CH7、Disk

儲存裝置和其他周邊裝置

重點一：IO 設備

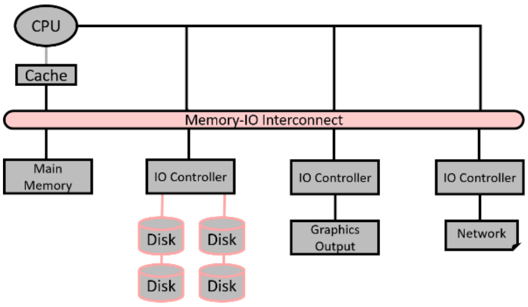
儲存設備的可靠性要比效能來得更要求。IO 種類非常多，我們常以下列三種特性來描述：

1. 行為 Behavior：輸入(鍵盤)、輸出(傳統螢幕)、或是儲存裝置(可讀可寫)
2. 夥伴 Partner：人、機器
3. 資料速率 Data Rate：IO 與 Memory 或 CPU 之最大資料傳送速率

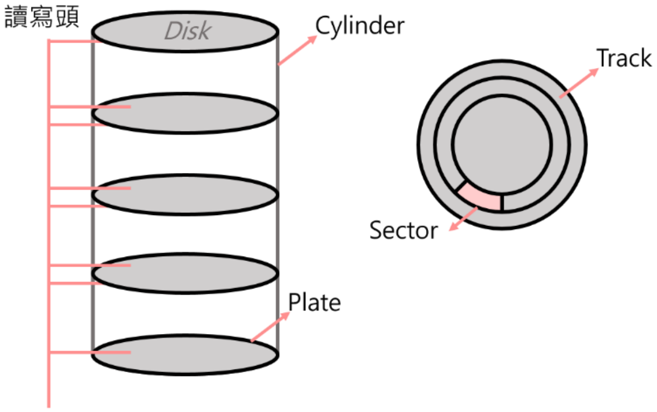
Device	Behavior	Partner	Data Rate(KB/s)
鍵盤	Input	人	0.01
滑鼠	Input	人	0.02
印表機	Output	人	1.00
Floppy Disk	Storage	機	50.00
LAN 網路	Input or Output	機	20-10000

評估 IO 效能的 2 種方式：

1. 單位時間內，移動多少資料？
2. 單位時間內，處理多少 IO operation？



重點二：硬碟



Disk Access Time
= Seek Time + Rotational Time + Data Transfer Time + Control Time + Waiting Time
Rotational Time => 0.5 * 1/Rate per second(RPS)
Data Transfer Time => Data amount / Data Transfer Rate => Data amount in Track * RPS

練習：對於轉速為 10000RPM 的典型磁碟，讀取或寫入一個 512 位元組磁區所花的平均時間為多少？廣告上的搜尋時間是 6ms，傳送速率是 50MB/sec，而控制器的額外時間為 0.2ns。假設磁碟是閒置的，因此沒有任何的等待時間。

Disk Access Time = 6.0ms + 0.5/10000RPM(10000/60RPS) + 0.5KB/50MB/s + 0.2ms
= 0.6 + 3.0 + 0.01 + 0.2 = 9.2ms

例(54)：Suppose we have a magnetic disk with the following parameters.
Controller overhead 1ms
Average seek time 12ms
# sectors per track 32 sectors/track
Sector size 512 bytes

1. If the disk's rotation rate is 360 RPS. What is the transfer rate for this disk?
2. What is the average time to read or write an entire track (16 consecutive KB). If the disk's rotation rate is 3600 RPM? Assumes sectors can be read or written in any order.
3. If we would like an average access time of 21.33ms to read or write 8 consecutive sectors (4K bytes), what disk rotation rate is need?

1. *Data Transfer Rate = 3600*32*512 = 57600 KB/sec*
2. *Access Time = 12ms + 1/60 + 1ms = 12+16.67+1 = 29.67ms*
3. *Suppose the rotation rate is R cycles per second*
   *Rotation time = 21.33ms = 12 + 0.5/R + 8/(R*32) + 1ms*
   ⇨ *R = 90.036 RPS*

例(22)：A hard disk has a track seek time of 10ms. The disk rotation speed is 9000 rpm. Each track on the disk has 600 sectors. Each sector has total 512 bytes data. What is the average time it takes in read 1024 bytes data?

Access Time = 10 + 0.5(1 / (9000/6)) + 1024 / (9000/60 * 600*512) = 13.355ms*

例(9)：A hard disk has a seek time of 4ms, 100 MB/sec transfer rate, and 0.2ms controller overhead. What is the average time to read or write a 512-byte sector for the disk rotating at 15000 RPM? Assume that the disk is idle so that there is no waiting time.

Disk Access Time = 4 + 0.5/(15000/60) + 0.5KB/100MB + 0.2 = 6.205ms

例(12)：Given a 10000 RPM disk with 80 MB/sec bandwidth and 10ms average seek time; please calculate the average time to read a 40KB block from the disk.

Access Time = 10 + 0.5/(10000/60) + 40KB/80MB = 13.5ms

[名詞]

磁柱 Cylinder：所給的時間點下，在所有碟面中位於磁頭下的所有磁軌

搜尋時間 Seek Time：移動磁頭到正確磁軌所花的時間

轉動延遲 Rotational Delay：到達所需資料的平均延遲*(一般以轉動半圈為準)*

傳輸時間 Transfer Time：傳送一個區塊的位元所需時間

控制時間 Controller Time：處理磁碟或磁碟與記憶體之間複雜的控制

等待時間 Waiting Time：程序在使用其他磁碟所產生的等待時間

例(20)：Which of the following statements are correct?

1. To improve the availability of an IO device, we have to increase its Mean Time to Failure(MTTF) and/or decrease its Mean Time to Repair(MTTR).
2. For a hard disk where seek time dominates the average read time, placing the data of a block on the same cylinder can improve the average read time.
3. DMA is a better way for IO devices to communicate with the processor than either polling or interrupt driven IO, because DMA works more efficiently with data caches.
4. RAID technology gives you disk subsystems with higher performance and greater dependability.

1、2、4

註(1)：Availability = MTTF / (MTTF+MTTR). So, increase MTTF and/or decrease MTTR will improve availability.

註(2)：DMA is for memory access not for cache access

例(15)：Which of the following statements are true?

1. MTTF (Mean Time to Failure) is a commonly used reliability metric for hard drive
2. Rotation latency is the required time for the desired sector of a hard disk to rotate under the read/write head
3. Using RAID 0 allow higher performance than a single disk
4. Instead of soring a complete copy of the original data for each disk, RAID 3 only adds enough redundant information to restore the lost information on a failure
5. None of the above

1、2、3、4

練習：A program repeatedly performs a three-step process: It reads in a 4KB block of data from disk, does some processing on that data, and then writes out the result as another 4KB block elsewhere on the disk. Each block is contiguous and randomly located on a single track on the disk. The disk drive rotates at 7200 RPM, has an average seek time of 8ms, and has a transfer rate of 20MB/sec. The controller overhead is 2ms. No other program is using the disk or processor, and there is no overlapping of disk operation with processing. The processing step takes 20 million clock cycles, and the clock rate is 400 MHz. What is the overall speed of the system in blocks processed per second assuming no other overhead?

Disk Read Time for a 4KB block = $8 + 0.5*60/7200 + 4*1024 / 20*2^{20} + 2 = 14.17ms$

Processing Time = $20*10^{6}*(1/(400*106) = 1/20 = 0.05s = 50ms)$

Disk Write Time for 4KB block = 14.17ms

*Time to completely process a 4KB block = 2*14.17 + 50 = 78.34ms*

Number of blocks processed per second = 1000/78.34 = 12.76

例(11)：A program repeatedly performs a three-step process: It reads in a 4KB block of data from disk, does some processing on that data, and then writes out the result as another 4KB block elsewhere on the disk. Each block is contiguous and randomly located on a single track on the disk. The disk drive rotates at 7200 RPM, has an average seek time of 8ms, and has a transfer rate of 20MB/sec. The controller overhead is 2ms. No other program is using the disk or processor, and there is no overlapping of disk operation with processing. The processing step takes 20 million clock cycles, and the clock rate is 400 MHz. What is the overall throughput of the system in blocks processed per second?

*(Seek Time + Rotational Delay + Data Transfer Time + Control Time) *2 + processing time*

⇨ $[8 + 0.5/(7200/60) + 4K/20M +2] *2 + (20*10^6)/(400*10^6) = (8+4.17+0.2+2)*2+50 = 78.37ms$

Block processed/second = 1/78.37ms = 12.76

快閃記憶體 Flash Storage

- 非揮發性半導體儲存體
- 比磁碟快 100-1000 倍
- 體積較小、較省電
- 使用有限次數*(1000 次；因為讀取只需要測量是否可過電，但寫入需改變 Ground 的電壓差)*
- 單價較高
- 無法取代 RAM 或磁碟

分成：

1. NOR Flash：可隨機存取、常用在嵌入式系統的指令記憶體、存取以 Word 為單位*(與 DRAM 較相似)*
2. NAND Flash：較便宜、常用在 USB、多媒體、存取以 Block 為單位*(與磁碟較相似)*

例(21)：True or False

1. An IO device raises a signal to inform the processor that it requires attention of the processor. This process is called polling and often used with a fast IO.
2. Writing data into a flash memory is faster than reading the data out of the flash memory.
3. NAND flash is much less expensive per gigabyte but memory could only be read and written in blocks.
4. Checking the status bit of an IO address to see if it is time for the next IO operation is called interrupt.
5. When an interrupt occurs, the processor must always repond to the interrupt and enter the interrupt service routine.

1	*2*	*3*	*4*	*5*
F	*F*	*T*	*F*	*F*

1. *Should be interrupt and is used for slow IO*
2. *Writing data into flash memory is much slower than reading data from it*
3. *True*
4. *Should be polling*
5. *When the higher priority interrupt is in service lower priority interrupt will be disable*

練習：Which of the following are true about flash memory?

1. Like DRAM, flash is a semiconductor memory.
2. Like disks, flash does not lose information if it loses power.
3. The read access time of NOR flash is similar to DRAM.
4. The read bandwidth of NAND flash is similar to disk.

All are true.

重點三：Dependability、Reliability、Availability

Dependability：一個系統能提供服務的品質

Reliability：一開始是好的，用了一陣子還是好的機率

Availability：瞬間是好的機率*(MTTF / MTTF+MTTR)*

Mean Time to Failure 平均錯誤時間

Mean Time to Repair 平均修復時間

Mean Time Between Failure：平均失敗間隔時間：MTTF + MTTR

例(3)：(前略)Answer the following questions:

1. Define MTBF, MTTR, and MTTF.
2. Considering a disk with an MTTF of 1 year and an MTTR of 1 day, compute its MTBF.
3. Following the previous question, what's the availability of this disk?
4. What's the implication as the MTTR approaches 0? Describe a technique in disk arrays that may result in very low MTTR.

1. *MTBF：系統連續兩次失效之間隔時間 = MTTF + MTTR*
   MTTR：系統失效後平均的修復時間
   MTTF：系統發生失效的平均時間
2. *MTBF = MTTF + MTTR = 365+1 = 366 days = 8760 hours*
3. *Availability = MTTF / (MTTR+MTTF) = 365/366 = 0.997*
4. *MTTR approaches 0 imply Availability => 1.0*
   Set them up in a RAID 5 configuration may result in very low MTTR. This way even when one drive fails, the parity information enables the disk array to continue operation and rebuild the failed drive online once it has been replaced.

練習：同上

如何增加 MTTF：

1. Fault tolerance： 系統錯誤依然可以繼續執行
2. Fault avoidance：投票制：多個執行，選多數的結果輸出
3. Fault forecasting：估計壞點並事先移除

練習：If a disk manufacturer quotes MTTF as 1200000hr (140 year) and you have 1000 disks. How many will fail per year?

*Annual Failure Rate (AFR) = 1000 * 8760/1200000 = 0.73%*

重點四：磁碟陣列

使用多個較小(較慢、但成本較低)的硬碟，來替代一個較大的硬碟。因為多個硬碟有較多讀寫頭，因此存取效能獲得改善；可靠度較低(壞掉機率較高)，因此需要多餘的硬碟(redundancy)來改善其 Reliability

N 個磁碟的 Reliability = 1 個磁碟的 Reliability/N

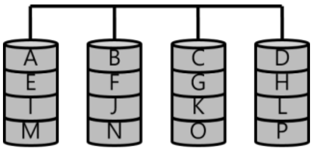
RAID(Redundant Arrays of Inexpensive Disks)

透過多個容量小、便宜的硬碟，使其能獲得比單一較大、較貴的硬碟(Single Large Expensive Drive, SLED)在存取上有更佳的效能；我們使用兩種技術，能使 Performance 及 Availability 都能提升：

1. Data Stripping：將資料分散到不同的硬碟，使得一次的資料存取會強迫同時到多個硬碟存取，以增加 Performance
2. Redundancy：使用多餘的硬碟來增加磁碟陣列的 Availability。此法只能改善一個系統的 Availability，但不能改善其 Reliability。Reliability 只能透過改善製造技術、或在系統中使用較少的零件來改善

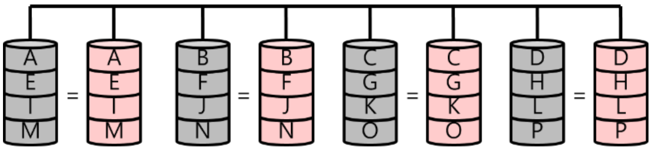
RAID 0(No redundancy)：

只將資料分散到不同硬碟中(Stripping)，強迫一次資料的存取會同時存取到多個硬碟。為非容錯硬碟，只要任一硬碟發生錯誤，則資料遺失*(有些名不符實，因為並沒有多餘硬碟的存在)*，具有最佳之 Small Read Latency



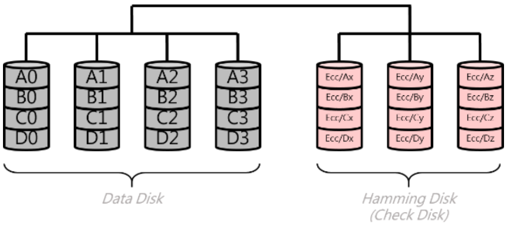
RAID 1(Mirroring)：

映射(Mirroring)或投影(Shadowing)為容忍硬碟錯誤最普遍的作法，直接 copy 一份到 Redundant Disk。是最昂貴的 RAID、需要最多硬碟，無法 Parallel Small Reads

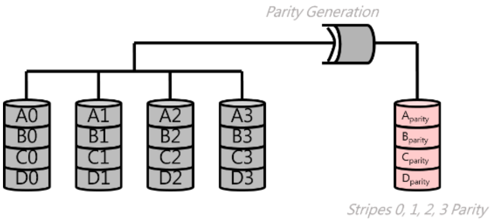


RAID 2(Error Detecting and Correcting Code)：

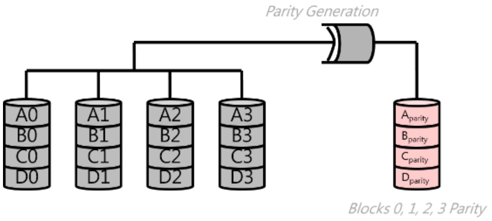
利用 Hamming code 方式來偵錯與校正；對較小的字組而言，具有非常高的 ECC (Error Correcting Code)，且每次寫入時，都要重新計算 Hamming code、並寫入 ECC 硬碟，故效率不佳



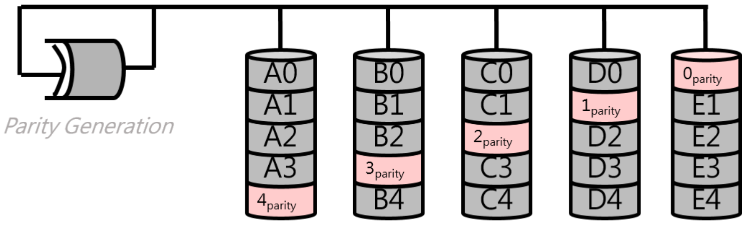
RAID 3(Bit-Interleaved Parity)：
加入多餘資訊(同位 Parity)，以在錯誤發生時能夠還原正確資料*(利用 XOR)*。『每次存取都會動用到所有硬碟(Read 效能較 RAID 4 差，且無法 Parallel Small Write 跟 Small Read)』，因此較適合使用大量寫入(Big Write)；反之，少量寫入(Small Write)就會有非常低之效能



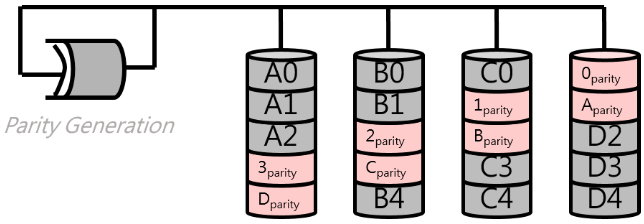
RAID 4(Block-Interleaved Parity)：
與 RAID 3 有類似之資料比例，但兩者存取資料方式不同，每次 Read 只需要用到少量 Disk，且 Parity 是以區塊的方式被儲存。故適用於少量寫入(Small Write)；但 Parity 硬碟成為效能的瓶頸



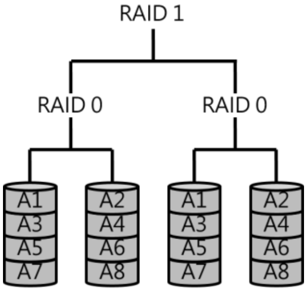
RAID 5(Distributed Block-Interleaved Parity)：
改善 RAID 4 每次寫入都要不斷更新 Parity 之缺點，將 Parity 分散至所有硬碟



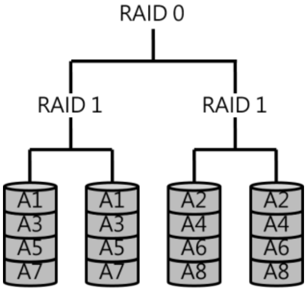
RAID 6(P+Q Redundancy)：
最大特色是可以修復第二個失效區塊，但也因此較 RAID 5 有 2 倍的檢查硬碟



RAID 1+0(Striped Mirror)



RAID 0+1(Mirrored Stripe)



練習：Which of the following are true about RAID levels 1, 3, 4, 5, and 6?

1. RAID systems rely on redundancy to achieve high availability.
2. RAID 1 (mirroring) has the highest check disk overhead.
3. For small writes, RAID 3 (bit-interleaved parity) has the worst throughput.
4. For large writes, RAID 3, 4, and 5 have the same through put.

1, 2, 3, 4

例(40)：What does RAID stand for? Regarding RAID levels 1, 3, 4 ,5 ,6, which one has the highest check disk overhead? Which one has worst throughput for small writes?

1. *RAID (Redundancy Arrays of Inexpensive Disks); An organization of disks that uses an array of small and inexpensive disks so as to increase both performance and availability.*
2. *RAID 1 has the highest check disk overhead.*
3. *For small writes, RAID 3 has the worst throughput.*

例(25)：RAID 1 has the highest check disk overhead among RAID levels 1, 3, 5. On the other hand, RAID 3 and 5 have the same throughput for large writes.
True / False.

True

例(23)：About RAID levels 1, 3, 4, 5, please answer each of following statements as True or False.

1. RAID systems rely on redundancy to achieve high availability and throughput.
2. RAID 1 (mirroring) has the highest redundancy overhead.
3. For small writes, RAID 3 has the worst throughput.
4. For large writes, RAID 3, 4, 6 have the same throughput.

1. *False (RAID systems rely on data stripe to achieve high throughput)*
2. *True*
3. *True*
4. *True*

例(49)：RAID (Redundant Arrays of Inexpensive Disks) have been widely used to speed up the disk access time. Several levels of RAID are supported. Please make the right binding between the following RAID levels and explanations.

1.	RAID 0	a.	block-interleaved parity
2.	RAID 1	b.	non-redundant striping
3.	RAID 4	c.	mirrored disks
4.	RAID 5	d.	block-interleaved distributed parity

RAID 0	*RAID 1*	*RAID 4*	*RAID 5*
b	*c*	*a*	*d*

例(42)：Please explain the designs and advantages for RAID 0, 1, 2, 3, 4, respectively.

Level	*Design description*	*Advantages*
RAID 0	*This technique has striping but no redundancy of data. It offers the best performance but no fault-tolerance.*	*1. Best performance Is achieved when data is striped across multiple disks* *2. No parity calculation overhead is involved and easy to implement*
RAID 1	*Each disk is fully duplicated.*	*Very high availability can be achieved*
RAID 2	*This type uses striping across disks with some disks string error checking and correcting (ECC) information.*	*Relatively simple controller design compared to RAID levels 3, 4, 5*
RAID 3	*This type uses striping and dedicates one drive to storing parity information.*	*Very high read and write data transfer rate since every read and writes go to all disks*
RAID 4	*RAID 4 differs from RAID 3 only in the size of the stripes sent to the various disks.*	*Better for small read (just one disk) and small writes (less read)*

例(37)：True or False：
RAID 3, 4, 5 all have the capability of performing parallel reads and writes.

False. For small access, it is true that RAID 4 and 5 have the capability of performing parallel reads and writes. But only one request can be served at a time for RAID 3, no matter the amount of access is small or big.

例(41)：In the RAID design, seven levels of the RAIDs are introduced in a commonly used textbook.

1. Which level of RAID uses the least storage redundancy? How much is the redundancy?
2. Which level used the most redundancy, and how much is it?
3. What is the most noticeable drawback of RAID 4 (block-interleaved parity)? And how does RAID 5 correct this drawback?

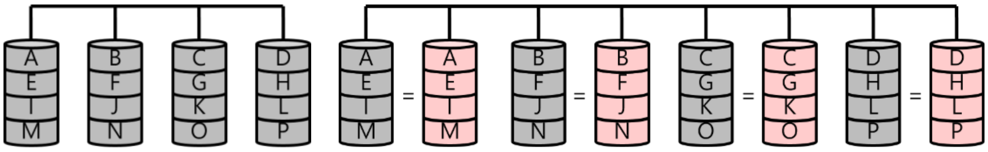
1. *RAID 0 ：0 redundancy*
2. *RAID 1 ：the number of data disks*
3. *Parity disk is the bottleneck ：Spread the parity information throughout all the disks to avoid single parity disk bottlenecks.*

例(29)：Consider three RAID disk systems that are meant store 10 terabytes of data (not counting for redundancy). System A uses raid 0, system B uses RAID 1, and system C uses RAID 5 with four disks in a “protection group”

1. How many terabytes of storage are needed for system A, B, C, respectively?
2. Determine which system is most reliable and which is least reliable.

1. *System A: 10 terabytes*
   System B: 20 terabytes
   System C: 10 + (10/4) = 12.5 terabytes
2. *System B (RAID 1) is most reliable*
   System A (RAID 0) Is least reliable

例(27)：Redundant Arrays of Inexpensive Disks, as named by the inventors and commonly referred to as RAID, is a technology that supports the integrated use of two or more hard-drives in various configurations for the purposes of achieving greater performance, reliability through redundancy, and larger disk volume sizes through aggregation.
RAID 0 is a striped set without parity. RAID 1 is a mirrored set without parity. The two different schemes are illustrated below. A1 stores the first block of file A , A2 stores the second block of file A, etc.



Let the size of the disk be 320 GB each.

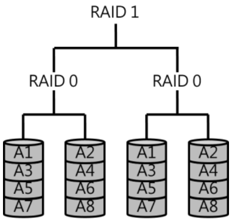
1. What is the total capacity of a RAID 0 set with 2 disks?
2. What is the total capacity of a RAID 1 set with 2 disks?
3. For write operations, which RAID set has the higher throughput?
4. For read operations, can the RAID 1 set deliver the same throughput as the RAID 0 set?
5. Which RAID set has the higher availability?

For the following questions, suppose that you have 4 disks and you are considering the following RAID configuration:

RAID 0: a 4-disk striped set
RAID 0+1: 2 striped sets in a mirrored set
RAID 1+0: 2 mirrored sets in a striped set

6. Draw a picture to illustrate how a file is stored ono the RAID 0+1 set
7. Which configuration has the highest write throughput?
8. When one of the disks has failed, which configuration has the highest write throughput?

1. *320 GB * 2 = 640 GB*
2. *320 GB*
3. *RAID 0*
4. *Yes (One Write or two Reads possible per mirrored pair)*
5. *RAID 1*
6. *Configuration of RAID 0+1:*



7. *RAID 0*
8. *RAID 1+0*

例(18)：Which of the following statements about RAID are true?

1. RAID 0 has no redundancy and no performance improvement.
2. RAID 4 has better system performance than RAID 4 as many read operations happen simultaneously.
3. When many writing operations are performed to a RAID 4 disk, the performance bottleneck will be the parity disk because all parity updates must be sequentially done.
4. The parity disk at RAID 5 can be used to recover the lost data if more than one data disk fails.

3

註(2)：只有 small read 時，4 才比 3 好

例(4)：For RAID 0, 1, 3, 4, 5, please answer question below.

1. Which one cannot have “small writes” to occur in parallel?
2. Which one cannot have “small reads” to occur in parallel?
3. Which one has the best “small reads” latency?
4. Which one is least tolerant to disk failure?

1	*2*	*3*	*4*
RAID 1, RAID 3	*RAID 3*	*RAID 3*	*RAID 0*

例(6)：About RAID, which of the following is (are) true?

1. RAID 0 can repair one lost disk from its mirror disk.
2. RAID 1 can repair all the lost disks from its mirror disk.
3. RAID 5 can repair one lost disk from a single parity disk.
4. RAID 4 can repair one lost disk from its parity disk.

[4] 註(3)：RAID 5 can repair one lost disk from a single parity disk and all other survived disks in a protection group

重點五~重點十：

(略)

重點十一：IO 裝置與處理器的溝通

Polling I/O

(一)Def：又叫 Busy-waiting I/O 或 Programmed I/O

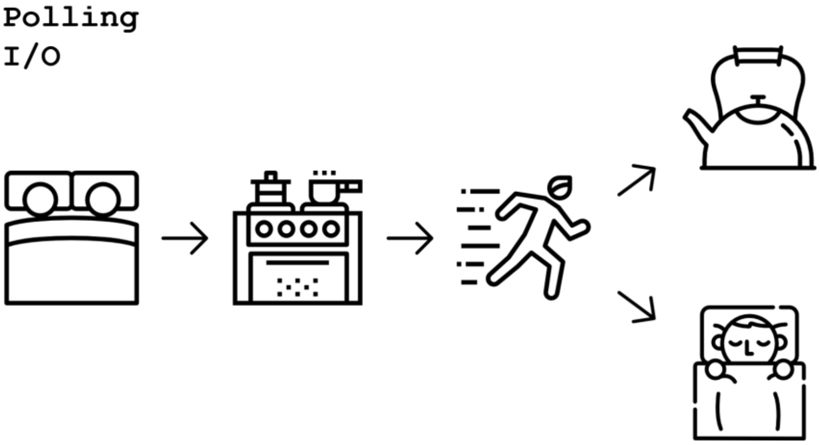
step 如下：

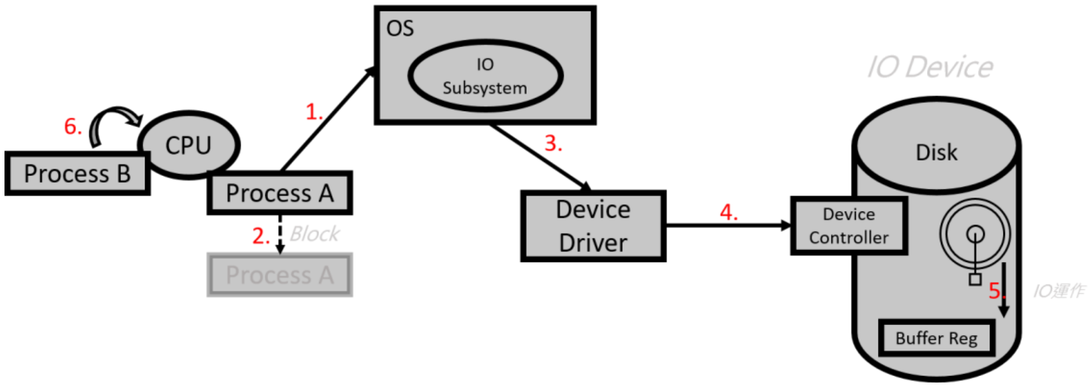
1. user process 發出 I/O request 給 OS
2. OS 收到請求後，(可能)會暫停此 process 執行並執行對應的 system call
3. kernel 的 I/O subsystem 會 pass 請求給 Device driver
4. Device Driver 依此請求設定對應的 I/O Commands 參數給 Device Controller
5. Device Controller 啟動監督 I/O Device 之 I/O 運作進行
6. 在此之時，OS(可能)將 CPU 切給另一個 process 執行
7. 然而，CPU 在執行 process 工作過程中都要不斷去 Polling Device Controller，以確定 I/O 運作是否完成或有 I/O error

比喻：老婆要求煮開水(最簡單的 Polling)

(前情提要：老婆要求簡單的煮開水，但家中煮水壺十分陽春，完全無功能)

1. *(早上起床，)**老婆**發出想喝咖啡的請求給我*
2. *我收到請求後，會暫停和**老婆一起躺在床上**的動作，並執行**煮開水的動作***
3. *我到廚房後，**手**打開**瓦斯爐開關***
4. ***瓦斯爐開關**會依照不同火力大小，打開**瓦斯爐***
5. ***瓦斯爐**啟動火焰，開始**燒水的動作***
6. *等水燒開的時候，我(可能)會去小孩的房間**叫小孩起床***
7. *然而，因為家裡的煮水壺不會自動提醒水已煮開，所以需要**不斷去廚房檢查水是否已燒開**，以確定不會燒完水的危險發生*





(二)缺點：CPU 耗費大量時間用於 polling I/O Device Controller 上，並未全用於 process execution 上，故 CPU utilization 低，throughput 不高

Interrupt I/O (中斷式)

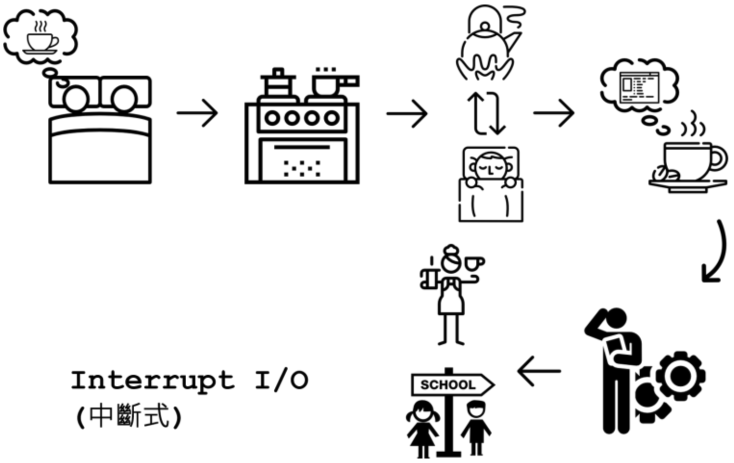
(一)Def：step 如下(1-6 同 polling)：

1. *user process 發出 I/O request 給 OS*
2. *OS 收到請求後，(可能)會暫停此 process 執行並執行對應的 system call*
3. *kernel 的 I/O subsystem 會 pass 請求給 Device driver*
4. *Device Driver 依此請求設定對應的 I/O Commands 參數給 Device Controller*
5. *Device Controller 啟動監督 I/O Device 之 I/O 運作進行*
6. *在此之時，OS(可能)將 CPU 切給另一個 process 執行*
7. 當 I/O 運作完成，Device Controller 會發出”I/O Completed” Interrupt 通知 OS (CPU)
8. OS 收到中斷後，(可能)會先暫停目前 Process 的執行
9. OS 必須查詢 Interrupt Vector，確認中斷發生，同時也要找到該中斷之服務處理程式(ISR：Interrupt Service Route 的位址)
10. Jump to ISR 位址，執行 ISR
11. ISR 完成後，return control to kernel，kernel 也許作一些通知工作
12. 恢復(resume)原先中斷之前的工作或交由 CPU Scheduler 決定

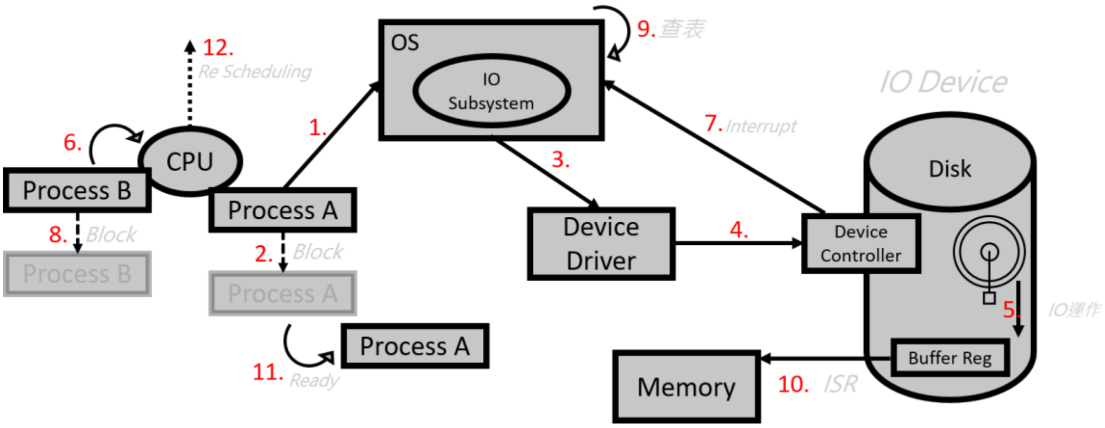
比喻：老婆要求煮咖啡(較進階的 Interrupt)

(前情提要：老婆要求較困難的泡咖啡，而家中煮水壺有進步，煮開時會發出通知聲)

1. *(早上起床，)**老婆**發出想喝咖啡的請求給我*
2. *我收到請求後，會暫停和**老婆一起躺在床上**的動作，並執行**煮開水的動作***
3. *我到廚房後，**手打開瓦斯爐開關***
4. ***瓦斯爐開關**會依照不同火力大小，打開**瓦斯爐***
5. ***瓦斯爐**啟動火焰，開始**燒水的動作***
6. *等水燒開的時候，我(可能)會去小孩的房間**叫小孩起床***
7. *當水煮開後，煮水壺會**自動發出通知聲**，來通知**我***
8. ***聽到**水煮開的**通知聲**後，我會先暫停叫小孩的動作，到廚房關水壺*
9. *到了廚房，**確認開水是否已經煮開**，而後查詢老婆想要咖啡之**配方***
10. *執行該咖啡**配方***
11. *完成咖啡後，**控制權回到我**(原本控制權是在配方上)*
12. ***回到**老婆身邊給咖啡、或執行更緊急的事(ex：小孩上學要遲到了)*



Interrupt I/O
(中斷式)



Interrupt	Address	Type	Function
00h	0000:0000h	Processor	Divide By Zero
01h	0000:0004h	Processor	Single Step
02h	0000:0008h	Processor	Non-maskable Interrupt (NMI)
03h	0000:000Ch	Processor	Breakpoint Instruction
04h	0000:0010h	Processor	Overflow Instruction
05h	0000:0014h	BIOS/Software	Print Screen
05h	0000:0014h	Hardware	Bounds Exception (80286, 80386)
06h	0000:0018h	Hardware	Invalid Op Code (80286, 80386)
07h	0000:001Ch	Hardware	Math Coprocessor Not Present
08h	0000:0020h	Hardware	Double Exception Error (80286, 80386) (AT Only)
08h	0000:0020h	Hardware	System Timer -IRQ 0
09h	0000:0024h	Hardware	Keyboard -IRQ 1
09h	0000:0024h	Hardware	Math Coprocessor Segment Overrun (80286, 80386) (AT Only)
0Ah	0000:0028h	Hardware	IRQ 2 -Cascade from Second programmable Interrupt Controller
0Ah		Hardware	Invalid Task Segment State (80286, 80286) (AT Only)
0Ah		Hardware	IRQ 2 -General Adapter Use (PC Only)
0Bh	0000:002Ch	Hardware	IRQ 3 -Serial Communications (COM 2)
0Bh		Hardware	Segment Not Present (80286, 80386)
0Ch	0000:0030h	Hardware	IRQ 4 -Serial Communications (COM 1)
0Ch		Hardware	Stack Segment Overflow (80286, 80386)
0Dh	0000:0034h	Hardware	Parallel Printer (LPT 2) (AT Only)
0Dh		Hardware	IRQ 5 -Fixed Disk (XT Only)
0Dh		Software	General Protection Fault (80286, 80386)
0Eh	0000:0038h	Software	IRQ 6-Diskette Drive Controller
0Eh		Software	Page Fault (80386 Only)
0Fh	0000:003Ch	Software	IRQ 7 -Parallel printer (LPT 1)

補充：中斷向量表

IRQ 0	System Timer
IRQ 1	Keyboard
IRQ 2	Cascaded with IRQ 9
IRQ 3	Default COM2 and COM4
IRQ 4	Default COM1 and COM3
IRQ 5	LPT2
IRQ 6	Floppy Drive Controller
IRQ 7	LPT1
…	…

IO 裝置對應代碼：最初因為電腦周邊並不多，所以只設計了 8 個，後來不夠用而又增加了一組，但是第一組的最後一個要作為與第二組溝通使用，所以實際上只有 15 個而已。
IRQ=Interrupt request

中斷處理的方式：

查表即先確認中斷號*(ex：00h，即了解發生了" Divide By Zero"的中斷)*，再至中斷表中找到對應該『中斷處理的程式碼』所存放的起始位址*(ex：處理" Divide By Zero"的中斷處理碼，起始位址存在 0000:0000h)*，而後跳至該起始位址*(0000:0000h)*，並執行中斷處理程式碼(ISR)

(二)優點：CPU 不須耗費時間用於 polling I/O Device，而是可以用於 Process execution 上，所以 CPU utilization 提高 throughput

(三)缺點：

1. Interrupt 之處理仍需耗費 CPU 時間
   如果 I/O 運作時間 ＜Interrupt 處理時間，則使用 Interrupt I/O 就不划算，所以 polling I/O 仍有其必要性
2. 若中斷發生頻率太高、會佔用幾乎所有的 CPU time，則系統效能會很差
3. CPU 仍需耗費一些時間用於監督 Device 與 Memory 之間傳輸的過程

[補充]若 ISR 正在執行時，又有其他中斷發生，則該如何處理？
法一：Disable 其他中斷(但不適用 Real-Time System、Time Sharing System)
法二：Interrupt 有優先權高低之分

比喻：
使命必達、勇往之前
老婆最大，其他人先別吵

DMA (Direct Memory Access)

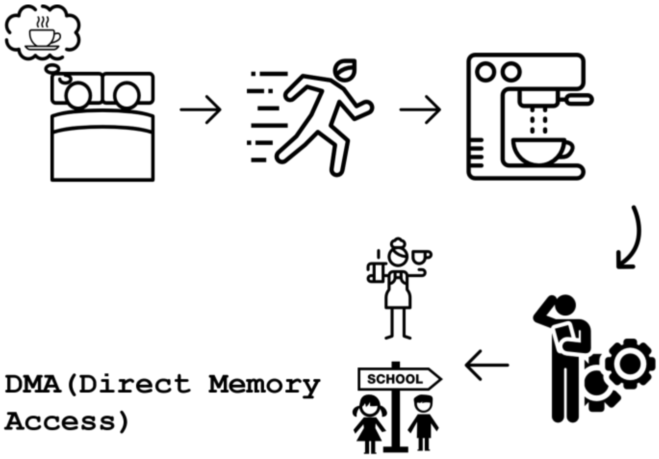
(一)Def：DMA Controller 負責 I/O Device 與 Memory 之間的 Data transfer 工作過程中無需 CPU 之參與監督，所以 CPU 有更多時間用於 process execution 上

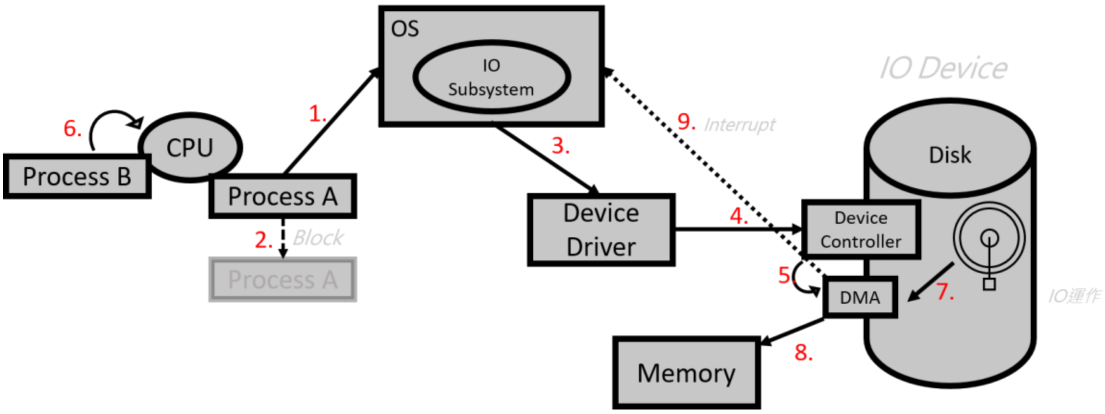
1. *user process 發出 I/O request 給 OS*
2. *OS 收到請求後，(可能)會暫停此 process 執行並執行對應的 system call*
3. *kernel 的 I/O subsystem 會 pass 請求給 Device driver*
4. *Device Driver 依此請求設定對應的 I/O Commands 參數給 Device Controller*
5. *Device Controller 啟動 DMA 以進行資料的傳輸*
6. *在此之時，OS(可能)將 CPU 切給另一個 process 執行*
7. *Device Controller 將資料傳至 DMA Controller*
8. *DMA Controller 將資料傳送至 Memory 目標位址*
9. *完成傳輸後，發出中斷以通知 kernel 資料傳送完成*

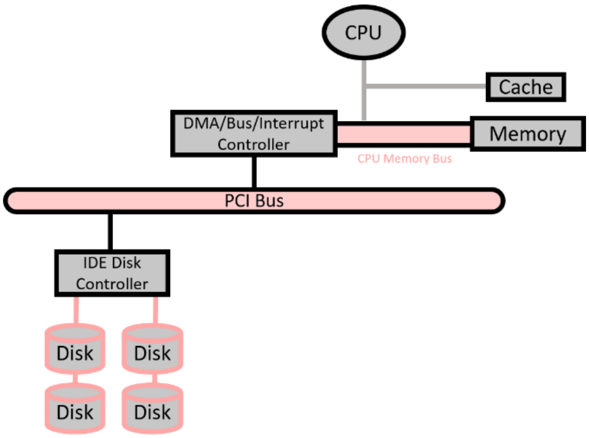
比喻：老婆要求煮咖啡，但已買咖啡機(可視為較進階的 Interrupt)

(前情提要：老婆要求較困難的泡咖啡，但家裡已經買了新的咖啡機)

1. *(早上起床，)**老婆**發出想喝咖啡的請求給我*
2. *我收到請求後，會暫停和**老婆一起躺在床上**的動作，並執行**泡咖啡的動作***
3. *我到廚房後，因為有了**咖啡機**，所以只要將老婆想要的**咖啡種釋資訊**，**用手按下**對應的**按鍵***
4. *按鍵會啟動咖啡機的**電腦***
5. *主機確認好後，**機器本體**會開始進行泡咖啡的動作*
6. *等機器泡咖啡的過程，我(可能)會去小孩的房間叫小孩起床*
7. *咖啡機本體運作中*
8. *咖啡機完成咖啡，**倒入杯子***
9. *咖啡機煮好**發出通知聲給我**，我拿咖啡給老婆、完成任務*







(二)優點：

1. CPU utilization 更高
2. 適合用在 Blocked transfer oriented I/O Device*(不適用在 Byte-transfer oriented I/O Device)*上(代表中斷發生頻率不致過高。ex：Disk)

(三)缺點：引用 DMA Controller 會增加 HW 設計複雜度(Complicated the HW design)

原因：DMA Controller 會與 CPU 爭搶 Memory、Bus 之資源使用權，若 DMA 佔用了 Memory、Bus 時，CPU 要被迫等待

[補充]

1. DMA Controller 採用”cycle stealing”的技術*(或 Interleaving[忍])* 與 CPU 輪番(交錯)使用 Memory 和 Bus。

   如果 CPU 因 DMA Controller 發生 conflict(同時都要 Memory/Bus)則會給 DMA Controller 較高的優先權

   人如果要喝水，可以自行選擇喝多喝少、且能夠觀查哪個時間裝水不會影響到咖啡機的運作，故水流的優先權會給予咖啡機較高的使用優先權
2. 通常系統會給予”對該資源需求量、頻率較小”的對象有較高的優先權，會獲得平均等待時間較小、產出較高。

   人可能隨時會突然口渴想喝水，但咖啡機需要有指令才會運作，且時間是相對可預測的，所以優先權較高
3. 機器指令 Stages：IF(Instruction Fetch)、DE(Decode)、FO(Fetch Operands)、EX(Execution)、WM(Write Memory)

	CPU 會 Memory Access	DMA 會用到 Memory
IF	會	衝突(Conflict)
DE	不會	沒問題
FO	可能	沒問題或衝突
EX	不會	沒問題
WM	可能	沒問題或衝突

CPU 設定 DMAC(Controller)的資料有：

1. IO Commands
2. Device data source/destination 位址
3. Memory 之 source/destination 位址
4. Timer：用以確認 transfer length*(依長度設定一 Timer 數值，開始傳輸即 Timer 倒數，倒數至 0 就表示傳輸完成)*

咖啡機比喻：

1. *選擇咖啡種類並按下咖啡機*
2. *咖啡機相關資料(廚房可能有很多台)*
3. *開水來源的管線、最終要倒入的地方(杯子)*
4. *依杯子大小設定時間(倒入時間較長則份量較多、時間較短則份量較少)*

[補充] IO processor *(智慧型的 DMAC)*

1. OS 會在自己的 kernel 空間中空出一 IO program 空間
2. CPU 通知 IO processor 到 IO program 去執行 IO
3. IO processor 一次可以處理多個 IO、且為自動執行

例(8)：Explain the following terms: DMA (Direct Memory Access)

DMA (Direct Memory Access): a mechanism that provides a device controller with the ability to transfer data directly to or from the memory without involving the processor.

例(55)：

1. Briefly describe the basic concept of the direct memory access (DMA). What advantages may the DMA have as compared with the polling and interrupt-driven data transfer techniques?
2. Briefly describe the three steps in a DMA transfer.

1. *DMA: a mechanism that provides a device controller with the ability to transfer data directly to or from the memory without involving the processor.*
   Other than polling and interrupt transfer techniques which both consume CPU cycles, during data transfer DMA is independent of the processor and without consuming all the processor cycles.
2. *Step 1: The processor sets up the DMA by supplying the identity of the device, the operation to perform on the device, the memory address that is the source or destination of the data to be transferred, and the number of bytes to transfer.*
   Step 2: The DMA starts the operation on the device and arbitrates for the bus.
   Step 3: Once the DMA transfer is complete, the controller interrupts the processor.

例(56)：What information should be contained in a DMA controller so that it can do the direct transfer between memory and IO device?

1. *Identity of the device*
2. *The operation to perform on the device*
3. *The memory address that is the source or destination of the data to be transferred*
4. *The number of bytes to transfer*

例(13)：Which of the following statements about DMA are true?

1. In terms of lowest impact on processor utilization from a single IO device, the order is DMA, interrupt driven and polling.
2. Before DMA transfer. DMA controller must notify processor by supplying identity of the device, operation, memory address, number of bytes to transfer, etc.
3. During DMA transfer, DMA controller is the bus master which directs each Read/Write between devices and memory.
4. On the completion of DMA transfer, interrupt mechanism is used to notify processor.

1、3、4

例(7)：About IO, which of the following is (are0 true?

1. Programmed IO is performed by the processor which executes IO programs.
2. Programmed IO is performed by the DMA controller which executes IO commands.
3. A DMA controller performs IO operation by the processor.
4. A DMA controller performs IO operations by itself.

4
註(1)：which executes CPU programs

練習：In ranking of the three way of doing IO, which statements are true?

1. If we want the lowest latency for an IO operation to a single IO device, the order is polling, DMA, and interrupt.
2. In terms of lowest impact on processor utilization from a single IO device, the order is DMA, interrupt driven, and polling.

1、2

例(53)：True or False

1. A daisy chain bus uses a bus grant line that chains through each device from lowest to highest priority.
2. There are three ways in interfacing processors and peripherals: polling, interrupt, and DMA. Among them, the polling IO consumes the most amount of processor time.

1. *False. The priority is from the highest to the lowest.*
2. *True*

例(52)：Compare the main differences among the following three IO data transfer technique: polling, interrupt, and DMA. Also describe their main advantages and disadvantages clearly and briefly.

Types	*Polling*	*Interrupt*	*DMA*
Differences	*The processor periodically checking the status of an IO device to determine the need to service the device*	*IO devices employs interrupts to indicate to the processor that they need attention*	*DMA approach provides a device controller the ability to transfer data directly to or from the memory without involving the processor*
Advantages	*Simple*	*Can eliminates the need for the processor to poll the device and allows the processor to focus on executing programs*	*DMA can be used to interface a hard disk without consuming all the processor cycles*
Disadvantages	*Waste a lot of processor time*	*More complex than polling*	*Require hardware support*

例(44)：Consider three types of methods for transferring data between an IO device and memory: polling, interrupt driven, and DMA. Rank the three techniques in terms of lowest impact on processor utilization.

DMA 、Interrupt driven 、Polling

例(10)：Briefly describe three techniques used for performing IO data transfer.

Polling	*The processor periodically checking the status of an IO device to determine the need to service the device*
Interrupt	*IO devices employs interrupts to indicate to the processor that they need attention*
DMA	*DMA approach provides a device controller the ability to transfer data directly to or from the memory without involving the processor*

例(34)：

1. Both networks and buses connect components together. Which of the following are true about them?
   (1) Networks and IO buses are almost always standardized.
   (2) Shared media networks and multi-master buses need an arbitration scheme.
   (3) Local area networks and processor-memory buses are almost always synchronous.
   (4) High-performance networks and buses use similar techniques compared to their lower-performance alternatives: they are wider, send many words per transaction, and have separate address and data lines.
2. In ranking of the three ways of doing IO, which statements are true?
   (1) If we want the lowest latency for an IO operation to a single IO device, the order is polling, DMA, and interrupt driven.
   (2) In term of lowest impact on processor utilization from a single IO device, the order is DMA, interrupt driven, and polling.

1. (1)、(2)

2. (1)、(2)

練習：

1. Prioritize interrupts from the devices listed in each table.
2. Outline how an interrupt from each of the devices listed in the table would be handled.

a	Power Down	Overheat	Ethernet Controller Data
b	Overheat	Reboot	Mouse Controller

1.

a	*Power Down: 2*	*Overheat: 1*	*Ethernet Controller Data: 3*
b	*Overheat: 1*	*Reboot: 2*	*Mouse Controller: 3*

2.

Power Down Interrupt	*Jump to an emergency power down sequence and begin execution*
Ethernet Controller Data Interrupt	*Save the current program state. Jump to the Ethernet controller code and handle data input. Restore the program state and continue execution*
Overheat Interrupt	*Jump to an emergency power down sequence and begin execution*
Mouse Controller Interrupt	*Save the current program state. Jump to the Ethernet controller code and handle data input. Restore the program state and continue execution*
Reboot Interrupt	*Jump to address 0 and reinitialize the system*

練習：

1. Why is DMA an improvement over CPU programmed IO?
2. When would DMA transfer be a poor choice?

1. *DMA is a mechanism that provides a device controller the ability to transfer data directly to or from the memory without involving the processor. This allows the CPU to perform arithmetic and other instructions while the DMA is going on in parallel.*

2. *DMA is not useful when the amount of data to transferred between memory and the IO device is very less. In this case, the overhead of setting up the DMA transfer would out weigh the benefits of direct data transfer without the interference of the processor.*

練習：假設 Polling 操作命令需 400 時脈週期，並且處理器以 500MHz 的速度來執行。決定以下三種不同的裝置，其消耗 CPU 時間的比例。假設有足夠的 Polling，所以沒有資料會遺失，並且裝置永遠都有可能是忙碌的：

1. 滑鼠每秒必須 Polling 30 次，以確保我們能捕捉到使用者所做出的任何移動。
2. 軟硬機以 16 位元為單位，傳送資料到處理器，其 data rate 為 50KB/sec
3. 硬碟以 4 個字組為單位在傳送資料，其 data rate 為 4MB/sec。而且此處不會遺失任何資料傳輸

1. *每秒 Polling 所花費的時脈週期 = 30*400 = 12000 cycles/sec*
每秒消耗處理器時間的比例= $(12*10^3) / (500*10^6) = 0.002\%$

2. *Polling 的次數為每秒：50KB/2byte = 25K polling*
*因此，Polling 每秒所需要的週期數= 25K*400*
消耗處理器時間的比例 = $(10*10^6) / (500*10^6) = 2\%$

3. *我們每秒 Polling 的次數為 4MB/16bytes = 250K*
*Polling 每秒所花費的週期 = 250K*400*
消耗處理器時間的比例 = $(100*10^6)/(500*10^6) = 20\%$

練習：假設我們有和上個範例中一樣的硬碟和處理器，不過我們使用中斷驅動 IO。每一次傳輸包含中斷的負荷為 500 時脈週期。如果硬碟只有 5%的時間在傳輸資料，試著找出處理器消耗時間的比例。

當磁碟一直是忙碌的，那中斷速率就和 Polling 速率相同。因此：磁碟每秒花費的週期 = 250K * 500 = 125 * 10^6 cycles/sec
處理器所消耗時間的比例 = (125*10^6) / (500*10^6) = 25%
假設磁碟只有 5%的時間在傳輸資料，則中斷所消耗處理器時間的比例 = 25% * 5% = 1.25%

練習：假設我們有和上一個範例中一樣的硬碟和處理器。假設 DMA 傳輸的啟始設定需要花費處理器 1000 時脈週期，並且處理 DMA 完成時所產生的中斷需要花費處理器 500 時脈週期。硬碟使用 DMA 時其傳送速率為 4 MB/sec。若磁碟傳送的資料單位平均為 8KB，且處理器速度為 500MHz，如果磁碟 100%的時間都在傳輸資料，試問處理器消耗的時間比例？忽略匯流排和 DMA 控制器之間競爭匯流排的狀況

每次 DMA 傳送花費 8KB/(4MB/sec) = 2*10^{-3} sec
如果磁碟不斷地在傳輸資料，則需 (1000+500) / (2*10^{-3}) = 750*10^3 cycle/sec。
因為處理器的速度為 500MHz，所以消耗處理器的時間比例 = (750*10^3) / (500*10^6) = 0.15%

例(46)：Assume that a hard disk in a computer transfers data in one-word chunks and can transfer at 2MB/sec. Assume that no transfer can be missed. Assume that the number of clock cycles for polling operation is 100 and that the processor executes with a 50-MHz clock. Determine the fraction of CPU time consumed by the hard disk assuming that you poll often enough so that no data is ever lost.

Polling per second = (2MB/sec)/4bytes = 500*10^3
Cycles per second for polling = 500*10^3*100 = 50*10^6
The fraction of CPU time = (50*10^6)/(50*10^6) = 100%

例(47)：Suppose there are a processor running at 1.5 GHz and a hard disk. The hard disk has a transfer rate of 8MB/sec and uses DMA. Assume that the initial setup of a DMA transfer takes 800 clock cycles for the processor, and assume the handling of the interrupt at DMA completion requires 400 clock cycles for the processor. If the average transfer from the disk is 16 KB, what fraction of this processor is consumed if the disk is actively transferring 100% of the time? Ignore any impact from bus contention between the processor and DMA controller.

Number of interrupts per second = 8MB/16KB = 500
Cycles per second for interrupts = 500*(800+400) = 6*10^5
The fraction of CPU time = (6*10^5) / (1.5*10^9) = 4*10^{-4} = 0.04%