2019，作業系統概論，期中考

筆試部分

(2 pt) 1.你的學號

(2 pt) 2.你的姓名

(3 pt) 3.一個安全性的作業系統（如：Linux）最起碼需要二個執行模式（dual mode operation）。請問： (5 Points)

甲、 Dual mode 中的 kernel mode 和 user mode 的差異（存取硬體？記憶體？）

乙、 在記憶體方面需要什麼樣的配合？（hint：記憶體好像檔案一樣對不同人要有不同的？？）

丙、 系統如何從 user mode 切換到 kernel mode（hint：特別的指令、單一進入點？）

甲：

Kernel mode 可以存取硬體 以及 所有記憶體

User mode 僅能存取自己這個 task 被分配到的記憶體

乙：

Kernel mode 要有自己的記憶體空間 一般稱為 Kernel space，僅有 kernel mode 能存取

User mode 也會有自己的記憶體空間 一般稱為 user space，user mode 以及 kernel mode 都能存取

丙：

以 x64 為例：

要從 user mode 切換到 kernel mode 需要通過 syscall 這個指令，並且會將 rip 設為 syscall 的單一進入點

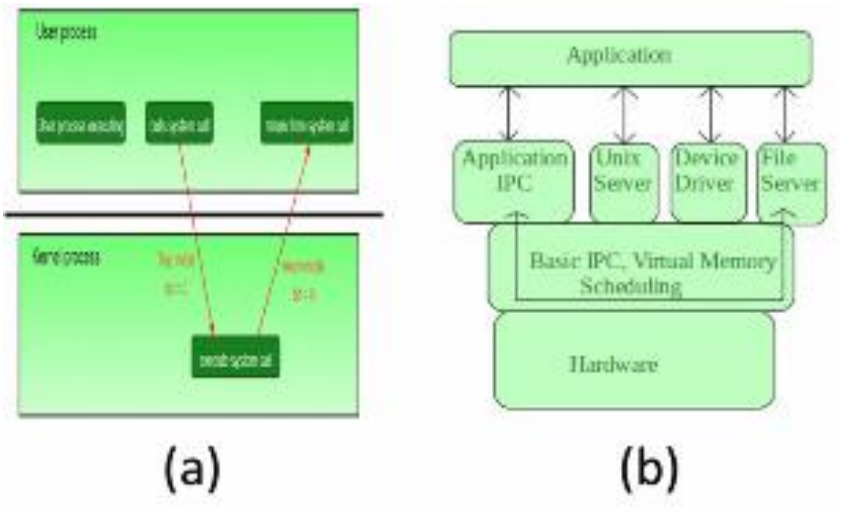
要從 kernel mode 切換到 user mode 需要通回 sysret 這個指令，並且會將 rip 設為 user space 裡下一行要執行的指令位置

(3 pt) 4.Signal 對 system call 的影響為何？（hint：在 interrutable 和 uniterrutable 時，system call 收到 signal 會怎樣？restart？）(3 Points)

1. interrutable 收到之後會暫停當前的 system call 並且處理中斷，處理完畢後，通常會重新執行 system call

2. uninterrutable 收到之後會忽略當前的 system call，待處理完 system call 後，再執行中斷

(3 pt) 5.請從 context-switch 和 mode change 的角度，說明 monolithic kernel 的優點 (3 Points)



因為 monolithic kernel 設計理念是將眾多底層模組都放到 kernel mode 中執行，相比微核心會將他們放到 user mode，模組間的溝通都涉及 context-switch 和 mode change monolithic kernel 中的模組需要互相溝通時，僅需要付出 function call 的代價即可，不需要進行 context-switch 和 mode change 也因此效能較佳

(3 pt) 76.電腦的主記憶體的三大主要用途？請問是否需要定期清除記憶體，說明你的觀點

1. cache
用來存放 disk 或儲存裝置的一些相關快取，以提升效能
2. buffer
用來將 I/O 的資料先給緩衝下來，等待量達到一定程度後再一起送出
3. program
儲存程式執行期間所用到的資料，如 text segment, data segment, heap, stack

我認為不用，因為核心對記憶體的設計原則本來就是盡量使用，不要浪費，若是記憶體有不足，則核心會自己分配，
若我們清除了，反而會造成效率降低，因為像是快取可能就被清掉了，還要再重新取得

(3 pt) 7.請設計一個系統，可以監控印表機的使用量（hint：印表機在 Linux 中為一個特殊檔案「/dev/usb/lp0」，假設可以使用「lpr」將文字檔案送到印表機，請問你要怎樣設定這些檔案的權限）（hint：/dev/usb/lp0 的權限要設定為何，lpr 又為何？）

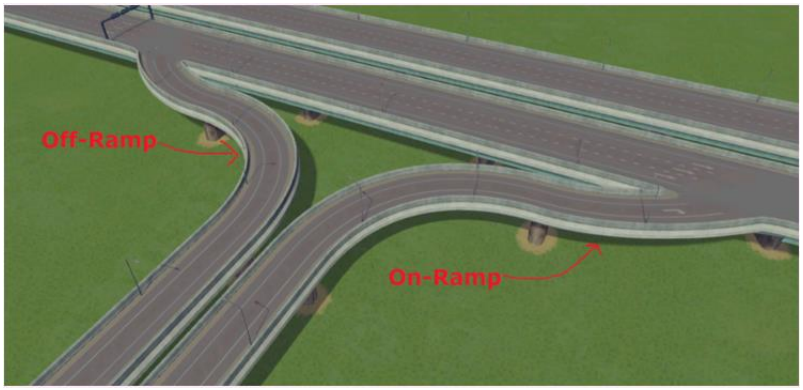
首先先限制 /dev/usb/lp0 僅有 root 可以存取， 再來將 lpr 也設為只有 root 可以執行，
這樣我們就可以讓一般使用者無法使用印表機，
之後可以考慮撰寫一個有 setuid 且 owner 為 root 且 任何人都能執行的系統(可以是自己撰寫的)，
一般使用者僅能利用這個方式使用印表機， 我們就可以利用這個系統進行監控，撰寫任何我們想要的功能。

(3 pt) 8.請問何謂 port I/O 和 memory mapped I/O？請問 memory mapped I/O 的優勢。

port I/O 為使用 port 進行定址空間進行 I/O，這邊使用的通道與一般的 memory mapped 的定址空間不同，需要配合特定組語
如： in, out
memory mapped I/O 為使用一般的 memory 空間進行定址，可以使用像是: mov 指令進行 I/O

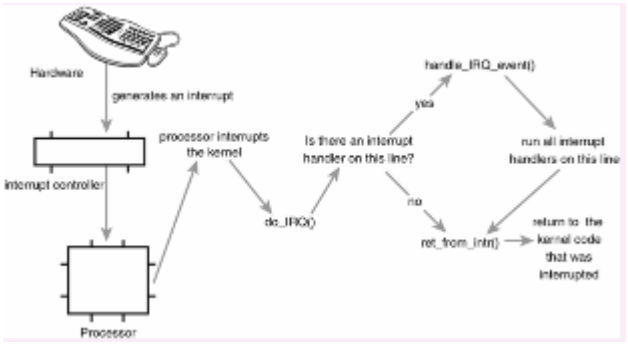
memory mapped I/O 相較 port I/O 較容易使用，擴展性也較高，定址空間也較大。

(3 pt) 9 目前的硬碟的控制器多半具有 DMA，請問 DMA 的功能為何？請問硬碟中為什麼要有 DRAM 充當 buffer？（注意：是「buffer」不是「cache」）



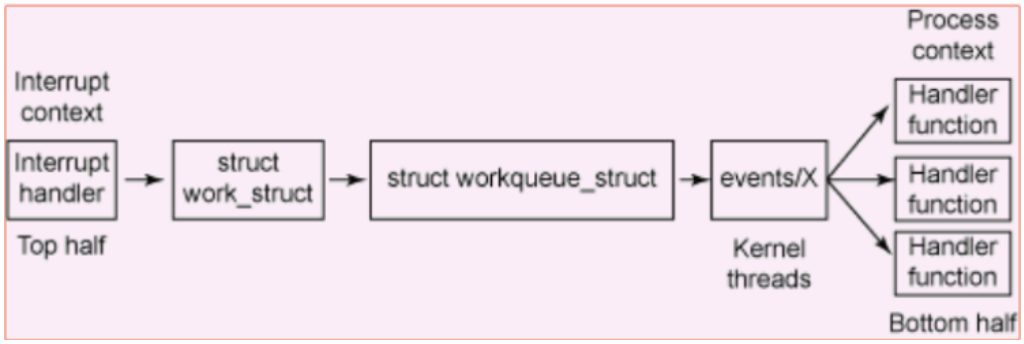
1.DMA 主要負責從裝置幫忙搬移資料 2.因為通常硬碟本身的傳輸速度不快，若是直接傳送到 bus 上，因為 bus 的頻寬較大，而硬碟所佔用的頻寬較少，很容易造成資源上的浪費，因此可以先將資料放到 buffer 內，等待資料量足夠了再一起傳輸

(3 pt) 10.請說明中斷的流程（hint：硬體、中斷訊息、CPU 怎樣暫停目前的執行、從哪邊找到這個中斷的「軟體處理流程」，中斷處理完以後要幹什麼事）



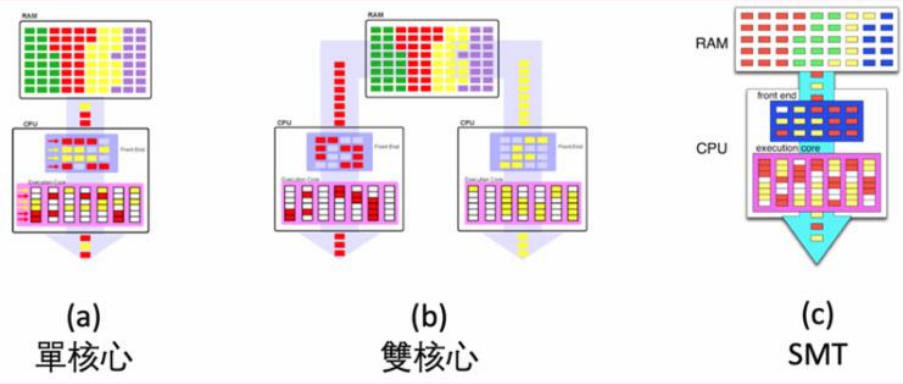
首先硬體對 CPU 呼叫中斷
CPU 收到中斷後會暫停當前 task 並將該 task 相關資料保存下來(如：暫存器值)
CPU 透過中斷向量表找到對應的 ISR
處理完中斷後，恢復先前儲存的暫存器，切換回去執行 task，

(3 pt) 11.Linux 為什麼要把驅動程式分成 top half 和 bottom half ? (3 Points)



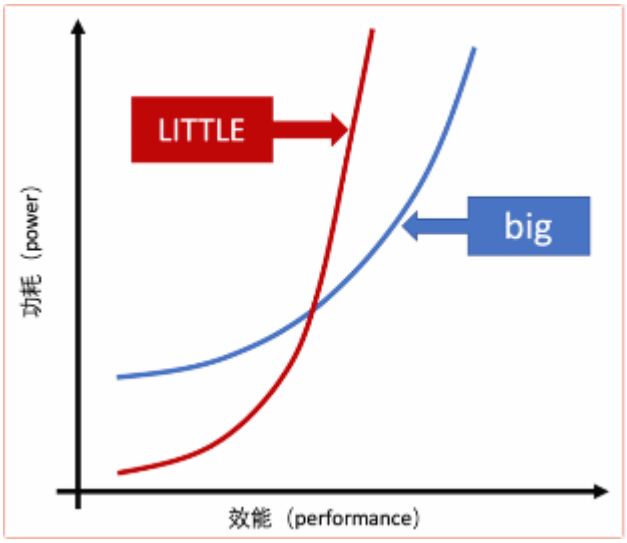
因為通常當中斷發生後，我們會想要盡快完成中斷發生時該做的事，這邊指的就是 top half 內做的事，一方面也可以提升 response time 而其他比較不重要的則可以留到 bottom half 去做。

(3 pt) 12.請解釋 SMT（同時多執行緒，Simultaneous multithreading）大致的運作原理。（hint：同時執行多個「行程」？一起執行的好處？）



SMT 在原本的處理器內多加了一套暫存器，讓原本的處理器可以同時處理多個行程，因為以往如果只執行單個行程，其實還會有很多運算資源沒被利用到，如果能夠同時執行多個行程，則可以使資源被更有效的利用

(3 pt) 13.請解釋 bigLITTLE 的原理（hint：小核心和大核心，主要是速度增加還是省電，如何達成）



將 處理器 分為 小核心和大核心

小核心：主要是執行一些較不需要效能的工作，可以透過低時脈的方式來達成省電的效果
大核心：當需要執行需要效能的工作，則可以使用時脈較高的大核心

(3 pt) 14.請問 32 條記憶體組成超寬通道（memory channel），和 32 條記憶體組成 8 條小通道的差異

因為我們的 CPU 正常情況下都不會利用到 32 條記憶體，而為了維護能同時存取這 32 條記憶體的存取，可能也犧牲了很多資源 所以我們可以改將記憶體幾個幾個一組，分為好幾個通道，而特定 CPU 就專門去存取某個通道的記憶體，可以有效的提升效能

(3 pt) 15.為什麼 CPU 剛啟動時 cache 沒有打開（hint：I/O）

因為需要先將 I/O 通道設為 non-cache

(3 pt) 16.請大略用組合語言，透過 system call 在螢幕上印出 “hello” 。（就是要跟附圖的方式意思一樣的程式碼）

```c
#include <stdio.h>
void main(){
    printf("Hello World");
}
```

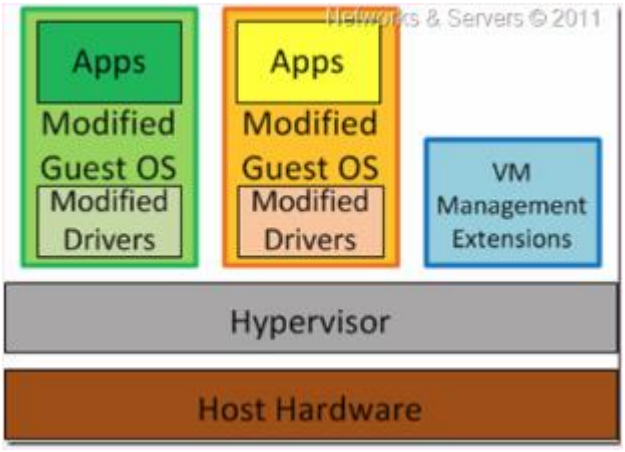
intel syntax:

```
mov rax, 0x2 // syscall number of write
mov rdi, 0x1 // stdout
mov rsi, address of "hello"
mov rdx, 0x6 // strlen("hello")
syscall
```

(3 pt) 17.什麼是 vdso（hint：核心將...跟使用者行程...，就是什麼樣的情況下會用 vdso，有什麼好處）

vdso 是核心將 一些較非機密的 function 從 kernel space 搬到 user space 並且去維護其在 vvar 中對應的資料 舉例來說像是 clockgettime 跟 getcpu 這種比較非機密的資料，這樣使用者存取的時候就不用進行 mode change，可以提升效能

(4 pt) 18.舉出一個例子說明 guest OS（就是被模擬機執行的那個 OS）最好要能與 vmm（virtual machine monitor，就是能執行 guest OS 的那個模擬機）溝通（例如：提供什麼樣的特殊指令，提供什麼樣的特殊驅動程式，為什麼要提供你所說的「額外機制」）



1.特權指令
2.trap
3.因為 Guest OS 如果使用特權指令 可能會得到並非想要的結果

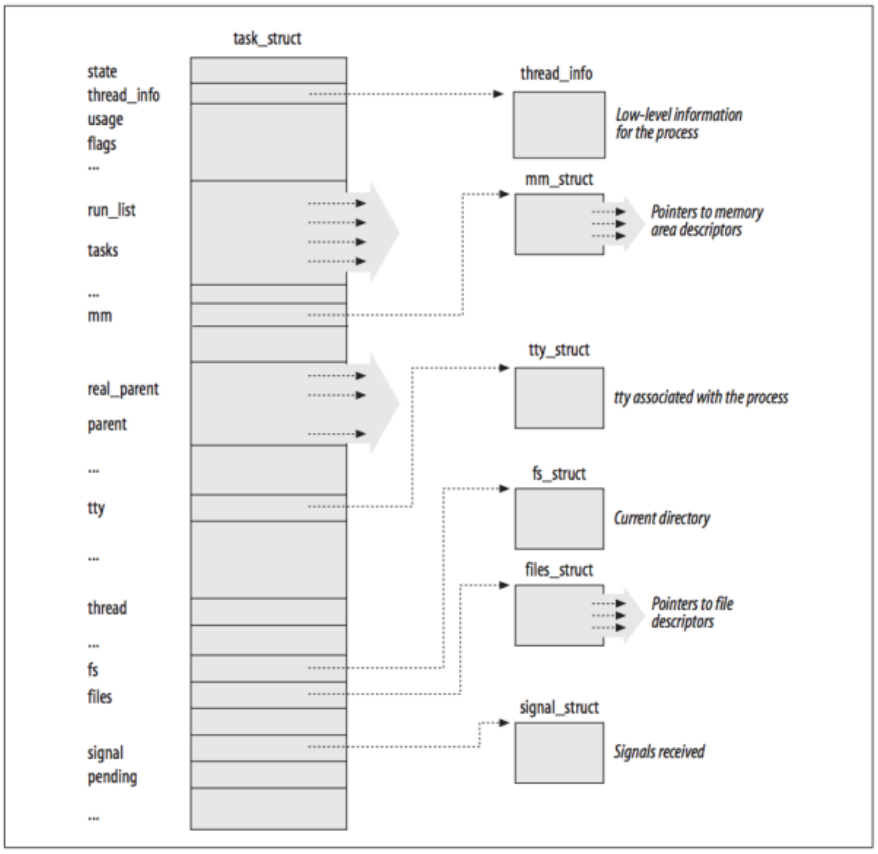
(3 pt) 19.請問如果取消 ASLR（位址空間組態隨機載入，Address space layout randomization）的話，可以進行怎樣的優化

可以針對比如說 libc 的位置，因為現在動態載入時，還需要透過 lazy-binding 機制，才能得知 libc function 在記憶體的位置， 如果我們可以一開始就知道位置，就不用再去進行動態載入了。

(3 pt) 20.請問如果讓 CPU-bound 的優先權比較高會造成什麼樣的現象（hint：注意一下，是「CPU」，稍微敘述一下為什麼會有這樣的現象。可以使用 C 代表電腦正在處理 CPU-bound，用 I 代表電腦正在處理 I/O bound）

如果今天 CPU-bound 的優先權較高的話，C 會比 I 先執行，然後 I 要等待 C 執行完畢 或者等 C 消耗掉他的回合時間，才能開始執行並且發出 I/O，然後等待 I/O 無法有效利用資源以及時間 如果我們能夠讓 IO-bound 的 task 先執行並發出 I/O，然後再換 C 執行，在等待的那段時間，我們就可以讓 C 先執行，藉此提昇效率

(3 pt) 21.在 Linux kernel 中，隸屬於同一個 process 的 thread 至少共用哪個物件？

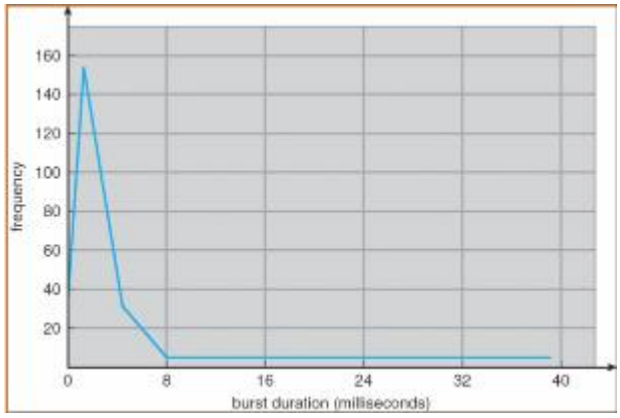


mm_struct

(3 pt) 22.舉一個例子說明，non-preemptive OS 在什麼樣的情境下效能會比 preemptive OS 好。大概說明原因。

Netware，若大部份的工作都是 I/O bound 的情況下(像是 FTP)，如果我們能讓每個 task 執行並且等待他發出 I/O 之後，馬上換下個 task 執行，可以非常有效的利用系統資源

(3 pt) 23.如果要讓 I/O 的效能提高，又要能夠有好一點的 response time，請問應該把 time slice 設定為多少？（hint：一個數字，說明為何要設定成那個數字）



8ms， 因為在這個時間之前大部分的 APP 都可以發出 I/O

(3 pt) 24.請問 Linux 2.4 是怎樣提高效能（hint：先定義 epoch 的轉換時機，然後累積了什麼，累積的東西跟優先權有何關係）

epoch 會在大部分 CPU bound task 的回合時間為 0 時轉換， 而 I/O bound 的剩餘回合時間會除以二後加上 time slice，因此永遠會有比較高的執行時間，I/O bound 的 task 永遠都會有比較高的優先權

(3 pt) 25.請問在 CFS 中有辦法提高 task 的優先權，使得他的優先權「絕對」高過 I/O 嗎？請說明原因

沒辦法，因為 I/O task 的時間，再進入 queue 時，會變成當前最小的 vruntime - X，因此 I/O 會有最小值

(3 pt) 26.什麼是 race condition（請寫清楚，不要英翻中。可以用舉例的方式說明）

通常是因為程式的執行順序與預期不同，或是重要指令間有執行上的時間差，導致程式有機會執行到一半出現非預期行為 ex. 兩邊 cpu 同時改到一個變數的值

(3 pt) 27.請解釋 critical section 一定要滿足 progress 的原因（hint：A 想進去，裡面沒人但… ）

因為如果不需要滿足 progress 如果 A 想進去，裡面沒人但還是不讓他進去(裡面永遠沒人)，這樣也符合 mutual exclusion

(3 pt) 28.A 可以 preempt B，如果要製造 critical section，可以怎樣做最簡單（hint：不可以使用 spinlock、semaphore 等，disable...）

B disable A

Mutex Type	Robustness	Relock	Unlock When Not Owner
NORMAL	non-robust	deadlock	undefined behavior
NORMAL	robust	deadlock	error returned
ERRORCHECK	either	error returned	error returned
RECURSIVE	either	recursive (see below)	error returned
DEFAULT	non-robust	undefined behavior†	undefined behavior†
DEFAULT	robust	undefined behavior†	error returned

(3 pt) 29.請問如果設定 mutex「不支援」巢狀鎖定的好處（換句話說，在什麼樣的情況下，你會用「不支援巢狀」）

如果 relock 會發生 deadlock 的情況下，我們若能選擇不支援巢狀，則我們可以即時發現問題所在

(3 pt) 30.DMA 和 CPU 都會更新 DRAM，請列出一種方法，可以確保 CPU 會看到最新的資料（hint：有硬體法跟軟體法，硬體法要想辦法把資料寫到ＸＸＸ，軟體法要想辦法取消ＸＸＸ）

硬體法要想辦法把資料寫到 CPU，軟體法要想辦法取消 cache line

(3 pt) 31.什麼是 readers-writers 問題？（hint：一定要寫出在平行化方面的優勢）

指一個東西，可以同時由多個 readers 讀，但 reader 不能跟 writer 同時存取該東西，也不能同時有兩個 writers。 如果僅須讀資料的話 平行化一次有很多 worker 讀資料，可以增加很多效能，因為不需要管 CS, race condition

(3 pt) 32.新「桓」結衣的注音（hint：可以寫在紙上）

Ⓤ注音符號表：ㄅㄆ ㄇ ㄈ ㄉ ㄊ ㄋ ㄌ ㄍ ㄎ ㄏ ㄐ ㄑ ㄒ ㄧ ㄨ ㄩ ㄚ ㄛ ㄜ ㄝ ㄞ ㄟ ㄠ ㄡ ㄢ ㄣ ㄤ ㄥ ㄦ

Ⓤ音韻表： ˉ聲、ˊ聲、ˇ聲、ˋ聲、˙輕聲

Ⓤ如：羅，ㄌㄨㄛˊ ˊ聲

(3 pt) 33.How was your night? Did you sleep well? (3 Points)

☑My knight is very brave.

☑I heard my alarm clock bleeping this morning.

☑想著如何脫離宅宅

☑整晚想著如何表白

☑覺得老肥宅壓床（如授課老師）是很恐怖的事情

(3 pt) 34.今天天氣如何？

☑fine, thank you

☑you are welcome

☑以何問天？天之道，損有餘而補不足。人之道，則不然，損不足以奉有餘。孰能有餘以奉天下，唯有道者。

☑這問題並不完整，我不曉得指的是嘉義還是紐約

☑本宅一直在讀 OS 還沒看到陽光

☑反正明天的天氣不會比今天更糟

☑老師您會不會覺得「不問蒼生，問蒼天」是很幼稚的？

☑難道我不能不完全不回答這個問題嗎？

☑天氣如何是不是應該問 Siri 呢？

☑今天的天氣如同我的心情，你知道的。

☑考試中，無法跑出去看，恕我無法回答