問題	解答
1a.舉例 特權機器指令, 為何訂為特權指令?	(a) I/O 指令, 避免user process 執行I/O 指令, 破壞I/O設備 (b) 設定 PTBR 暫存器之指令, 避免 process 修改page table 地址, 破壞其它process
1b. OS 如何保證CPU不會被一個process 永久佔據.	OS 可設定 timer 作倒數計時, 時間到timer 就發出中斷, 中斷使得OS 取得CPU
2a. 六群系統呼叫, 包含process控制、檔案操作、通訊, 舉例	P. 61.
2b. 用於系統呼叫之參數傳遞的一般方法.	P. 59, Fig. 2.7, 用一暫存器指示 一記憶體區塊
2c. OS結構"modules"相對於"multi-layers"的優點	不需多層呼叫而較有效率, 動態載入模組而可節省記憶體.
3a. 發生中斷時, running process 會被搬至 那一狀態	ready 狀態.
3b. 進行context switch 時, 那幾個資料項會被存至 PCB	各個暫存器的值、program counter值、記憶體管理的資訊(如PTBR、PTLR)
3c. shared memory 比起 message passing 的優缺點	p. 117, 優點: 不透過OS傳資料, 所以速度較快 . 缺點: 共用變數未同步好時,易產生bug.
4a. fork( )被呼叫時, OS作什麼動作?	p. 113, OS把呼叫 fork( )的process 作複製, 設定其一為父process, 另一為子process
4b. execlp("/bin/ls", "ls", NULL)被呼叫時, OS作什麼動作?	OS從/bin/ls路徑讀取 ls之程式檔案, 並把 ls程式覆蓋 原呼叫execlp( )之process的記憶體.
4c. wait( )被呼叫時, OS作什麼動作?	OS把呼叫 wait( )的process 放入waiting 狀態, 直到另一process執行完畢, 才會把它放至 ready 狀態.
5a. 產生user threads之多執行緒程式, 無法獲得之好處?	p. 155, 無法獲得responsiveness, scalability
5b. one-to-one 多執行緒模式可獲得什麼好處?	其中一個執行緒被綁住(blocking), 不會牽連其它執行緒
5c. many-to-one 多執行緒模式可獲得什麼好處?	可產生出數量很多的執行緒, 不必受OS限制
6a. 非強佔式(non-preemptive)排程, 只會在那 2種情況作用?	p. 185, (1)當一行程主動呼叫system call, (2)當一行程執行完畢
6b. convoy effect 會在那一種CPU排程法裡發生? 為何發生?	p. 189, FCFS排程法, 當一個CPU bound 行程排於ready queue 的最前面時...
6c. 一行程被無限期阻擋的情形會在那一種CPU排程法裡發生? 如何解決?	p. 193, 優先權式排程法; 解決方法: aging, 每隔一段時間, 提升優先權等級.
7a. 計算 先佔式 Shortest-Job-First 排程法之平均等待時間.	p. 192
7b. 計算 Round-Robin 排程法之平均等待時間.	p. 194
8a.在那幾個地址綁定時間? 一行程所佔之記憶區塊不能被搬動.	compiler time 及 load time
8b. 緊連式(contiguous)記憶體分配法, 會碰到外部碎片之問題, 給二種解決方法.	方法1(p. 327): 定期地把process 記憶體挪動, 以把記憶體 "洞" 作合併. 方法2(p. 328): 把緊連式記憶體分配法 改換成 分頁式(paging)記憶體分配法.
8c. 分頁式記憶體分配法裡, 計算effective access time	p. 334,