ch1

1.請說明並比較Trap及Interrupt

Interrupt屬於非同步發生的事件，在任何時間都可能發生且與processer在執行

的東西毫無關連，通常它由輸出入裝置，處理計時器，或時序產生。

Trap屬於同步狀態，通常由執行某一特別的指令。Trap可以藉由執行相同資料及

狀態下的程式而重複產生。其例有無效記憶體存取或除以零。

2.請解釋 Dual-Mode Operation。

Dual-Mode分為下列兩種：

1.Kernel mode

代表此時kernel 握有控制權，可以執行特權指令.

2. User mode

代表此時是user process握有控制權，不允許執行特權指令.

Dual-Mode之運行，需要額外的HW support.

3.在作業系統中請說明 multiprogramming及time sharing的意義，並比較其異同。

Multiprogramming(多元程式設計)

允許多個Process存在系統同時執行，運用CPU Scheduling技術，讓CPU在多個Processes之間切換執行，用以提高CPU utilization.

(eg. 當執行中Process因某些事件發生無法往下執行像是Wait for I/O complete，則CPU可切給其它Processes執行，故可確保CPU”always Busy”)

time sharing

是multiprogramming的一種，稱multitasking.

CPU在多個任務之間轉換頻繁，以致於在整個過程中，User都可與自己的程式進行互動。目的在於response time短、對user一律公平對待、讓每位user認為computer是為自己在執行的，始專屬於自己的

Ch2

1.在作業系統中請說明 Microkernel system architecture，有何優缺點?

將kernel中較不重要的服務項目移除，改成以system program或user process型式在user mode中執行以提供服務，如此可得出一個較小的kernel.

優點

1.加入及刪除容易﹝因為是Run在user mode，且不會對microkernel造成影響，即使有，修改幅度也很小﹞。

2. 安全性高

3. 可靠度高

4. 移植性高

缺點：service平均response time較久(效能差)

2.請解釋Virtual Machines 的觀念和做法，有何優缺點?

利用software simulation技術，建立一個和底層硬體一樣的功能介面。

做法：

1. 透過CPU Scheduling建立多顆CPUs效果

2. 透過Virtual memory 創造更大的memory space

3. 透過Spooling建立多套Virtual I/O Device.

優點：作為測試開發中的OS附載平台、節省成本。

缺點：不易製作、效能比real machine差。

3. 何謂policy？何謂mechanism，將policy和mechanism分開有何好處？

Policy : what will be done?

Mechanism : how to do?

Mechanism和Policy必須分開使系統容易修改。

由於mechanism和policy分開考慮，我們可以在不變更mechanism之下來改變policy使得系統更有彈性。

4.為何要什有system call?如何實作?呼叫時參數傳遞有哪些方法?這些方法有何優缺點?

自己想!!!

5. 請解釋 Operating-system generation(SYSGEN)?

Operating systems are designed to run on any of a class of machines; the system must be configured for each specific computer site

SYSGEN program obtains information concerning the specific configuration of the hardware system

Booting – starting a computer by loading the kernel

Bootstrap program – code stored in ROM that is able to locate the kernel, load it into memory, and start its execution

Ch3

1.名詞解釋

1. PCB-每個process在os中都有一個對應的PCB，內容有 : 行程狀態.程式計數器.cpu暫存器.cpu排班.記憶體管理資訊.會計資訊.IO狀態資訊
2. CPU utilization—cpu執行process的時間 / cpu總時間
3. CPU-bound process—行程中花去的計算的時間比io時間來的多
4. Degree-同時存在於記憶體中的程式數量

2.生生死死

Diagram of Process State (5%)

說明state的意義(15%)

**new**: 產生一個新的Process

**running**:指令正被執行

**waiting**: Process等待某些事件發生

**ready**: Process正等待分配給一個行程

**terminated**: Process已完成執行

State轉換間的原因(10%)

**new**→**ready** Process由long-term Scheduler選出一些Process載入到ready的Memory中等待執行。

**ready**→**running** Process由short-term Scheduler選出一個Process取得CPU使用權。

**running**→**ready** 執行中的Process被高優先權Process插隊、Time-out或中斷發生。

**running**→**terminate** Process執行完畢。

**running**→**waiting** 當Process發生事件，如I/O Complete，當Process在等待I/O時，會被迫放棄CPU，移到waiting等待I/O。

**waiting**→**ready** 等到I/O回應後，會回到ready重新等待取得CPU。

3. 3個term

**long-term Scheduler** (job scheduler)

從job Queue中挑選出合適的jobs，將它們載入到Memory中預備執行。

特點：執行頻率最低、可調控Multiprogramming Degree及CPU-Bound與I/O Bound Jobs之混合比例、用於Batch system，但real time及time-sharing system不用。

**short-term Scheduler** (Process scheduler、CPU scheduler)

從ready Queue中挑選合適的Process，讓它或取CPU執行。

特點：執行頻率最高、所有系統皆須要。

**medium-term Scheduler**

當Memory space不足且又有其它高優先權Process需要Memory space時，則此Scheduler 會去挑選某些Process，將它們SWAP OUT到Disk中保存其process image，以空出Memory space，等到未來Memory足夠，再將它SWAP IN回Memory中。

特點：執行頻率居中、可調控Multiprogramming Degree及CPU-Bound與I/O Bound Jobs之混合比例、用於time-sharing system。

4.fork(),exec()

Address space

Child duplicate of parent

Child has a program loaded into it

UNIX examples

fork system call creates new process

exec system call used after a fork to replace the process’ memory space with a new program



Ch4

1. 解釋並比較user threads和kernel threads

|  |  |
| --- | --- |
| user threads | kernel threads |
| Thread management完全由在user mode執行的thread Library負責 | Thread management完全由kernel負責 |
| Fast context switching、thread 管理成本低 | slower context switching、thread 管理成本高 |
| 若Process內某條user thread發出Blocking system call 且kernel是single-Threaded，則會導致整個Process也被Blocked(即使內還有available Thread存在)  java ,win32, posix | 若Process內的某條Thread發出Blocking system call，不會導致整個Process也Blocked(只要Process內還有available Threads存在)  win XP, Linux |

2. Scheduler Activations

排程機制，多了一個up call(kernel對上層thread支溝通管道)，可以解決user thread的問題，因為當thread要呼叫system call時，kernel可藉由up call與上層thread溝通，因此kernel會知道上層thread的存在，就不會有block的問題。