

1.

將 I/O register 和 memory 映射到一個定址空間(address space)，I/O 通常被規畫到 space 的頂端，接續 memory 定址的尾端，實現出 memory-mapped I/O。

CPU 透過發出欲存取的地址和 READ 訊號來存取 memory-mapped I/O，當地址落在自己(memory 或 I/O device)的範圍，則會響應。

2.

DMA 的產生是為了減少 CPU 存取資料所耗用的時間，當諸如硬碟的 I/O，若皆由 CPU 控制每個 byte 的 I/O，將會浪費許多 CPU time，因此透過 DMA 來控制其他 device I/O，硬碟可透過 DMA 直接存取記憶體。

CPU 設定 DMA 中的暫存器，接著 DMA 根據暫存器內容決定讀寫位置，將請求傳給硬碟控制器，硬碟控制器將資料從硬碟中寫入 buffer，當記憶體狀態許可則開啟傳輸，完成後回傳確認訊號給 DMA，DMA 將 countregister 數值-1，然後將地址往後遞增，若 count 不為 0 則重複上述步驟，再將新的請求傳給硬碟控制器。

3.(a)

FCFS

P1								P2	P3	P4	P5						
0							8	9	11	12							18

SJF

P2	P4	P3	P5						P1								
0	1	2	4					10									18

Non-preemptive priority

P2	P5						P3	P1							P4	
0	1					7	11								17	18

RR

P1	P2	P3	P4	P5	P1	P3	P5	P1	P5	P1	P5	P1	P5	P1	P5	P1
0	1	2		5				10						16		18

3.(b)

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

3.(c)

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

3.(d)

	FCFS	SJF	Non-pre	RR
P1	0	10	11	10
P2	8	0	0	1
P3	9	2	7	5
P4	11	1	17	3
P5	12	4	1	10
Total	40	17	36	29

So, SJF result in the minimal wating time.

4.

Kernel part 為執行 system call 的區域，一般來說，system call 可以並行執行，可視為一個 coroutine，若受到 blocking(諸如 I/O blocking)則可視為 subroutine。