**Chapter 1 Introduction**

**是非 (True/False)**

1. ( **T** ) An operating system call be called a resource allocator (作業系統可被稱為資源分配器)
2. ( **T** ) An operating system can be called a control program. (作業系統可被稱為控制程式)
3. ( **F** ) A P2P system is always with a centralized service. (P2P 系統總是帶有集中式服務)
4. ( **T** ) A real-time system has well-defined, fixed time constraints. (即時系統具有明確定義的、固定的時間限制)

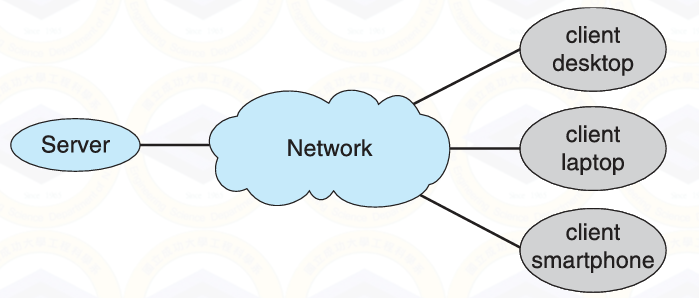
**寫出下列縮寫的英文全名**

* DMA: Direct Memory Access (直接記憶體存取)
* NUMA: Non-Uniform Memory Access (非均勻記憶體存取)
* LIFO: Last-In, First-Out (後進先出)
* FIFO: First-In, First-Out (先進先出)
* P2P: Peer-to-Peer (點對點)
* SaaS: Software as a Service (軟體即服務)

**解釋名詞**

* **Kernel (核心)**: 作業系統的核心組成部分，負責管理系統的資源。
* **Middleware (中介軟體)**: 作業系統與應用程式之間的中介層，讓它們能順利交換資料或服務。
* **Device driver (裝置驅動程式)**: 介於硬體與作業系統之間的程式，可以識別、控制並與硬體裝置互動。
* **Bootstrap program (啟動程式)**: 在啟動過程中載入作業系統到主記憶體（RAM）中執行的程式。
* **Firmware (韌體)**: 在硬體設備中的永久性軟體。讓硬體能夠正確運作，並與作業系統或應用程式溝通。
* **Blade server (刀鋒伺服器)**: 一種單板型態的伺服器，共享其他基礎設施，如電源供應、散熱、網路。
* **Daemon (常駐程式)**: 一種在後台執行的電腦程式，無需使用者直接互動即可執行各種任務。
* **Privileged instruction (特權指令)**: 只有**作業系統核心（Kernel Mode）**才能執行的指令，通常保留給作業系統執行 I/O 操作或記憶體管理等任務。
* **Hash function** : 一種數學演算法，將任意大小的資料轉換成固定大小的輸出值。
* **Bitmap (位元圖)**: 一種以**位元（bit）**為單位來表示資料狀態的**資料結構**，例如表示記憶體區塊是否被占用。
* **Public cloud (公有雲)**: 由**第三方雲端服務提供商**（如 AWS）所建立與管理的雲端環境，**多個使用者或企業可共同使用同一套雲端資源**，例如伺服器、儲存空間與網路。

**簡答**

1. We call that a processor needs to have at least two states to support the execution of operating system code and user-defined code. The two states are generally called?
   * **核心模式 (Kernel mode/Supervisor mode)** 和 **使用者模式 (User mode)**。
2. What is the interface that an operating provides for application programs to invoke operating system services?
   * **系統呼叫 (System calls)**。
3. Please draw a general structure of a client-server system.

**Chapter 2 Structures**

**寫出下列縮寫的英文全名**

* GUI: Graphical User Interface (圖形使用者介面)
* CLI: Command Line Interface (命令列介面)
* API: Application Programming Interface (應用程式介面)
* DLL: Dynamic-Link Library (動態連結函式庫)

**解釋名詞**

* **Shell (in Unix)**: 命令直譯器，為使用者提供 CLI 或 GUI 以與作業系統互動。
* **Linker (連結器)**: 將一個或多個目標檔案組合成單一可執行檔。
* **Loader (載入器)**: 負責將程式和函式庫放入記憶體中，讓 CPU 開始執行程式。
* **DLL (Dynamic Link Library，Windows)**: 可由多個程式同時使用的函式庫。程式在執行時，會「**動態載入（Dynamic Linking）**」這些函式庫，以便使用其中的功能，而不需要將所有程式碼都放進主程式中。
* **Policy (策略)**: 決定要做什麼（例如，哪個行程應該接下來執行）。
* **Mechanism (機制)**: 決定如何做（例如，如何從一個行程切換到另一個行程）。
* **Monolithic structure (of a kernel) (單核心架構)**: 將所有作業系統核心功能集中在同一核心空間中執行的設計，具有**高效能但低穩定性與維護性**的特點，是傳統作業系統的典型架構。
* **Microkernel (微核心)**: 是一種只保留最基本功能於核心、其餘作業系統服務在使用者模式運作的架構，具有**高安全性、穩定性與模組化設計**的特性，但效能稍低。

**簡答**

1. 一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

   AI 產生的內容可能不正確。Please draw a diagram to show the handling of a user application invoking the open() system call.
2. Three general methods are used to pass parameters to the operating system. Please write down one approach.
   * 三種常見的方法之一：
     + **直接用暫存器中傳遞參數。**
     + **暫存器存放參數所在的記憶體位址。**
     + **將資料逐一推入(push) 堆疊(Stack)，進入Kernel mode後再逐一取出 (pop)。**
3. 一張含有 文字, 螢幕擷取畫面, 圖表, 字型 的圖片

   AI 產生的內容可能不正確。The standard C library provides a portion of the system call interface. Please show an example to show how the standard C library performs as the interface between a C program and the operating system.
4. 一張含有 文字, 圖表, 行, 字型 的圖片

   AI 產生的內容可能不正確。Show the typical architecture of a microkernel.

**Chapter 3 Process**

**是非 (True/False)**

1. ( **T** ) A process is selected from ready queue for execution is also called that the process is dispatched. (從就緒佇列中選取一個行程執行也被稱為行程被派遣。)

**寫出下列縮寫的英文全名**

* PCB: Process Control Block (行程控制區塊)
* IPC: Inter-Process Communication (行程間通訊)
* TCP: Transmission Control Protocol (傳輸控制協定)
* UDP: User Datagram Protocol (使用者資料包協定)
* RPC: Remote Procedure Call (遠端程序呼叫)
* XDR: eXternal Data Representation (外部資料表示)

**解釋名詞**

* **Activation record (活動記錄)**: 是當程式呼叫一個函式或副程式（Function / Procedure）時，用來儲存該次呼叫所需資訊的記憶體區塊。
* **Degree of multiprogramming**: 指**系統主記憶體中同時存在的Process數量**。
* **Swapping** : 一種記憶體管理技術，暫時將行程從主記憶體移到硬碟，等需要再執行時再將其移回記憶體繼續執行。
* **Context switch (上下文切換)**: 是指作業系統為了將 CPU 從一個正在執行的**程序 (Process)**切換到另一個程序或執行緒而進行的過程，**儲存 (Save)** 當前執行單元的狀態，**載入 (Load)** 下一個執行單元的已儲存狀態。
* **Cascading termination**: 指在一個程序（通常是**父程序**）終止或被強制終止時，作業系統會自動地、遞迴地終止所有與之相關聯的**子程序**以及子程序所衍生的**所有後代程序**（即孫程序、曾孫程序等）。
* **Blocking or nonblocking (synchronous and asynchronous) message passing Socket**:
  + **阻塞 (Blocking/同步 Synchronous)**: 傳送者或接收者會等待直到訊息傳送或接收完成。阻塞呼叫將等待直到事件發生才會返回。
  + **非阻塞 (Nonblocking/非同步 Asynchronous)**: 傳送者或接收者不會等待，而是立即返回，因為另一個執行緒執行工作並在工作完成時發出標誌或事件。
* **Remote procedure call (遠端程序呼叫)**: 一種讓程式可以**呼叫遠端電腦上的函式**的技術。
* **Stub (in RPC)**: 負責在客戶端與伺服器之間傳遞呼叫與回傳資料的中間層。
* **Port (in RPC)**: 是**伺服器在網路上提供服務時的識別編號**，就像郵局的信箱號碼，讓 client 能找到正確的服務。
* **XDR (external data representation)**: 在不同電腦系統之間傳輸資料時，確保資料格式一致的標準化資料表示方式。
* 一張含有 文字, 螢幕擷取畫面, 字型 的圖片

  AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

  AI 產生的內容可能不正確。**Matchmaker (in RPC)**: 用來幫助 **Client**找到 **Server**所提供的服務位置的**機制**（通常是 **Port 號碼**）。

一張含有 文字, 螢幕擷取畫面, 字型, 圖表 的圖片

AI 產生的內容可能不正確。

**簡答**

1. 一張含有 文字, 螢幕擷取畫面, 數字, 行 的圖片

   AI 產生的內容可能不正確。Draw the layout of a process in memory.
2. 一張含有 文字, 圖表, 螢幕擷取畫面, 字型 的圖片

   AI 產生的內容可能不正確。Draw the memory layout of a C program
3. 一張含有 文字, 螢幕擷取畫面, 圖表, 字型 的圖片

   AI 產生的內容可能不正確。Suppose that you are to design an operating system which would have five states: NEW, RUNNING, WAITING, READY, TERMINATED. Please draw a diagram to show process state transition with a appropriate events.
4. 一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

   AI 產生的內容可能不正確。Show three pieces of information associated with a specific process.
   * 儲存在行程控制區塊 (PCB) 中的三個資訊片斷：
     1. **行程狀態 (Process state)** (例如：New, Ready, Running)。
     2. **程式計數器 (Program counter)** (下一個要執行的指令位址)。
     3. **CPU 暫存器 (CPU registers)** (所有行程可見暫存器的內容)。 *(其他有效答案包括 CPU 排程資訊、記憶體管理資訊、會計資訊和 I/O 狀態資訊。)*
5. Draw a diagram to show the process creation model of the UNIX fork() system call.

一張含有 行, 螢幕擷取畫面, 圖表, 文字 的圖片

AI 產生的內容可能不正確。

1. Refer to fig. 3.8 (p.118) the sample program to create a separate process using the UNIX fork() system call. Fill the blanks of the sample program.

**根據 fork() 的回傳值判斷：**

* + - If pid **<** 0: process creation failed. (若 pid 小於 0，行程建立失敗。)
    - If pid **==** 0: this code is executed by the child process. (若 pid 等於 0，由子行程執行此程式碼。)
    - If pid **>** 0: this code is executed by the parent process. (若 pid 大於 0，由父行程執行此程式碼。)

1. 一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

   AI 產生的內容可能不正確。Please draw diagrams to show the communication models: (a)shared memory, (b)message passing.

**Chapter 4 Threads and Concurrency**

**是非 (True/False)**

1. ( **F** ) A user thread is always assigned to a kernel thread. (一個使用者執行緒總是會被指派給一個核心執行序。)

**寫出下列縮寫的英文全名**

* LWP: Lightweight Process (輕量級行程)
* APCs: Asynchronous Procedure Calls。
* GCD: Grand Central Dispatch (大型中央調度)

**解釋名詞**

* **Data parallelism (資料平行)**: 將相同的操作同時應用在不同資料上的平行處理方式。
* 一張含有 文字, 螢幕擷取畫面, 字型, 黑色 的圖片

  AI 產生的內容可能不正確。**Task parallelism (任務平行)**: 不同的核心或執行序**同時執行不同的任務**。
* **Kernel thread (核心執行序)**: 由作業系統核心直接管理的執行序。
* **Thread library (執行序函式庫)**: 為開發者提供用於建立和管理執行序的 API 的函式庫。
* **Pthread**: 一種 UNIX / Linux 系統使用的標準執行序函式庫。
* **Thread pool**: 一種預先建立固定數量執行序，它們等待被指派工作。這有效減少系統中的執行序數量，並避免了不斷建立和銷毀執行序的開銷。
* **Deferred cancellation (of a thread) (執行序延遲取消)**: 當一個執行序被要求終止時，不會立刻被強制中止，而是等到執行序執行到「可安全取消的檢查點（Cancellation Point）」時才結束。
* 一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

  AI 產生的內容可能不正確。**Asynchronous cancellation (of a thread) (執行序非同步取消)**: 一種執行序取消技術，一個執行序會立即終止目標執行序。
* **Thread local storage (執行序區域儲存)**: 一種機制，允許多執行序行程中的每個執行序擁有自己的全域或靜態變數私有副本。
* **Upcall** : 當核心發生某個事件時，它會**主動呼叫回**使用者程式中的處理函式，告訴使用者層發生了什麼事。

**System Call（系統呼叫）**：使用者程式 → 呼叫核心功能。

**Upcall（上行呼叫）**：核心 → 通知使用者程式。

* **Signal (信號)**: 是作業系統中用來**通知行程（Process）發生某種事件**的機制。

**簡答**

1. Describe the distinction between concurrency and parallelism.
   * **並行 (Concurrency)**: 是「看起來同時在做多件事」，實際上是**交錯進行**。
   * 一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

     AI 產生的內容可能不正確。**平行 (Parallelism)**: 是「真的同時在做多件事」，需要**多核心或多處理器**支援。
2. What is the Amadhl's Law? Suppose we have an application that is 75 percent parallel and 25 percent serial. If we run the application on a system with four processing cores, we can get a speedup of ?
   * **Amdahl's Law (阿姆達爾定律)**: 當系統的一部分被改進（例如平行化）時，**整體效能能提升的程度是有限的**。描述了程式的序列 (Serial) 部分如何限制增加處理器數量所能帶來的整體加速比。即使你可以將大部分工作平行化，但剩餘的序列部分仍必須單獨執行，這部分就成為了效能的瓶頸。
   * 一張含有 字型, 文字, 圖形, 符號 的圖片

     AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 字型, 圖表 的圖片

     AI 產生的內容可能不正確。**加速比計算**: