

CH1

CH2

CH3

CH4 Process Management, Thread Management

Process

process state diagram

- New
- Ready
- Running
- Waiting
- Terminal

PCB

1.Process state 2.Program counter 3.CPU registers 4.CPU-scheduling information 5.Memory-management information 6.Accounting information 7.I/O status information (8.Process ID)

scheduling criteria

- CPU utilization
- Throughput
- Turnaround time
- Waiting time
- Response time

CPU Scheduling Algorithm

- **FCFS**
- SJF
 - Non-preemptive SJF->**SJF**
 - Preemptive SJF->**SRJF**
- **RR**
- **priority**
 - aging
- **multilevel queues**
- **multilevel feedback queues**

Multiple-Processor scheduling

- ASMP(沒什麼好設計的)
- SMP
 - load balancing
 - push migration
 - pull migration
 - processor affinity
 - soft affinity
 - hard affinity

Real-Time system scheduling

若有Priority Inversion,用Priority Inheritance解決

- Hard real-time(preemptive kernel)
 - Rate-Monotonic scheduling
 - EDF scheduling
- Soft real-time(preemptive kernel)
 - 不提供aging

Threads

- private
 - program counter
 - CPU registers set
 - stack
 - local variables
 - thread ID
- shared
 - code section
 - data section(global data)
 - heap
 - static local variables
 - other OS resources(open files, signals, I/O resources,etc.)
- Benefits
 - **responsiveness**
 - **resource sharing**
 - **economy**
 - **scalability**(utilization of multiprocessors architecture)
- Thread management
 - user thread
 - provide a library entirely in user space with no kernel support
 - implement a kernel-level library supported directly by the OS
 - kernel thread

- multithreading models
 - Many-to-One model
 - One-to-One model
 - Many-to-Many model
- 2 strategies of creating multiple threads
 - Asynchronous threading(父,兒之thread concurrently execute)
 - Synchronous threading(父thread要等兒thread做完)
- Pthreads library
 - Pthreads is a specification
 - Run on UNIX
 - Can't run on windows os

CH5 Deadlock Management

- necessary conditions
 - **mutual exclusion**
 - **hold and wait**
 - **no preemption**
 - **circular wait**
- resource-allocation graph
 - no cycle, no deadlock
 - 有cycle不一定有deadlock
 - if every resource only has exactly one instance, 有cycle就有deadlock
- methods for handling deadlocks
 - **deadlock prevent**
 - 破除mutual exclusion條件(辦不到)
 - 破除hold and wait條件
 - 破除no preemption條件
 - 破除circular wait條件: resource ordering
 - **deadlock avoidance**
 - banker's algorithm($O(n^2m)$,n: process,m: resource)
 - if system consisting of **m** resources of the same type with **n** processes running in the system
 - $1 \leq MAX_i \leq m$
 - $\sum_{i=1}^n MAX_i < n + m$
 - **deadlock detection and recovery**
 - detect it, and recover(允許系統進入deadlock)
 - detection algorithm($O(n^2m)$,n: process,m: resource)

- **ignore deadlock**
- Recovery from deadlock
 - process and thread termination
 - abort all deadlocked processes
 - abort one process at a time until the deadlock cycle is eliminated(盲目地砍一個)
 - resource preemption

CH6

CH7 Main Memory

Binding Time

- compile time
- loading time
- execution time

Memory Management methods in OS

- **Contiguous Memory Allocation**

- external fragmentation
 - First Fit
 - Best Fit
 - Worst Fit

- **Page**

- internal fragmentation
- page table
 - hierarchical paging
 - hashed page table
 - inverted page table

- **Segment**

- external fragmentation
- Base and Limit

- Paged Segment

CH8 Virtual Memory

- 實現Virtual Memory 技術: Demand Paging -pure demand paging -prepaging

Page Replacement Algorithm(沒有最差，只有最佳)

- **FIFO**(belady's anomaly)
- **OPT**(stack property)

- **LRU**(stack property)
- LRU-approximation(stack property)
 - **Additional reference bits usage**
 - **Second chance**
 - **Enhanced second chance**
- **LFU**(belady's anomaly)
- **MFU**(belady's anomaly)
- **Thrashing**
 - CPU utilization down
 - Paging I/O devices 異常忙碌
 - processes spends more time on paging I/O than normal execution
 - technique to handle Thrashing
 - **decrease multiprogramming degree**(已經thrashing)
 - **page fault frequency control**
 - **working set model**
- Allocation Kernel Memory
 - Buddy system
 - Slab allocation(has no internal, external fragmentation)

CH9 Massive Storage System

Hard Disk

- cylinder
- tracks
- sectors(磁碟控制器控制read, write之基本單位)
- Disk Access Time
 - Seek Time
 - Rotational latency
 - Transfer Time

Free-Space Management

- Bit vector
- Linked List
 - Grouping
 - Counting

File Allocation Methods

- Contiguous Allocation
- Linked Allocation

- 變形: FAT
- Indexed Allocation
 - Linked scheme
 - Multilevel index
 - Combined scheme(UNIX i-Node structure)

HDD scheduling(沒有最好與最差之法則)

- **FCFS**
- **SSTF**
- **SCAN**
 - elevator
- **C-SCAN**
- **LOOK**
 - elevator
- **C-LOOK**

RAID

- improvement of reliability via redundancy
 - mirror
 - parity check
- improvement in performance via parallelism
 - data striping
 - bits-level
 - block-level
- RAID0(N部)
 - block-level striping
- RAID1(mirror)(N/2部)
- RAID2(ECC-Error-Correcting Code)
 - 沒有實際產品
- RAID3(ECC-Error-Correcting Code)(N+1部)
 - bit-level striping
 - parity check
- RAID4(ECC-Error-Correcting Code)(N+1部)
 - block-level striping
 - parity check
- RAID5(ECC-Error-Correcting Code)(N+1部)

- RAID6(ECC-Error-Correcting Code)(N+2部)
- RAID1+RAID0(更好)
- RAID0+RAID1

File Directory Structure

- Tree-structured Directory
- Acyclic Graph Directory
- General Graph Directory(允許有cycle)

File Access Control

- Owner, Group, Other
- RWX(Read, Write, eXecute)
- command: `chmod 755 file`

Consistency Semantic

- UNIX semantic
 - 訂票系統
- Session semantic
 - 網站上的檔案提供下載讓user填寫
- Immutable-Shared-Files semantic
 - 總經理公告文件第3009號

NAS(Network-Attached Storage)

- File-based operation
- 會占用網路頻寬

SAN(Storage-Area Network)

- Block-based operation
- private network
- 不佔用一般網路頻寬