OS HW3-1 B083040045 李亦晴

1. memory-mapped I/O 是記憶體對映輸出輸入，也就是 I/O 和 memory 共用記憶體空間，不用特別指令處理 I/O（因為和存取 memory 的指令相同→資料匯流排和資料記憶體共用，所以系統可用單一指令存取 Memory 和 I/O Device、）.

2. DMA 是 Direct Memory Access → 直接記憶體存取，是特殊的硬體結構，允許直接讀取、寫入記憶體、而不用經過 CPU 的處理

→好處：①使 CPU 使用效率提升 ②、用 DMA 操作記憶体速度 > CPU 操作速度.

↳因為使用 CPU 去做 I/O 時，在讀取&寫入過程被占用，無法執行其他工作

運作：大致上是 CPU 先啟動 DMA controler，此時 CPU 可進行其他操作，等 DMA 結束後接收中斷(interrupt)、

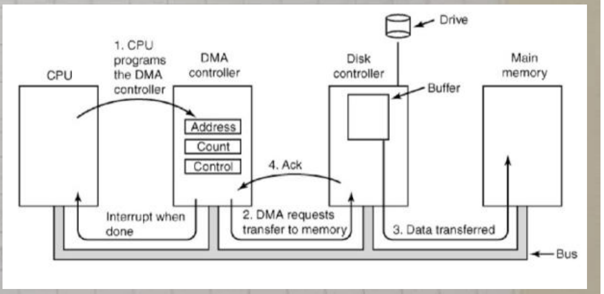
(1) CPU 把讀取到的資料傳給 DMA，要求 Disk controller 把資料放入 buffer，CPU 可執行其他操作

(2)、DMA controller 要求 Disk controller 把方才存的資料從 buffer 放進 memory.

(3)、Disk controller 實行 DMA 上述要求

(4)、Disk controller 完成，回傳 ack 信號

(5)、DMA 接收到後，傳中斷信號(interrupt) 給 CPU，使其知道可以繼續執行、



3. (a) Draw 4 Grantt charts. → FCFS, SJF, Non-preemptive priority, RR.

→ FCFS (First-come-first-served)

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
process	P1								P2	P3		P4	P5					

→ SJF (Shortest Job First)

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Process	P2	P4	P3		P5						P1							

→ Non-preemptive priority (可搶先的最優等級排程)

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Process	P2	P5						P3		P1								P4

→ RR(Round-Robin) q=1

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Process	P1	P2	P3	P4	P5	P1	P3	P5	P1	P5	P1	P5	P1	P5	P1	P5	P1	P5

(b)、求turnaround time of each scheduling algrithms.

→ 執行某行程花的時間(由入readyqueue ~ 執行完離开)、

	P1	P2	P3	P4	P5
FCFS	8	9	11	12	18.
SJF	18	1	4	2	10
Non-preeptive priority	17	1	9	18	1
RR	18	2	7	4	16.

(c)、求waiting time、

	P1	P2	P3	P4	P5
FCFS	0	8	9	11	12
SJF	10	0	2	1	4
Non-preeptive priority	9	0	7	17	1
RR	10	1	5	3	10

(d)、哪了有最少的waiting Time.

FCFS: 40 / SJF: 17 / Non-preeptive priority: 34 / RR: 29,

→ SJF waiting Time 最短，共17 單位時間

4、UNIX有user和kernal part, kernal part 像subroutine(副程式)、coroutine

* subroutine 是順序執行次序
* coroutine 是跳躍執行次序.

→ user part 用fork去執行其他指令，比較像call function → 跳躍

→ kernal part 被userpart 呼叫，只能從被呼叫的点依序執行，結束回傳到原來的点給user part → kernal part / subroutine ※