Chapter1 Instruction

	Bare Machine	Extend Machine
Def.	純粹只有 hardware component 構成的機器，其上無任何輔助 user 的 system program 的存在	在 Bare Machine 上加上 system program 所組成

- OS 之作用/功用
  - 作為 user 與 computer hardware 之間的溝通介面，協助 user 易於操作 computer
  - 提供一個讓 user programs 易於操作的環境
  - 作為 resource 之協調分配的管理者
  - 監督 processes 的執行過程，防止非法運作所引起的重大危害

- Cpu Idle time 過長的解決策略

	早期	Now
原因	人為 set-up(介入)時間過長	I/O Device 運作太慢，導致 CPU Idle 時間過長
解決方式	**Resident Monitor(常駐管理程式)** Ex. Control card, command interpreter, Automatic job sequencing, device driver, interpreter 機制	• **將較快的設備/機制 介入 CPU 與慢速 I/O Device 之間** • 讓 **CPU always busy**

- CPU-Bound Job & I/O Bound Job

	I/O Bound Job	CPU-Bound Job
Def.	此 JOB 的 I/O operation 量較大，但對於 CPU computation 的需求較少，導致於整個 JOB 的效能取決於 I/O Device 的速度	此 JOB 的 CPU computation 量較大，但對於 I/O operation 的需求較少，導致於整個 JOB 的效能取決於 CPU 的速度
Notice	在 Buffering 的機制下，若充斥著大量的 I/O Bound jobs, 造成 I/O Device 較慢，則 1. 在 input 的時候，cpu 被迫 wait，因為面對空的 input buffer 2. 在 output 時，cpu 被迫 wait，因為面對滿的 output buffer	在 Buffering 的機制下，若充斥著大量的 CPU Bound jobs,，則 1. 在 input 的時候，I/O Device 被迫 wait，因為面對滿的 input buffer 2. 在 output 時，I/O Device 被迫 wait，因為面對空的 output buffer
結論：CPU-Bound job 與 I/O Bound job 希望能比例均勻，使得 I/O Device 與 CPU 可以同步執行，雙方皆可提高使用率		

- SPOOLing (Simultaneous Peripheral Operation On-Line)

Def.	將 disk 做為一個極大的緩衝區塊，以 output 為例，cpu 將 output data 先送往 spooling area，cpu 便認為 I/O 已經結束，便繼續執行下一個 computation 的工作，而 I/O Device 也可在此時或之後從 spooling area 取出資料輸出。
優點	不同 job 間的 cpu computation 與 I/O operation 可以同步執行，故 cpu 與 I/O device 的使用率提高
缺點	1. OS 能須在 memory 中占用些許表格空間來記錄 I/O request 的處理狀況 2. 在 Disk spooling area 與 I/O device 的資源分配要得宜，否則會產生死結
Note	1. Buffering 只允許自身工作的 cpu computation 與 I/O operation 可以同步執行，但不同工作間則不行(前提假設為"user process 是使用自己的 user memory area 作為 buffering"，如果是使用 OS 的 memory 則不會出現此種問題)

	2. Spooling 不會出現問題的原因：spooling 是利用硬碟來作，歸 OS 管，在 memory 中有 page table 來記錄 I/O request，但是 buffering 是利用 user memory，OS allocate 之後便不於理會

- Buffering, Caching, Spooling

Buffering	Caching	Spooling
I/O device 預先將輸入/輸出的資料送至 memory，促使 CPU 與 I/O device 同時 busy，提升 utilization. 以 Output 為例，CPU 將 Output data 送往 buffer 之後，即可繼續向下執行，而 I/O device 會於此時或稍後至 Buffer 中取出資料。	將遠端資料或 I/O device 之最近存取資料存放在 Caching area 之中，以便將來有需求可至 caching area 取得，避免花費時間再到 remote site 或 I/O device 存取	運用 Disk 作為極大的緩衝空間在使用，使得不同 job 之間的 CPU computation 與 I/O operation 可以 Overlay execution

- Multiprogramming System & Time-Sharing System

	Multiprogramming System	Time-Sharing System
Def.	系統中允許多個程序同時執行(i.e. 可以 concurrent or parallel exection)，目的是為了提高 cpu 使用率	是 Multiprogramming System 的一種
作法 or 特徵	利用 cpu 排班技術讓多個 cpu 在不同的程序間切換，可以確保 cpu always busy，避免 idle	1. CPU scheduling 採用 RR(RR 排班法則：OS 制訂一個 CPU Quantum，若 process 取得 cpu 之後，未能在 Quantum 之內完成，便會被迫放棄 cpu，等待下一輪使用) 2. 強調 user 一律公平，強調 response time 要短 3. 適用於 user interface 的環境 4. 替 user 創造出一個專屬電腦的假象，使用技術如下 1. 透過 RR CPU 排班，創造出具有多顆 CPU 的

		假象 2. 運用 Virtual Memory，擴大 user area space 的感覺 3. 利用 spooling 的技術，營造多套 I/O Device 的效果
NOTE	Multiprogramming Degree：系統中存在的執行 process 的個數，一般而言 Multiprogramming Degree 越高，則 cpu 的使用率也越高, but "thrashing"例外(Thrashing 這種現象指的是，欲使用的資料從虛擬記憶體搬至實體記憶體時，實體記憶體可能有一部份需要被置換，這些置換的資料再搬至虛擬記憶體中。然而被置換的部可能馬上又要使用，再從虛擬記憶體搬至實體記憶體，如此週而復始不斷進行搬運的動作，所有時間都花在這些無關緊要的置換搬運，導致 CPU 的使用率大幅降低)	[恐]別稱: Multitasking System

※什麼樣的情況下適合用 time-sharing system?

→1. the task is large and hardware is fast. the problem can be solved than other PC.

2. when lots of users need resources at the same time

- Multiprocessor System && Distributed System

	Multiprocessor System	Distributed System
別稱	Multiprocessing System, Tightly Coupled System, Parallel System	Loosely Coupled System
特徵	1. 一個 machine 具多顆 processor 2. Process 之間彼此共享 memory, bus. I/O devices, power supply, 機殼, etc. 3. 通常由同一個 clock 控制 4. 通常由同一個 OS 管控 5. Process 間的溝通大都採取 share	1. 由多部 machine 組成 2. Process 擁有自己的 memory, bus. I/O devices, power supply, 機殼, etc. 3. 通常不受同一個 clock 控制

	memory 的方式 6. 支援 Parallel Computing	4. 通常不受同一個 OS 管控 5. Process 間的溝通大都採取 Message Passing 的方式 6. Process 間彼此透過 Network or High-speed bus 相互連結
優點	1. Increased Throughput 2. Increased Reliability 3. Economy of scale(硬體擴充)	1. Resource sharing • cost 節省，因為某些 resource 在某部 machine 上有提供，故 machine 不須自己私有，而可共享之。 2. speed up -> throughput 提升 3. reliability 4. communication
缺點	They are most complex in both hardware and software than uniprocessor systems.	

- SMP && ASMP

	SMP (Symmetric Multiprocessor)	ASMP (Asymmetric Multiprocessor)
Def.	每個 processor 的功能(能力)皆相同，強調 ”Load Balancing “，避免各 processor 的負擔輕重不一	每個 processor 負擔之功能不盡相同，通常會有一個 Master Processor 負責工作、資源的分配，而其餘的 processor 稱為 Slave Processor，此種架構稱為” Master-slave architecture “
優點	1. reliability 較高 2. performance 較好	OS 的設計較簡單，因為容易從單一 processor、OS 做移植/改變
缺點	OS 的設計會較複雜，因為必須對共享資源提供相關的互斥存取控制	(1) reliability 較低 (2) performance 較差 (因為 Master Processor 是瓶頸 bottle neck)

- Hard real-time system && Soft real-time system

	Hard real-time system	Soft real-time system
Def.	系統必須保證 the critical task 可在規定的時間內完成(the system must ensure the critical task complete on time)，即對於工作完成的時間有極嚴格的要求，若未在規定的時間內完成或失敗，其嚴重性往往很高	系統必須保證 the real-time process 具有 the highest priority than the others，即可維持數值不衰減直到完工為止，但對於工作的完成時間，要求不如 HW real-time 嚴格
盡量降低 kernel dispatch latency	V （通常 HW real-time system 很少有 OS 的存在,若有也是極小化，如：microkernel）	V
CPU scheduling 技術	所有會造成時間 delay 的因素需全盤納入設計考量估計（eg: sensor data 傳輸, process 處理速度, etc.）"確保這些時間加總能小於等於 Time constract 以確定是否 schedulable(可排班化)"	• 要能支援 priority scheduling • 不提供"Aging"的技術
Virtual Memory	絕對不用 （因為 page fault 處理時間太長且不可預期） Disk 等輔助暫存器不用或少用	可 （但要求 All pages of the real-time process never SWAP out until it completes）
現行商用 OS	不支援此系統 ，因為一般桌電具有 secondary storage，且商用 OS 多半支援 virtual memory 會造成執行某一 process 不可避免或不可預期的時間變化所以會 miss deadline requirement.	支援

※What the main difficulty that a programmer must overcome in writing an OS?
→ when writing an OS, the writer must ensure that his scheduling schemes don't allow the response time exceed the time constraint. 如果超過，real-time 會中斷整個系統

- Batch System
  - Def. ：只有相同性質或具週期性，非急迫性的工作可以累積成堆，在整批送入系統處理
  - 目的： 希望提升 resource utilization, 通常不適用於 real-time 環境或需 user interaction 環境

- Handheld System

HW 上的限制	SW system 之設計要求
1. 通常是使用 slower processor(因為 power 供應有所限制) 2. Physical Memory size 有限或很小 3. Display monitor 不大	1. 程式運算不宜複雜，避免增加 processor 之運算負擔 2. 程式的 size 盡量精簡，不用的 memory space 要即時 release 3. 顯示的內容減少或客製化過

- Handheld device, real-time system

	handheld system	real-time system
Batch programming	x	x
virtual memory	x	v
Time sharing	v	x

- Consider a computing cluster consisting of two nodes running a data base. Describe two way in which the cluster software can manage access to the data on the disk

	asymmetric clustering	parallel clustering
Def.	one host running the data base with the other host monitor it. if the server fail, the monitor become the active server	兩個 host 同步的在執行 data base application
優	提供備份的 host	使用率高，速度較好
缺	host 的 utilization 不高	對於 shared area data 之互斥存取設計是困難的