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。