1.What is an Operating System?

* Linux (UNIX) / Window98 / DOS / WindowNT

  Why so many kinds of OS?

* 菲傭 vs. 台傭? (Efficient vs. Convenient)

* Dependent on users' need, different OS's provide different services.

* CS vs. 人的生活
  OS (居家生活，服務業)
  NW (交通)

* A run-time environment (服務生，外部觀點)

  – For efficient and convenient usage of computer system.
    (Performance + User-friendly + Reliable)

  – 硬體一演進，軟體跟著演進

* a control program (管家)

  – Control execution of user programs.

  – Prevent errors / misuse
    (不可做超出本份的事)

* A resource allocator (分配者，協調者，內部觀點)
  - CPU time
  - Memory space
  - File storage (HD)
  - I / O device
  - Shared code / data

  (含軟硬體；jobs 之間之要求會有衝突)
  →考慮分享、規劃、管理使用、保護、偵錯、復原
  Fig. 1.1 (on the top of applications ➔ users)

System Goal (what it does)
1. efficient computation (efficiency)
2. convenience for users (user-friendly)
3. reliable (reliability)
4. intelligent (AI)

* History

– 軟體跟著硬體走
  軟體之演進：系統 → Application →language (OO)

(1) programmer = operator (老闆兼工友，一人公司)

– a significant amount of set-up time in the running of a job. (一堆開關，on / off)

– 程式目的很簡單(如當今之 99 元的計算器)

– programmed in binary

  EX. 2 + 3 ⇨ 0010 (2)
  1001 (+)
  0011 (3)

– 沒有程式軟體觀念(不存入 memory)，也沒有系統軟體觀念

(2) 程式放到 memory 後才開始執行

- 讀卡機，Tape (sequential access)
- PASCAL / COBOL / FORTRAN
- 由 Assembler → Compiler
- linker, loader, library, device driver
- loading FORTRAN compiler tape →
  running the compiler →
  Unloading the compiler tape →
  loading the assembler tape →
  running the assembler →
  Unloading the assembler tape →
  loading the object program →
  換一種 compiler

  EX.

X=2+1		Operator	Operand
Load 2	⇨	1001	0010
Add 1		1011	0001
Store X		1100	1100
			memory address

- No interactive response,很難 debug
- 改進之

  (a) 有經驗之 operator (operator ≠ programmer)

  (b) batch processing

(3) simple batch systems – resident monitor

含 (a) loader

(b) job sequencing (ex. KTV /餐廳領班)

(自動 transfer control from one job to the next)

(c) control card interpreter

(d) device drivers

(e) interrupt and trap vectors

– reduce set-up time

– improve CPU utilization

– 缺點：lack of interaction (難 debug)

– control card

$JOB
$USERID
$FORTRAN
(程式)
$RUN
$DATA
data
$END

(4) Overlapped CPU and I/O operations

(a) Off-line processing
(input 之東西沒有直接由 CPU 接收)

– Replace slow (speed) input device with faster units

– 以前 card reader → CPU → printer

– now, card reader → tape → CPU → tape → printer

– 不要浪費 CPU time (CPU 很貴，要物盡其用，時間就是金錢)

– Tape (logical I/O device)→device independence
因為以前(Ex. Fortran/Cobol)之 language 叫 I/O 時，要說清楚是用哪一種 I/O device.

– off-line processing 之缺點：使用者必須等 Tape 滿了，CPU 才 read Tape

(b) Spooling

– Disk 來臨(Random Access)

– Simultaneous Peripheral Operation On-Line

– Overlap the I/O of one job with the computation of the others.

– 優點

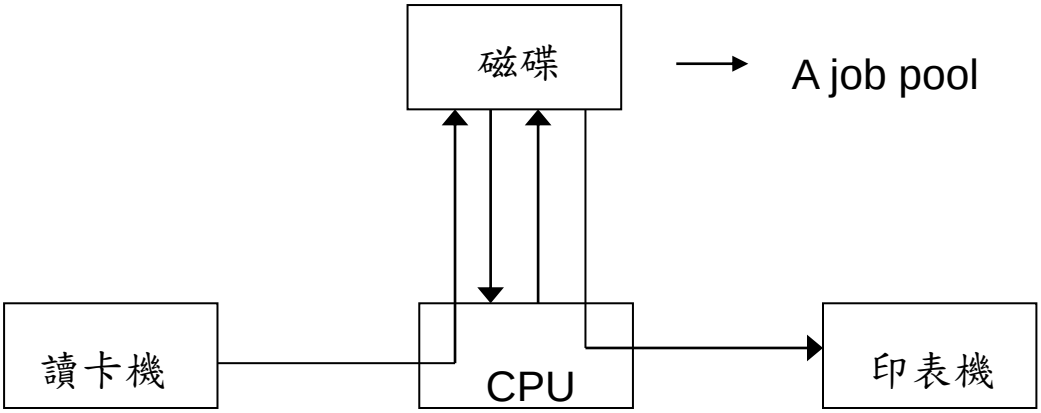
(1) 提供 multi-job 功能

(2) 具有 logical I/O device 特性，使程式具有 device independence 特性

(3) device independent：程式的執行與所使用之 I/O device 無關

(4) 使 CPU 與 I/O device 均 busy

(5) buffering 能使 I/O 操作與自己的 Computation overlap，但 spooling 能使某一 job 之 I/O 工作與很多 job 的 computation overlap



– on-line processing (用 disk)

– Virtual device
透過 spooling 技術把低速且專用(dedicated) device (例：card reader / printer)轉換成高速且可共用(shared) device (例：disk)，轉換產生之 device 稱為 virtual device.

(5) Multiprogrammed batched systems

– 遇 I/O operation，就換 job

(need job scheduling, memory management, CPU scheduling)

(6) Time sharing – interactive use (multitasking)

– Terminal (分享 CPU,用 Timer)

– Debug 容易

– Reduce turnaround time

(process)

(need virtual memory management, file system, disk management, concurrent execution, synchronization, communication, deadlock detection)

(7) Microcomputer (PC)

(need protection)

(8) Parallel systems

– 優點

(a) Speedup – throughput

(b) Lower cost (as compared to mainframe)

(c) More reliable – graceful degradation

– 分類

(a) Symmetric multiprocessing model – each processor runs an identical copy of the OS.

(b) Asymmetric multiprocessing model – master-slave

– Multiprocessing vs. Multiprocessor
(軟體 vs. 硬體)

(9) Distributed systems

– 因為 NW (and PC) cost ↓ speed ↑

– loosely coupled systems

– no shared memory / clock

– Heterogeneous vs. Homogeneous

– 優點

(a) Resource sharing

(b) Computation speedup (load sharing , load balance)

(c) Reliability (redundancy)

(d) Communication (ex.e-mail)

(10) Client-Server Model vs. Mainframe + Terminal
File Server (Data sharing)
Printer (Services sharing)
Mail

Server	Client
loop	
Any request	-Send a request
Accept one	-wait for a result if any
叫手下做	

(11) Real-time systems

(a) Hard real-time
the system has failed if a timing constant (deadline) is not met.

(b) Soft real-time
missing a timing constraint is serious , but does not necessarily result in a failure unless it is excessive.

– Real-time means on-time , instead of fast.

– Applications

(a) Air traffic control

(b) Multimedia system

(c) Space shuttle

(d) 股票市場

系統之分類

(1) 依同一時間內，可支援之使用者個數
單工(single job) vs. 多工(multi-task)
multi-programming (多程式系統) vs. multi-processor (多處理機系統)

(2) 依存取型態(何時對你的 data 做出反應)
batch processing (批次) vs. timesharing system (分時，interactive)

(3) 依組織型態(hardware organization)
centralized processing (集中式) vs. distributed processing (分散式)