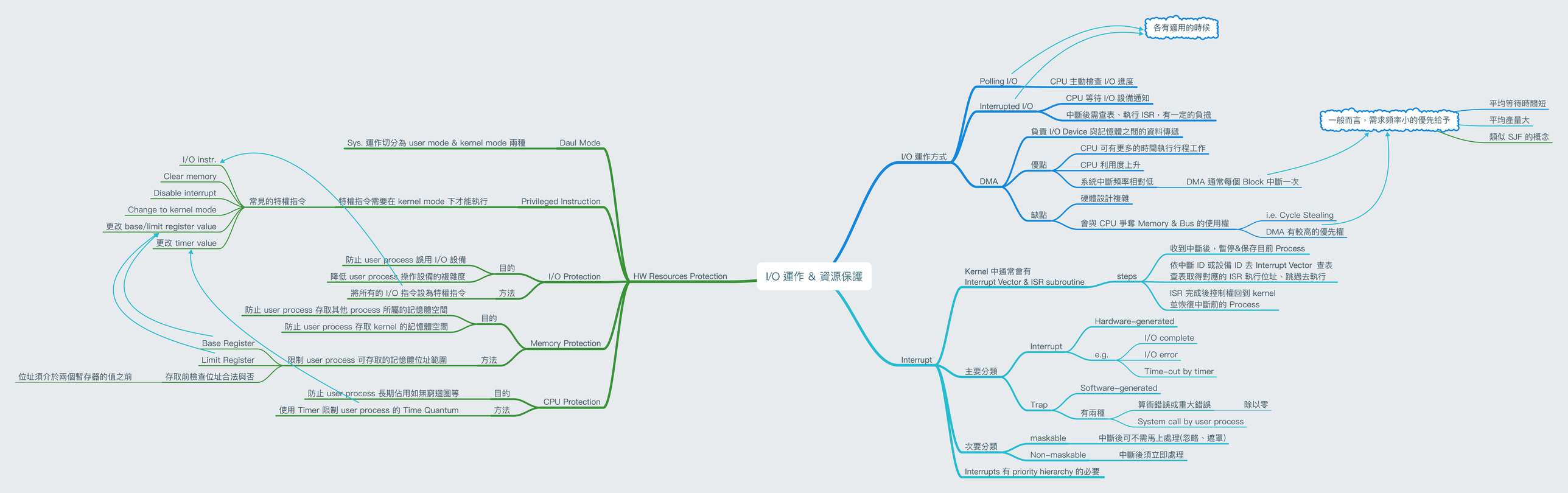
I/O 運作 & 資源保護
HW Resources Protection
Daul Mode
Sys. 運作切分為 user mode & kernel mode 兩種
Privileged Instruction
特權指令需要在 kernel mode 下才能執行
常見的特權指令
I/O instr.
Clear memory
Disable interrupt
Change to kernel mode
更改 base/limit register value
更改 timer value
I/O Protection
目的
防止 user process 誤用 I/O 設備
降低 user process 操作設備的複雜度
方法
將所有的 I/O 指令設為特權指令
Memory Protection
目的
防止 user process 存取其他 process 所屬的記憶體空間
防止 user process 存取 kernel 的記憶體空間
方法
限制 user process 可存取的記憶體位址範圍
Base Register
Limit Register
位址須介於兩個暫存器的值之前
存取前檢查位址合法與否
CPU Protection
目的
防止 user process 長期佔用如無窮迴圈等
方法
使用 Timer 限制 user process 的 Time Quantum
I/O 運作方式
各有適用的時候
Polling I/O
CPU 主動檢查 I/O 進度
Interrupted I/O
CPU 等待 I/O 設備通知
中斷後需查表、執行 ISR，有一定的負擔
DMA
負責 I/O Device 與記憶體之間的資料傳遞
優點
CPU 可有更多的時間執行行程工作
CPU 利用度上升
系統中斷頻率相對低
DMA 通常每個 Block 中斷一次
一般而言，需求頻率小的優先給予
平均等待時間短
平均產量大
類似 SJF 的概念
缺點
硬體設計複雜
會與 CPU 爭奪 Memory & Bus 的使用權
i.e. Cycle Stealing
DMA 有較高的優先權
Interrupt
Kernel 中通常會有
Interrupt Vector & ISR subroutine
steps
收到中斷後，暫停&保存目前 Process
依中斷 ID 或設備 ID 去 Interrupt Vector 查表
查表取得對應的 ISR 執行位址、跳過去執行
ISR 完成後控制權回到 kernel
並恢復中斷前的 Process
主要分類
Interrupt
Hardware-generated
e.g.
I/O complete
I/O error
Time-out by timer
Trap
Software-generated
有兩種
算術錯誤或重大錯誤
除以零
System call by user process
次要分類
maskable
中斷後可不需馬上處理(忽略、遮罩)
Non-maskable
中斷後須立即處理
Interrupts 有 priority hierarchy 的必要