISBN编号：7302274894

出版时间：2012-1

出版时间：清华大学出版社

作者：乔德赫里

页数：640

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<操作系统>>

### 内容概要

《操作系统：原理与设计》以简单易懂的语言，全面而系统地介绍了操作系统的概念、原理和设计，包括CPU调度、设备调度、死锁、内存管理、文件系统以及系统的安全性等内容，并给出了丰富的C语言演示程序，所有这些C程序都在Linux操作系统下测试通过。

本书以单独章节介绍了设备管理、文件管理和低功耗系统设计，并详细阐述了如何防御对Linux系统的攻击，不仅是一本很好的操作系统教材，也是从事操作系统研究人员的很好参考用书。

<<操作系统>>

作者简介

作者：(印度)乔德赫里(Pabitra Pal Choudhury)

## &lt;&lt;操作系统&gt;&gt;

## 书籍目录

preface	xiii
acknowledgements	xv
1 introduction to the operating system	
1.1 os functions	
1.1.1 os acts as a user/computer interface	
1.1.2 interaction with os	
1.1.3 operating system commands	
1.1.4 operating system as efficient resource manager	
1.1.5 os upgrading	
1.1.6 operating system security and protection	
1.2 evolution of operating system	
1.2.1 serial processing	
1.2.2 batch processing	
1.2.3 multiprogramming or simple concurrent processing	
1.2.4 resource management	
1.2.5 operating system structure	
1.2.6 future operating system trends	
summary	
exercises	
2 os prerequisites	
2.1 important software resources to be managed by os	
2.2 interaction with os in mainframe systems (interrupts in general)	
2.2.1 the program status word	
2.2.2 different fields of psw	
2.2.3 controlling i/o	
2.2.4 the interrupt concept	
2.2.5 interrupt priority	
2.2.6 what other manufacturers (other than ibm) do?	
2.2.7 interrupt cycle	
2.2.8 differences between subroutine call and interrupt scheme	
2.3 example from pc environment and related discussion	
2.3.1 different ways to control pc hardware	
2.3.2 accessing the bios and dos	
2.4 current ' state-of-the-art ' scenario in os design	
2.5 fundamental concepts related to ipc	
2.5.1 message queues	
2.5.2 semaphores	
2.5.3 shared memory	
summary	
exercises	
3 concurrent processing	
3.1 introduction to concurrent processing	
3.2 process concept	

## &lt;&lt;操作系统&gt;&gt;

- 3.3 introduction to process control block
- 3.4 interaction of user ' s process with the system
  - 3.4.1 exec sys
- 3.5 introduction to concurrent program
- 3.6 distinction between a procedure call and a process creation
- 3.7 introduction to process environment
  - 3.7.1 introduction to process id
  - 3.7.2 shells or command interpreter
- 3.8 process state transitions
- 3.9 difference between user, daemon and kernel processes
- 3.10 interprocess communication
  - 3.10.1 file and record locking
  - 3.10.2 simple pipes and named pipes
  - 3.10.3 pipes and sockets
- 3.11 hierarchy of processes
  - 3.11.1 process creation
  - 3.11.2 processes and threads
- summary
- exercises
- 4 scheduling
  - 4.1 cpu scheduling algorithms
    - 4.1.1 allocation of different resources
    - 4.1.2 scheduling queues
    - 4.1.3 different scheduling algorithms
    - 4.1.4 algorithm evaluation
  - summary
  - exercises
- 5 discussion on concurrency control
  - 5.1 bernstein ' s concurrency conditions
    - 5.1.1 fork and join construct
    - 5.1.2 alternative to fork-join
  - 5.2 process states and precedence graphs
  - 5.3 dijkstra ' s concurrent statement in the form of fork and join constructs
  - 5.4 the critical section problem
    - 5.4.1 bounded buffer producerconsumer problem
    - 5.4.2 critical section problem description and their solution
  - 5.5 hardware solution to the critical section problem
  - 5.6 new synchronization tool—semaphore
    - 5.6.1 usage of semaphores
  - 5.7 classical process coordination problems
    - 5.7.1 bounded buffer problem
    - 5.7.2 the reader ' s and writer ' s problem
    - 5.7.3 dining philosophers ' problem
    - 5.7.4 sleeping barber problem
    - 5.7.5 cigarette smoker ' s problem

## &lt;&lt;操作系统&gt;&gt;

5.8 language constructs for synchronization and modularization

5.8.1 critical regions

5.8.2 conditional critical region

5.8.3 monitor construct

summary

exercises

6 deadlock

6.1 introduction

6.2 definition of deadlock

6.3 deadlock and starvation

6.4 resource allocation graph

6.5 ways to solve deadlock problem

6.5.1 tackling deadlock using deadlock prevention

6.5.2 deadlock avoidance

6.5.3 deadlock detection

6.5.4 difference between deadlock detection algorithm (multiple instances of each resource type) and safety algorithm

6.5.5 solution of deadlock under different scenarios

summary

exercises

7 main memory management

7.1 introduction

7.2 memory management schemes

7.3 partitioned allocation

7.4 logical vs physical address space

7.4.1 address binding

7.4.2 address translation

7.5 simple paging

7.6 the role of the long-term scheduler

7.6.1 problem assignment

7.7 implementation of the page table

7.8 multiple process using a text editor code

7.9 simple segmentation

7.10 details of segmentation

7.11 combined system

7.1.11 combination used in ge645 multics system

7.12 inner details of segmentation and paging for intel systems

summary

exercises

8 virtual memory technique

8.1 introduction

8.2 overlays technique

8.3 virtual memory

8.3.1 virtual memory concept

8.4 demand paging

## &lt;&lt;操作系统&gt;&gt;

- 8.4.1 performance of demand paging
- 8.5 page replacement algorithm
  - 8.5.1 fifo (simplest page replacement algorithm is first in first out)
  - 8.5.2 optimal replacement
  - 8.5.3 least recently used (lru) algorithm
  - 8.5.4 second chance replacement (with the help of reference bit)
  - 8.5.5 ad hoc page replacement algorithms
- 8.6 free frame allocation algorithm
  - 8.6.1 equal allocation
  - 8.6.2 proportional allocation (according to their needs rather than equal)
- 8.7 thrashing
- 8.8 working set
  - 8.8.1 prepaging
  - 8.8.2 i/o interlock
  - 8.8.3 page size anomaly
- 8.9 program structure
- summary
- exercises
- 9 spooler and disk scheduling
  - 9.1 introduction to the design of a spooling system
  - 9.2 relationship between spooling and job scheduling
    - 9.2.1 a printer spooler
  - 9.3 device management
  - 9.4 physical characteristics of disks
    - 9.4.1 disks
  - 9.5 fcfs disk scheduling
  - 9.6 searching an item in the disk with indexed sequential file directory
    - 9.6.1 shortest seek time first (sstf)
- summary
- exercises
- 10 file system architecture
  - 10.1 overview
  - 10.2 initial notions
    - 10.2.1 files
    - 10.2.2 directories
    - 10.2.3 different types of directories under the unix system
    - 10.2.4 the directory and file naming conventions
    - 10.2.5 difference between com and exe program
    - 10.2.6 abbreviations for path names
    - 10.2.7 paths are used
  - 10.3 attributes of unix files (for the beginners)
    - 10.3.1 the file owner and the group
    - 10.3.2 access permissions

## &lt;&lt;操作系统&gt;&gt;

- 10.3.3 creating a file
- 10.3.4 removing files
- 10.3.5 maintaining files
- 10.4 file system and their type
  - 10.4.1 evolution of file system over the years
- 10.5 introduction of ext2 fs
  - 10.5.1 the view of inodes from the point of view of a blocks group
  - 10.5.2 the allocated blocks
  - 10.5.3 user and group id
- 10.6 virtual file system (vfs)
  - 10.6.1 the linux virtual file system layer
  - 10.6.2 registration procedures of vfs
  - 10.6.3 registering the file systems in kernel
- 10.7 program designs
- appendix
- end notes
- summary
- exercises
- 11 device driver for operating system
  - 11.1 introduction
  - 11.2 initial notions for the design
  - 11.3 interfacing device drivers with the kernel
    - 11.3.1 character devices
    - 11.3.2 block devices
    - 11.3.3 network device
  - 11.4 theory to practice
- summary
- exercises
- 12 linux kernel and security
  - 12.1 introduction
  - 12.2 basics notions
    - 12.2.1 what are lkms
    - 12.2.2 what are system calls
    - 12.2.3 what is kernel symbol table
    - 12.2.4 how to transform kernel to user space memory
    - 12.2.5 ways to use user space like functions
    - 12.2.6 list of daily needed kernel space functions
    - 12.2.7 what is kernel daemon
    - 12.2.8 creating your own devices
  - 12.3 experiments with linux system calls
    - 12.3.1 how to intercept syscalls
    - 12.3.2 interesting system calls to intercept
    - 12.3.3 file system-related hacks
    - 12.3.4 process-related hacks
    - 12.3.5 network (socket)-related hacks
    - 12.3.6 virus writing with lkms



<<操作系统>>

12.4 directives to the system administrators for secured system

12.4.1 theory and ideas on lkm detector

12.5 the last word on the ' lkm story '

summary

exercises

13 role of os towards low power design

13.1 low power design from various angles

13.2 power management in embedded linux platform

13.2.1 dynamic power management

13.2.2 main generic points in the low power design

13.2.3 d-bus interface

13.2.4 hotplug mechanism

summary

exercises

bibliography

index

## 章节摘录

版权页：插图：However , it is difficult and tedious to communicate with the computer using this type of language. The hardware alone is very difficult to use. To circumvent the problem , different translator programs and other utilities were gradually developed known as system programs. NOW the question arises who is going to control these software and the machine? The answer is certainly the OS , as it implements certain frequently used functions that assist in program creation , management of files and control of I / O devices. In this context , a computer without an OS may be compared to a bus without a driver and conductor. Such a bus can still be run , but the passengers might fight over who will drive it and where it should go. Similarly, the users of a computer system without an OS might fight over the right to use the computer resources. As this text proceeds , we will see how crude the hardware is and how much control of the software is required to manage all the system resources. viz. the hardware and the system programs.

## <<操作系统>>

### 编辑推荐

《操作系统原理与设计》是大学计算机教育国外著名教材系列(影印版)。

<<操作系统>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>