政大資科系

作業系統

Operating System

廖峻鋒

cfliao@nccu.edu.tw

Operating System

Midterm

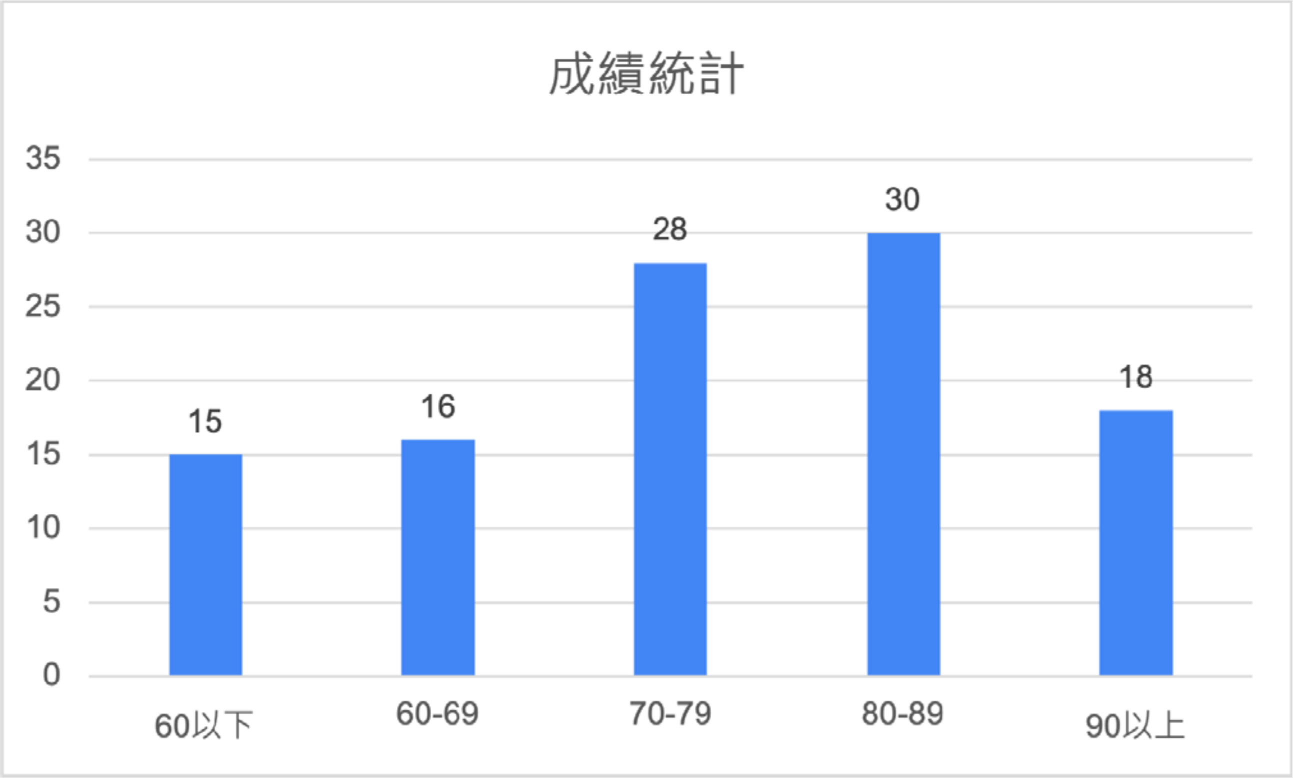
Chun-Feng Liao

廖峻鋒

Department of Computer Science

National Chengchi University

- 預警: 50分以下

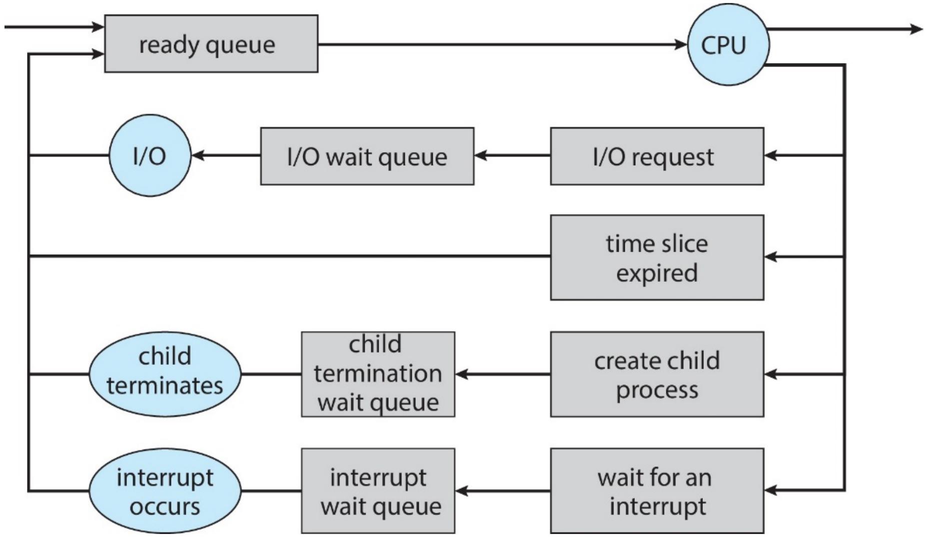


1

- A
- Which of the following statement about gRPC is false? (a) the gRPC stub and server must use the same programming language (b) gRPC stub and server communicates with Protocol Buffer over HTTP/2 (c) the .proto file is used to generate gRPC stub and server (d) gRPC is a modern high performance Remote Procedure Call (RPC) framework

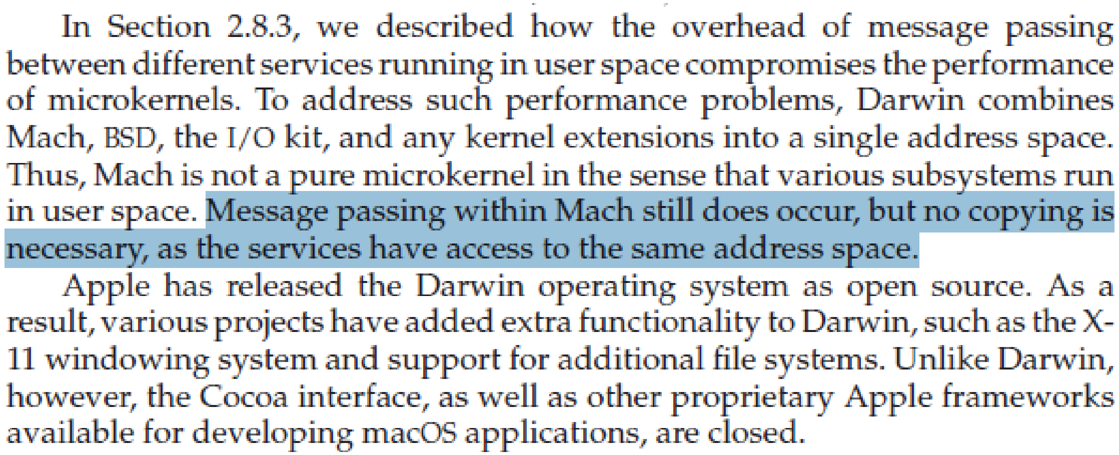
2

- D
- Which of the following is not a reason that causes a process to be switched out of CPU (a) I/O request (b) create a child process (c) wait for an interrupt (d) one of its child process is terminated



3

- C
- Which of the following are not correct regarding to Darwin? (a) Darwin is not a pure Microkernel (b) Darwin combines Mach, BSD, the I/O kit and kernel extensions into a single address space (c) ~~Copying is needed~~ for messaging passing in the Mach kernel (d) Darwin provide two types of system-call interfaces.
- 課本p.89

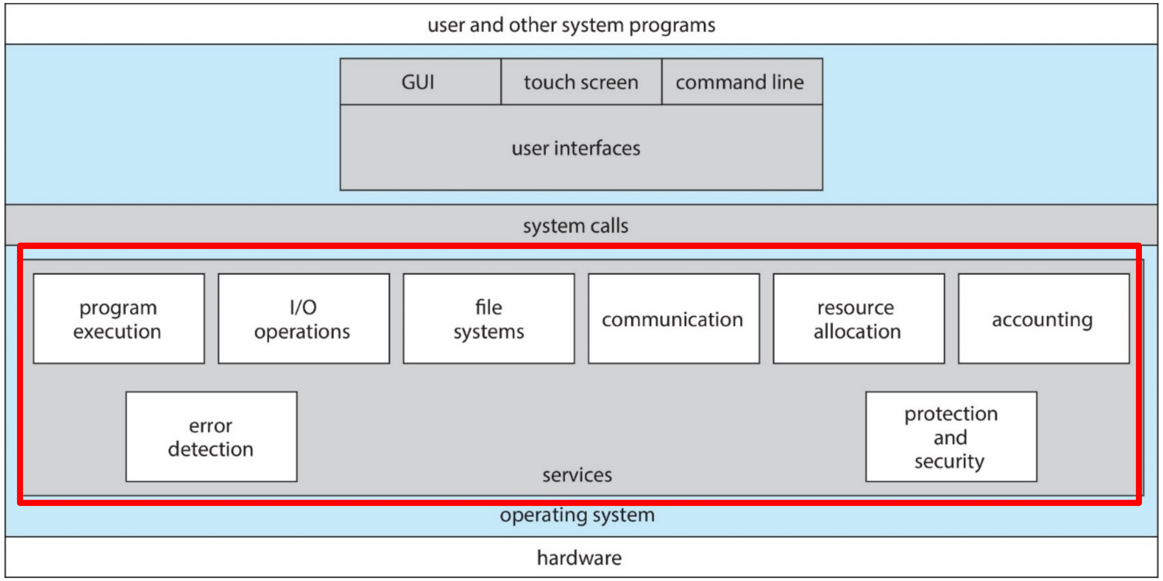


In Section 2.8.3, we described how the overhead of message passing between different services running in user space compromises the performance of microkernels. To address such performance problems, Darwin combines Mach, BSD, the I/O kit, and any kernel extensions into a single address space. Thus, Mach is not a pure microkernel in the sense that various subsystems run in user space. Message passing within Mach still does occur, but no copying is necessary, as the services have access to the same address space.

Apple has released the Darwin operating system as open source. As a result, various projects have added extra functionality to Darwin, such as the X-11 windowing system and support for additional file systems. Unlike Darwin, however, the Cocoa interface, as well as other proprietary Apple frameworks available for developing macOS applications, are closed.

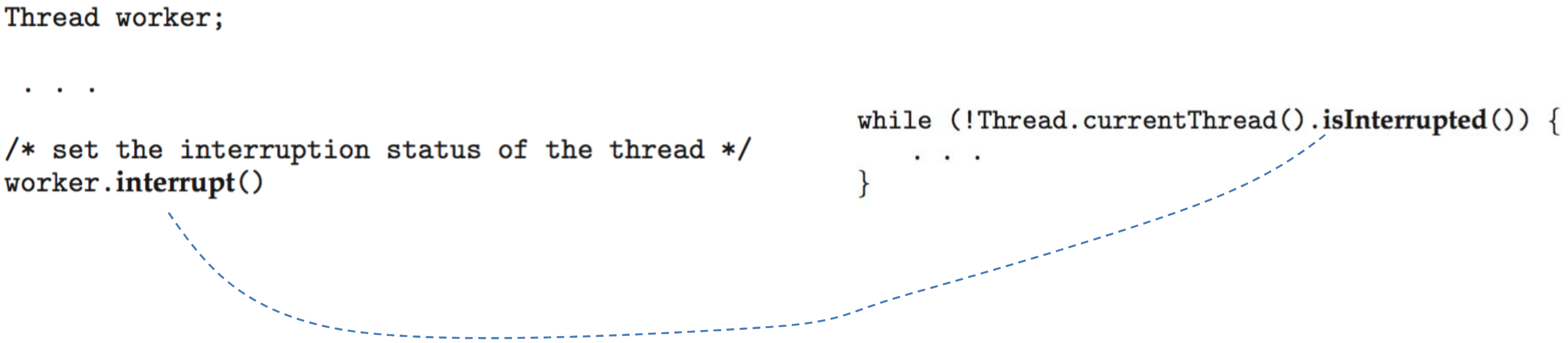
4

- D
- Which of the following are not a system service of an OS? (a) File system (b) Communication (c) Resource allocation (d) Shell



5

- D
- Which of the following is false? (a) In asynchronous cancellation mode, a thread is terminated immediately after it receive a cancellation request (b) The default cancellation mode in PThread is deferred. (c) Asynchronous cancellation mode is not recommended in PThread (d) In Java, the deferred cancellation is implemented as the ~~defer()~~ method



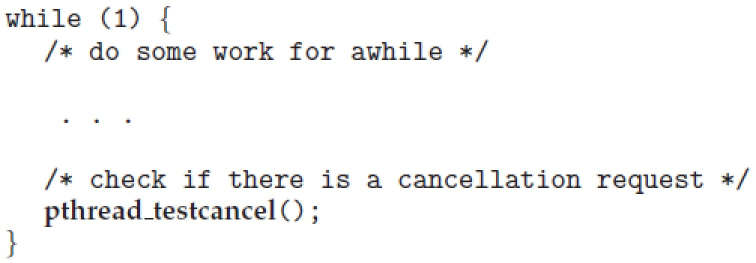
```
Thread worker;

 . . .

/* set the interruption status of the thread */
worker.interrupt()
```

```
while (!Thread.currentThread().isInterrupted()) {
    . . .
}
```

6

- D
- Which of the following Pthread function is used to respond to a cancellation request using deferred cancellation? (a) pthread_attr_init (b) pthread_create (c) pthread_join (d) pthread_testcancel

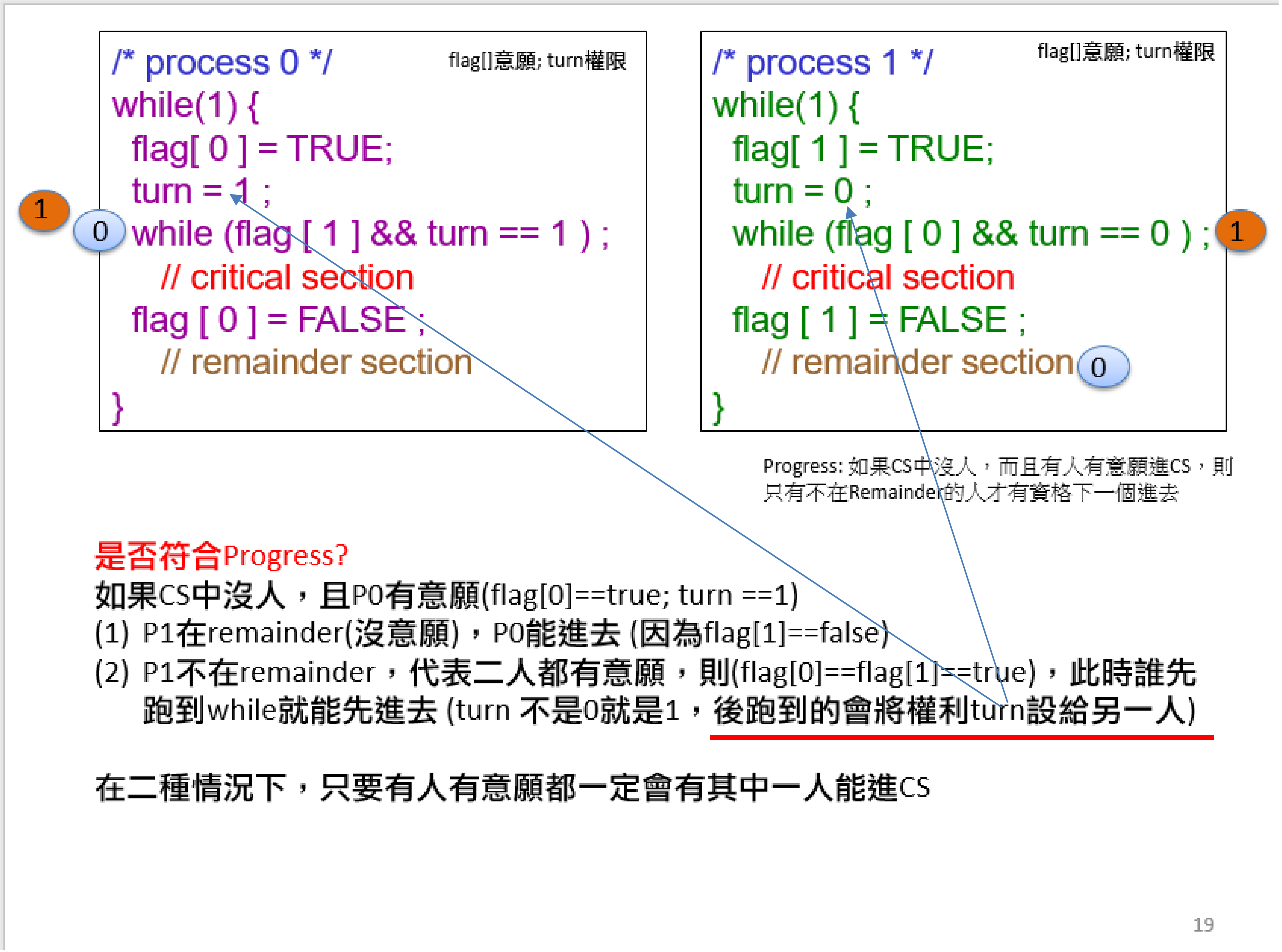
課本 p.191



```
while (1) {
   /* do some work for awhile */

    . . .

   /* check if there is a cancellation request */
   pthread_testcancel();
}
```

7

- C
- Which of the following statement is false? (a) We can simply disable interrupts to prevent race condition in a single core environment (b) In the non-preemptive kernel, a program runs until it exits kernel mode, blocks, or voluntarily yields CPU (c) In the Peterson’s solution, if flag[0]==flag[1]==true, then ~~the slower~~ process that hit the while loop may enter the critical section (d) In the Peterson’s solution, flag[j] must be false if the process j is in the remainder section.



flag[]意願; turn權限

```
/* process 0 */
while(1) {
  flag[ 0 ] = TRUE;
  turn = 1 ;
  while (flag [ 1 ] && turn == 1 ) ;
    // critical section
  flag [ 0 ] = FALSE ;
    // remainder section
}
```

flag[]意願; turn權限

```
/* process 1 */
while(1) {
  flag[ 1 ] = TRUE;
  turn = 0 ;
  while (flag [ 0 ] && turn == 0 ) ;
    // critical section
  flag [ 1 ] = FALSE ;
    // remainder section
}
```

Progress: 如果CS中沒人，而且有人有意願進CS，則只有不在Remainder的人才有資格下一個進去

是否符合Progress?

如果CS中沒人，且P0有意願(flag[0]==true; turn ==1)

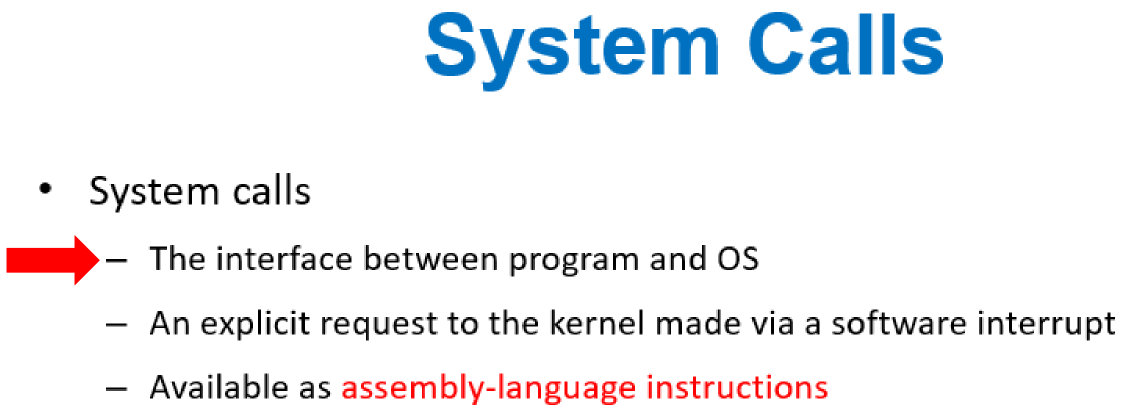
(1) P1在remainder(沒意願)，P0能進去 (因為flag[1]==false)

(2) P1不在remainder，代表二人都有意願，則(flag[0]==flag[1]==true)，此時誰先跑到while就能先進去 (turn 不是0就是1，後跑到的會將權利turn設給另一人)

在二種情況下，只要有人有意願都一定會有其中一人能進CS

8

- C
- The interface between a program and the OS is called (a) shell (b) programming interface (c) system call (d) middleware

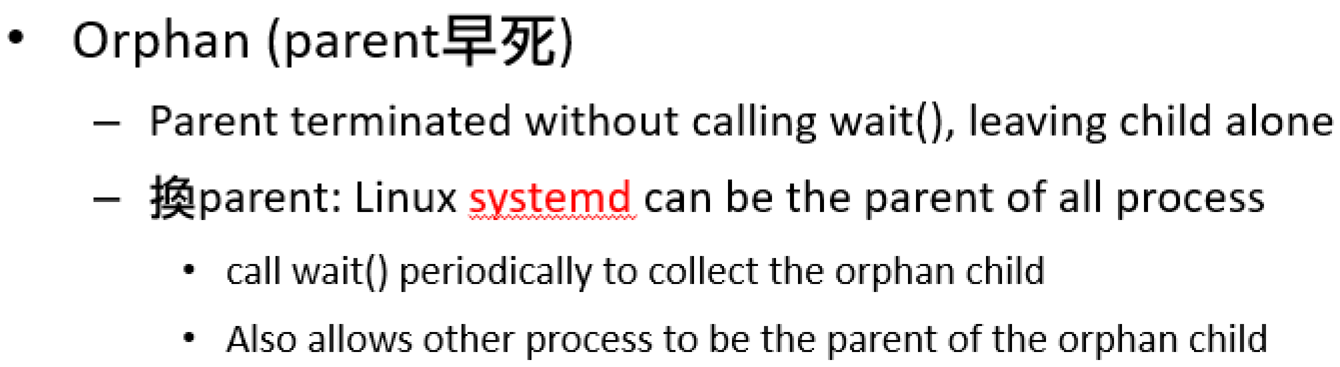


System Calls

- System calls
  - The interface between program and OS
  - An explicit request to the kernel made via a software interrupt
  - Available as assembly-language instructions

9

- A
- Parent terminated without calling wait(), leaving child alone is called (a) orphan (b) daemon (c) zombie (d) none of the above

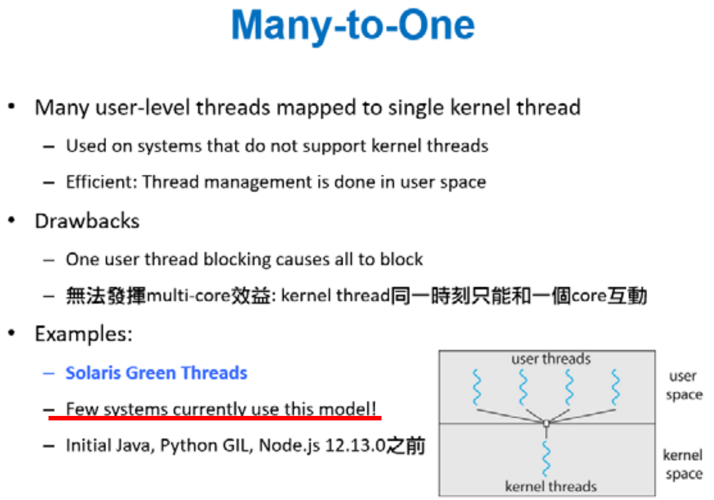


- Orphan (parent早死)
  - Parent terminated without calling wait(), leaving child alone
  - 換parent: Linux systemd can be the parent of all process
    - call wait() periodically to collect the orphan child
    - Also allows other process to be the parent of the orphan child

10

Many-To-One

- A
- Which of the following is false: (a) ~~One~~-to-One threading is rarely used in current systems (b) Implicit threading usually use Many-to-Many threading model (c) One-to-One model is slow as creating a user-level thread also creates a kernel thread (d) JVM on Windows uses One-to-One threading model



Many-to-One

- Many user-level threads mapped to single kernel thread
  - Used on systems that do not support kernel threads
  - Efficient: Thread management is done in user space
- Drawbacks
  - One user thread blocking causes all to block
  - 無法發揮multi-core效益: kernel thread同一時刻只能和一個core互動
- Examples:
  - Solaris Green Threads
  - Few systems currently use this model!
  - Initial Java, Python GIL, Node.js 12.13.0之前

11

A

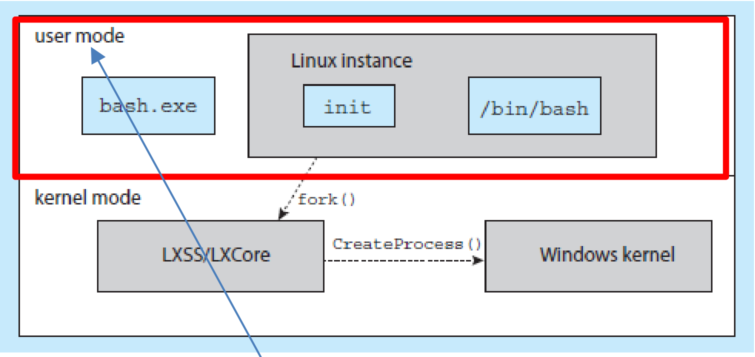
- Which of the following true? (a) The binder framework is an IPC mechanism supported by Android (b) In Mach Message Passing, the ~~Notify~~ Task-Self Port port is used to send messages to the kernel (c) In ~~Mach Message Passing~~, ALPC the section object is used to transfer large objects (d) In ~~Windows ALPC~~, Mach Message Passing the Bootstrap server is used to host a set of registered bootstrap ports.

課後閱讀 (會考)

- P115 手持裝置的Multitasking
- P135 Mach Messaging Passing: Task Self port的用途
- P136 什麼是Bootstrap server
- P136 Mach Message Passing如何傳送大檔?
- P138 ALPC是什麼? 作用為何?
- P139 ALPC為什麼需要section object?
- P151 Android中提供什麼支援來開發RPC?

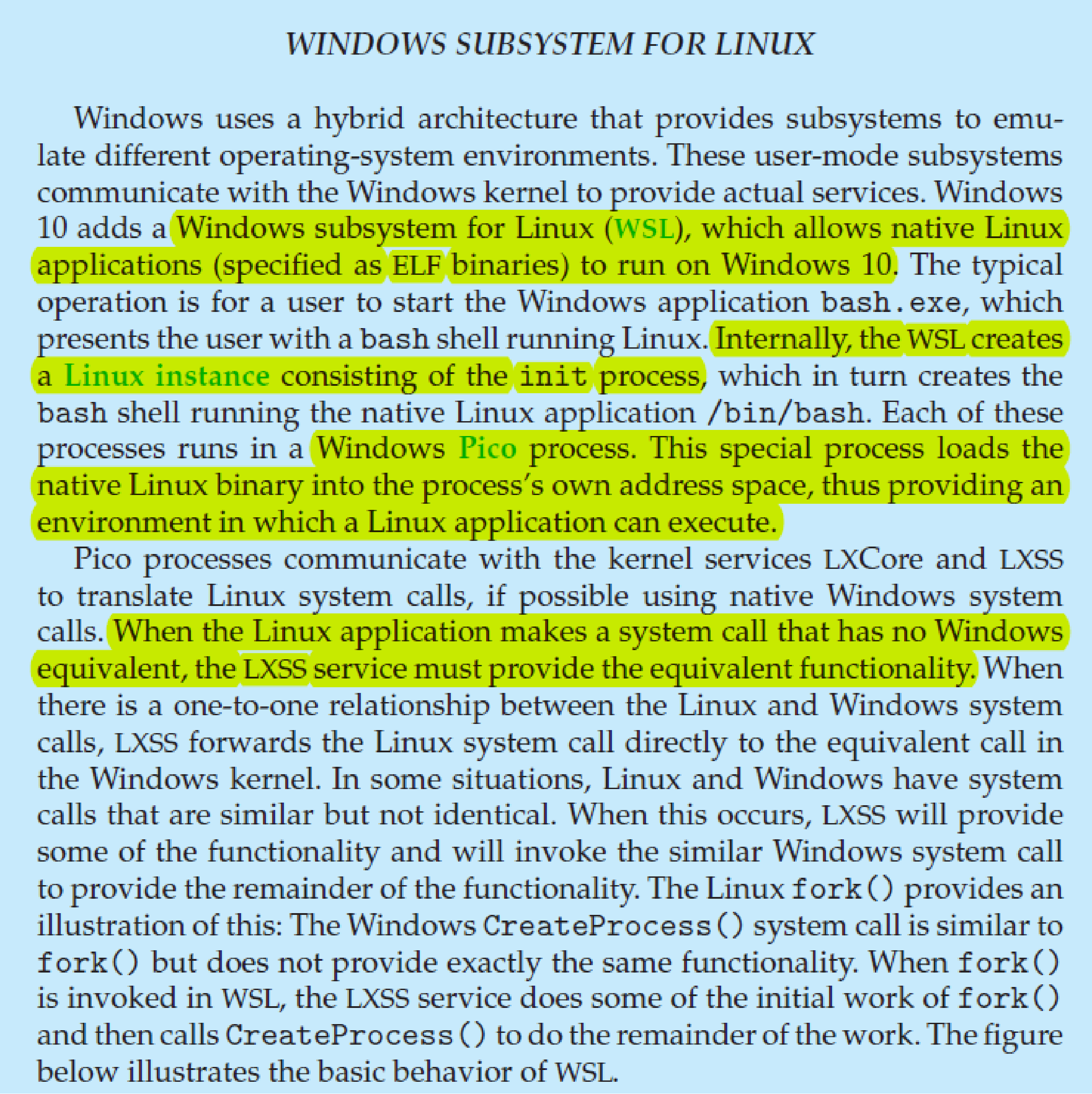
課本第91頁

12



- B
- About WSL (Windows Subsystem for Linux), which of the following statement is false? (a) WSL allows ELF binaries to run on Windows (b) Internally, WSL creates a Linux instance containing the init process; the init process runs in ~~kernel mode~~ (c) LXSS is used to provide the system call implementations that are missing in Windows (d) The Windows Pico process loads the ELF binary into its own address space so that the ELF can be executed in this space.

課本第91頁



WINDOWS SUBSYSTEM FOR LINUX

Windows uses a hybrid architecture that provides subsystems to emulate different operating-system environments. These user-mode subsystems communicate with the Windows kernel to provide actual services. Windows 10 adds a Windows subsystem for Linux (WSL), which allows native Linux applications (specified as ELF binaries) to run on Windows 10. The typical operation is for a user to start the Windows application `bash.exe`, which presents the user with a `bash` shell running Linux. Internally, the WSL creates a Linux instance consisting of the `init` process, which in turn creates the `bash` shell running the native Linux application `/bin/bash`. Each of these processes runs in a Windows Pico process. This special process loads the native Linux binary into the process's own address space, thus providing an environment in which a Linux application can execute.

Pico processes communicate with the kernel services LXCore and LXSS to translate Linux system calls, if possible using native Windows system calls. When the Linux application makes a system call that has no Windows equivalent, the LXSS service must provide the equivalent functionality. When there is a one-to-one relationship between the Linux and Windows system calls, LXSS forwards the Linux system call directly to the equivalent call in the Windows kernel. In some situations, Linux and Windows have system calls that are similar but not identical. When this occurs, LXSS will provide some of the functionality and will invoke the similar Windows system call to provide the remainder of the functionality. The Linux `fork()` provides an illustration of this: The Windows `CreateProcess()` system call is similar to `fork()` but does not provide exactly the same functionality. When `fork()` is invoked in WSL, the LXSS service does some of the initial work of `fork()` and then calls `CreateProcess()` to do the remainder of the work. The figure below illustrates the basic behavior of WSL.

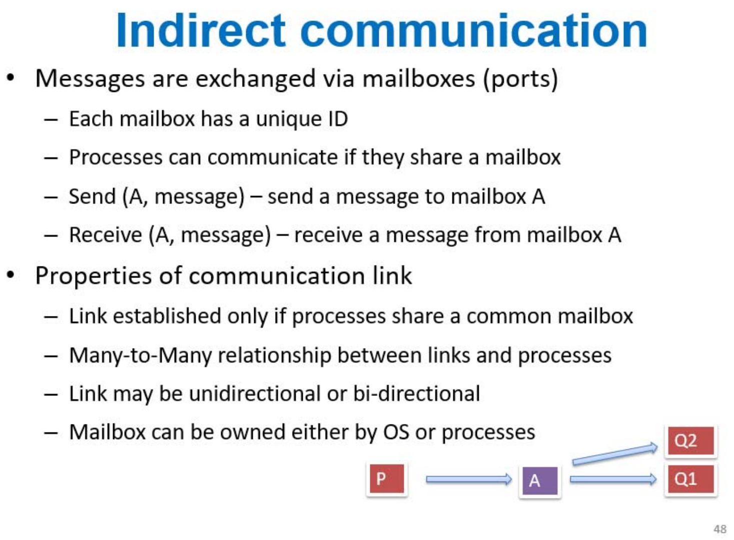
13

- B
- Which of the following is true? (a) In Windows, each JVM maps to a user ~~thread~~ (b) the Linux clone() API uses a set of flags to determine how much sharing between parent and child processes (c) OpenMP identifies “parallel regions” as blocks of code that may run in parallel using a pair of functions~~: parallel_regions_start() and parallel_regions_end().~~ (d) The dispatch queues in GCD are only used to run concurrent tasks.

serial and concurrent.

14

- A
- Which of the following is not correct about Indirect Communication? (a) processes must ~~name each other~~ explicitly (b) Messages are exchanged via mailboxes (c) Link established only if processes share a common mailbox (d) Mailbox can be owned either by OS or processes



Indirect communication

- Messages are exchanged via mailboxes (ports)
  - Each mailbox has a unique ID
  - Processes can communicate if they share a mailbox
  - Send (A, message) – send a message to mailbox A
  - Receive (A, message) – receive a message from mailbox A
- Properties of communication link
  - Link established only if processes share a common mailbox
  - Many-to-Many relationship between links and processes
  - Link may be unidirectional or bi-directional
  - Mailbox can be owned either by OS or processes

48

15

- D
- Typically, implicit threading is realized using which model? (a) one-to-one (b) Many-to-One (c) One-To-Many (d) Many-to-Many

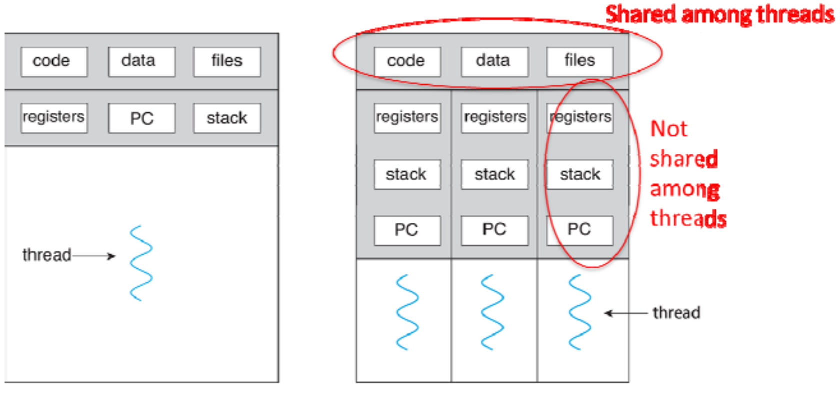
16

- C
- Which of the following is false regarding to the Peterson’s solution to the Critical Section Problem? (a) support only two processes (b) the solution may become invalid when the CPU reorder the executing instructions (c) use the flag[] array to denote ~~the right~~ to enter the critical section (d) meet the mutual exclusion requirement from pure software’s view point.

意願

17

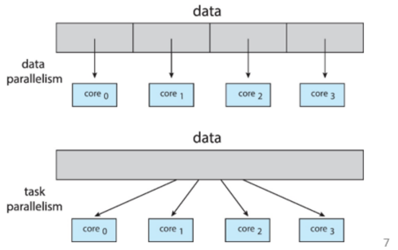
- b
- Which of the following statement is false? (a) “/sbin/init splash” is a symbolic link to systemd (b) ~~Concurrency~~ means to perform more than one task simultaneously (c) the opened files of a process is shared by all of the process’s threads (d) Data Parallelism means to perform the same task on different data



single-threaded process

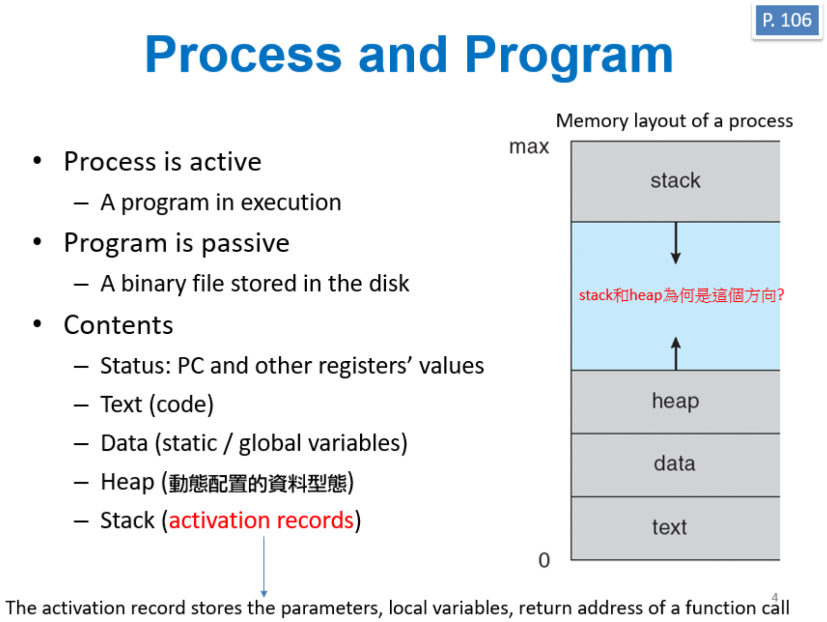
multithreaded process

- Data Parallelism (map)
  - Perform the same task on different data
- Task Parallelism
  - Perform different tasks on the same data



18

- C
- Which of the following regarding to the memory layout of a process is false? (a) the parameter values and the return address store in the stack area (b) the uninitialized global variables are stored in the bss area (c) the ~~global variables~~ are stored in the heap area (d) the stack area is also known as activation records



P. 106

Process and Program

- Process is active
  - A program in execution
- Program is passive
  - A binary file stored in the disk
- Contents
  - Status: PC and other registers' values
  - Text (code)
  - Data (static / global variables)
  - Heap (動態配置的資料型態)
  - Stack (activation records)

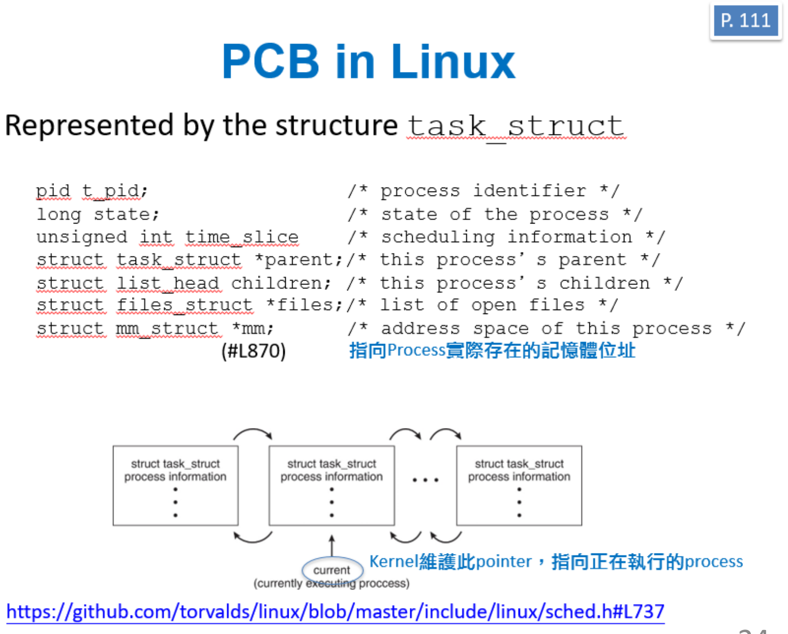
The activation record stores the parameters, local variables, return address of a function call

4

19

- D
- Which of the following statement is false? (a) Each process is associated with a PCB (b) In Linux, the PCB information can be accessed via /proc (c) the pointers to the opened files are stored in PCB (d) PCB information such as parent, child, and register values are usually kept in a structure called ~~mm_struct~~.

  task_struct



P. 111

PCB in Linux

Represented by the structure task_struct

```
pid t_pid;                  /* process identifier */
long state;                 /* state of the process */
unsigned int time_slice     /* scheduling information */
struct task_struct *parent;/* this process’ s parent */
struct list_head children; /* this process’ s children */
struct files_struct *files;/* list of open files */
struct mm_struct *mm;       /* address space of this process */
```

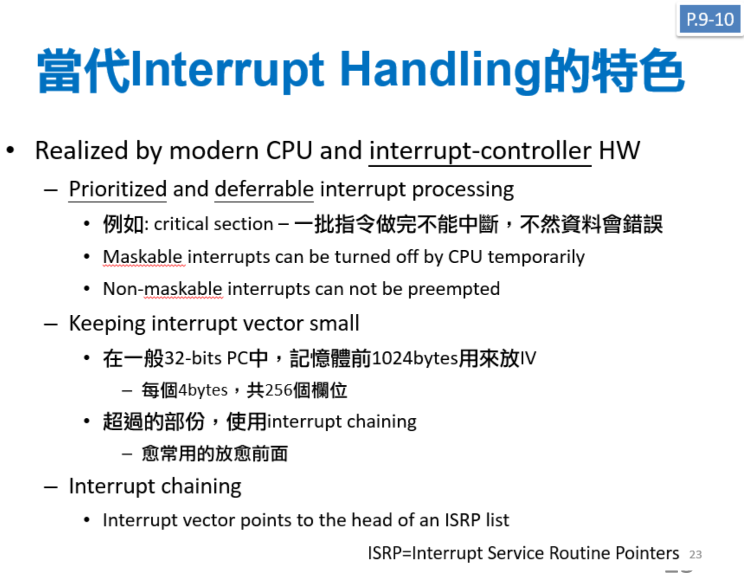
(#L870) 指向Process實際存在的記憶體位址

https://github.com/torvalds/linux/blob/master/include/linux/sched.h#L737

20

ISRP=Interrupt Service Routine Pointers

- C
- Which of the following statement about Interrupt is false? (a) Interrupt chaining refers to the design that some interrupt vector items points to the head of an ISRP list (b) Modern OS tends to keep the interrupt vector small (c) ~~Non-maskable~~ interrupts can be turned off by CPU temporarily (d) Upper half interrupt should be processed as soon as possible



P.9-10

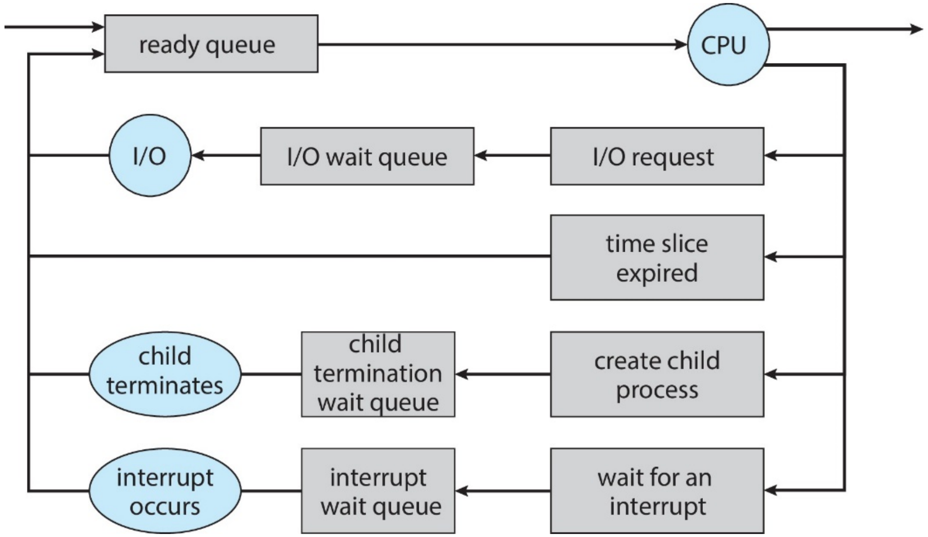
當代Interrupt Handling的特色

- Realized by modern CPU and interrupt-controller HW
  - Prioritized and deferrable interrupt processing
    - 例如: critical section – 一批指令做完不能中斷，不然資料會錯誤
    - Maskable interrupts can be turned off by CPU temporarily
    - Non-maskable interrupts can not be preempted
  - Keeping interrupt vector small
    - 在一般32-bits PC中，記憶體前1024bytes用來放IV
      - 每個4bytes，共256個欄位
    - 超過的部份，使用interrupt chaining
      - 愈常用的放愈前面
  - Interrupt chaining
    - Interrupt vector points to the head of an ISRP list

ISRP=Interrupt Service Routine Pointers 23

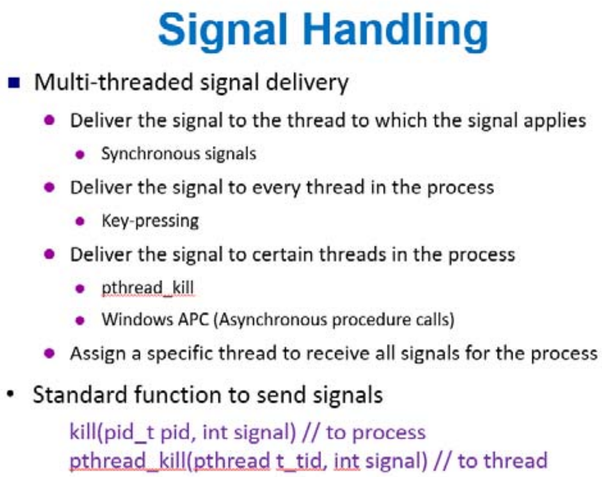
21

- C
- When a process being switched out of the CPU, which of the following situations does not cause the process being added to the wait queue? (a) I/O request (b) create child process (c) time slice expired (d) wait for an interrupt



22

- A
- Which of the following options ~~to deliver signals in multithreaded program~~ should be applied to a synchronous signal? (a) deliver the signal to the thread to which the signal applies (b) deliver the signal to every thread in the process (c) deliver the signal to certain threads in the process (d) assign a specific thread to receive all signals for the process



Signal Handling

- Multi-threaded signal delivery
  - Deliver the signal to the thread to which the signal applies
    - Synchronous signals
  - Deliver the signal to every thread in the process
    - Key-pressing
  - Deliver the signal to certain threads in the process
    - pthread_kill
    - Windows APC (Asynchronous procedure calls)
  - Assign a specific thread to receive all signals for the process
- Standard function to send signals

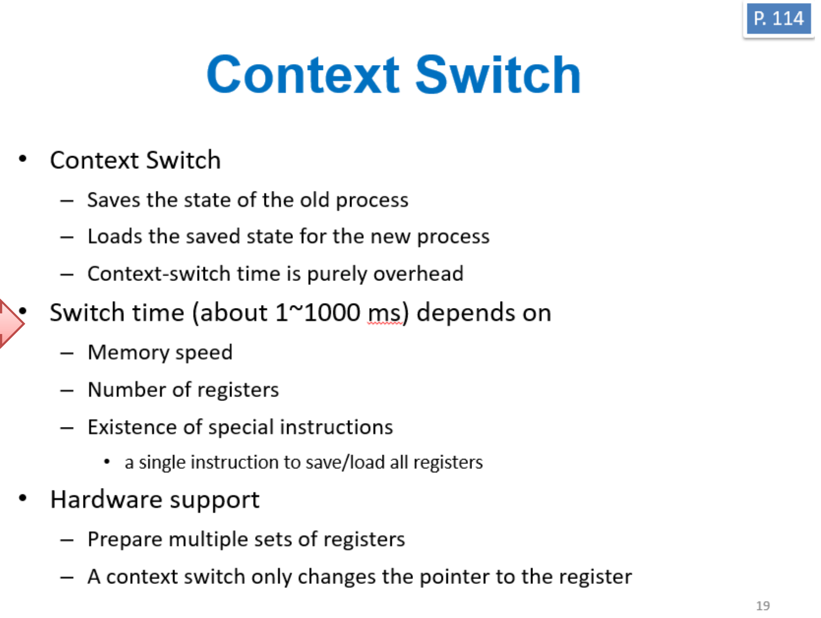
  kill(pid_t pid, int signal) // to process
  pthread_kill(pthread t_tid, int signal) // to thread

23

- A
- Which of the following statements regarding Java threads is false? (a) The ~~run()~~ method should be invoked by the client program after to start the execution a thread in the Java virtual machine (b) In Java, the “implementing Runnable interface” approach is more preferable than “extending the Thread class” (c) the interrupt() method of Thread is used for cancellation handling in Java (d) Java’s implicit threading mechanism is realized by the Executor class

24

- A
- Which of the following does not affect the context switch time? (a) memory size (b) number of registers (c) special instructions (d) all of the above affect the context switching time



P. 114

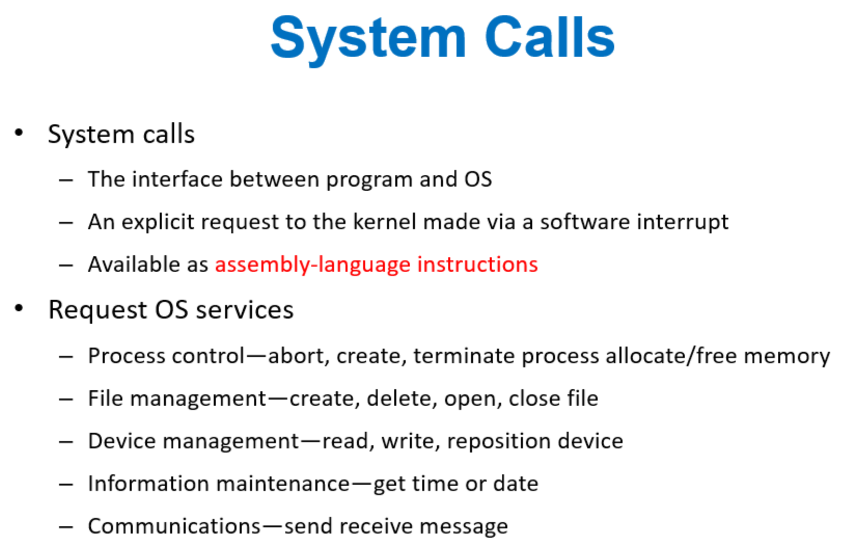
Context Switch

- Context Switch
  - Saves the state of the old process
  - Loads the saved state for the new process
  - Context-switch time is purely overhead
- Switch time (about 1~1000 ms) depends on
  - Memory speed
  - Number of registers
  - Existence of special instructions
    - a single instruction to save/load all registers
- Hardware support
  - Prepare multiple sets of registers
  - A context switch only changes the pointer to the register

19

25

- D
- Which of the following is not a type of system call (a) process control (b) file management (c) device management (d) user interfaces

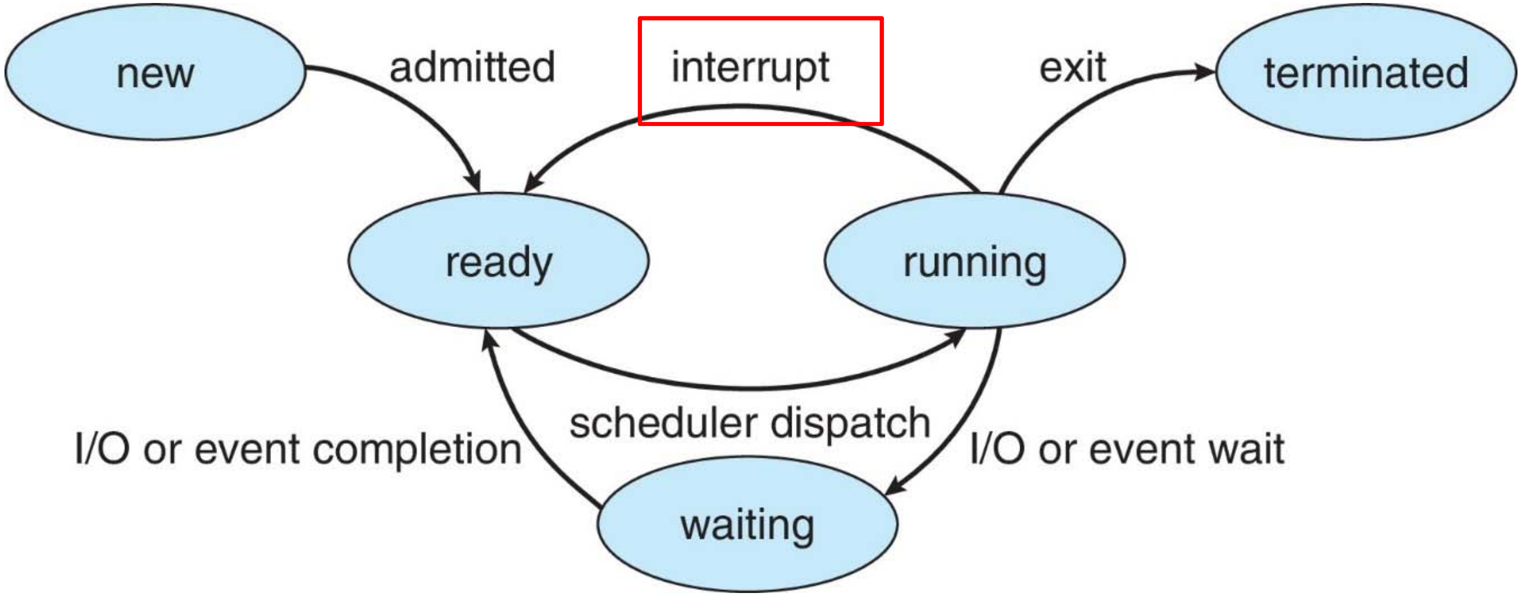


System Calls

- System calls
  - The interface between program and OS
  - An explicit request to the kernel made via a software interrupt
  - Available as assembly-language instructions
- Request OS services
  - Process control—abort, create, terminate process allocate/free memory
  - File management—create, delete, open, close file
  - Device management—read, write, reposition device
  - Information maintenance—get time or date
  - Communications—send receive message

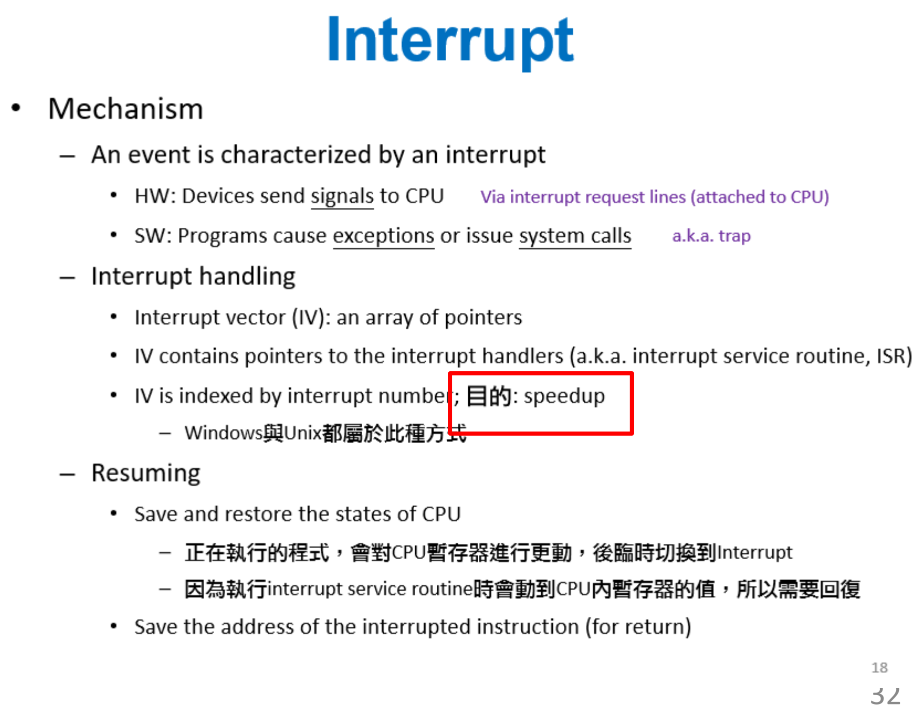
26

- B
- Given that a process is in the Running state. If it receives an interrupt, it becomes in which state? (a) New (b) Ready (c) Waiting (d) Sleeping



27

- B
- The reason that the interrupt vector is indexed by numbers is to increase (a) Stability (b) Speed (c) Security (d) Scalability



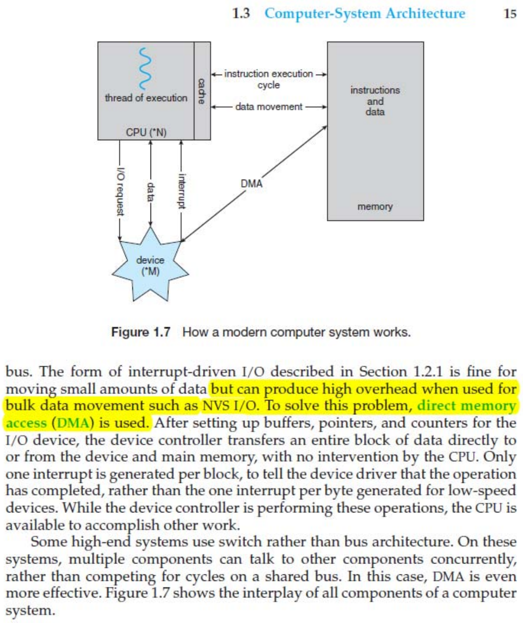
Interrupt

- Mechanism
  - An event is characterized by an interrupt
    - HW: Devices send signals to CPU   Via interrupt request lines (attached to CPU)
    - SW: Programs cause exceptions or issue system calls   a.k.a. trap
  - Interrupt handling
    - Interrupt vector (IV): an array of pointers
    - IV contains pointers to the interrupt handlers (a.k.a. interrupt service routine, ISR)
    - IV is indexed by interrupt number; 目的: speedup
      - Windows與Unix都屬於此種方式
  - Resuming
    - Save and restore the states of CPU
      - 正在執行的程式，會對CPU暫存器進行更動，後臨時切換到Interrupt
      - 因為執行interrupt service routine時會動到CPU內暫存器的值，所以需要回復
    - Save the address of the interrupted instruction (for return)

18

28

- C
- _____ is used to reduce the high overhead of CPU in bulk data movement between the memory and devices. (a) NUMA (b) SMP (c) DMA (d) PSW


thread of execution
cache
CPU (*N)
instruction execution cycle
data movement
instructions and data
memory
DMA
I/O request
data
interrupt
device (*M)

1.3 Computer-System Architecture 15

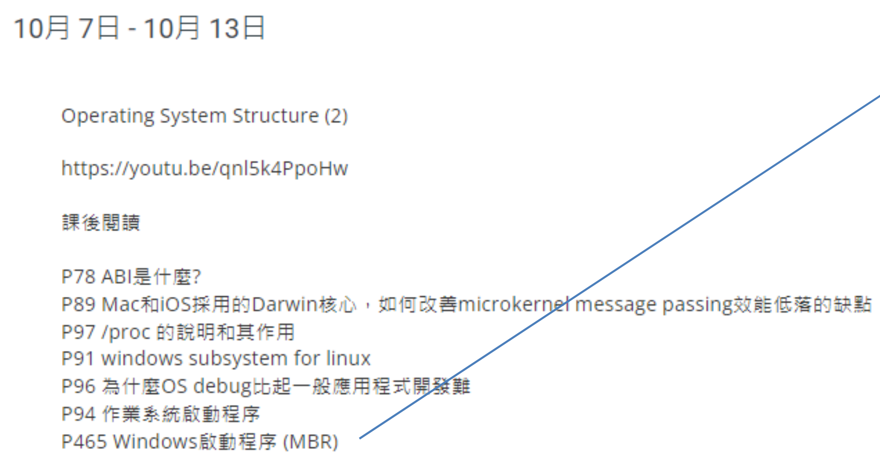
Figure 1.7 How a modern computer system works.

bus. The form of interrupt-driven I/O described in Section 1.2.1 is fine for moving small amounts of data but can produce high overhead when used for bulk data movement such as NVS I/O. To solve this problem, direct memory access (DMA) is used. After setting up buffers, pointers, and counters for the I/O device, the device controller transfers an entire block of data directly to or from the device and main memory, with no intervention by the CPU. Only one interrupt is generated per block, to tell the device driver that the operation has completed, rather than the one interrupt per byte generated for low-speed devices. While the device controller is performing these operations, the CPU is available to accomplish other work.

Some high-end systems use switch rather than bus architecture. On these systems, multiple components can talk to other components concurrently, rather than competing for cycles on a shared bus. In this case, DMA is even more effective. Figure 1.7 shows the interplay of all components of a computer system.

29

- D
- In Windows operating systems, the boot code and the partition table are placed in the first logical block of hard disk. This place is called (a) Boot disk (b) Boot sector (c) Boot partition (d) Master Boot Record



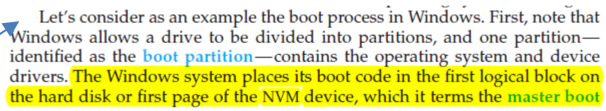
10月 7日 - 10月 13日

Operating System Structure (2)

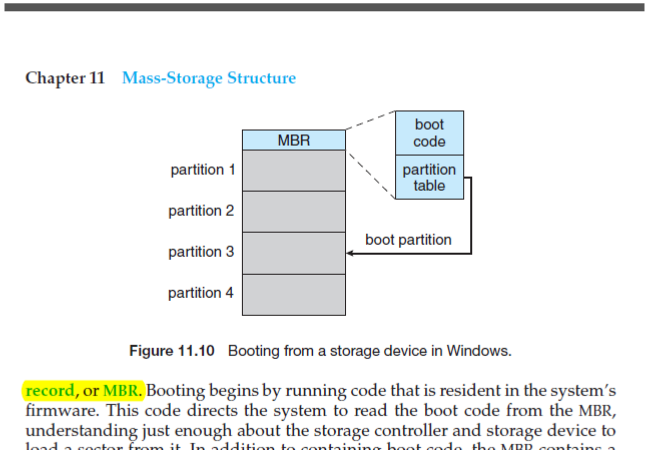
https://youtu.be/qnl5k4PpoHw

課後閱讀

P78 ABI是什麼?
P89 Mac和iOS採用的Darwin核心，如何改善microkernel message passing效能低落的缺點
P97 /proc 的說明和其作用
P91 windows subsystem for linux
P96 為什麼OS debug比起一般應用程式開發難
P94 作業系統啟動程序
P465 Windows啟動程序 (MBR)



Let's consider as an example the boot process in Windows. First, note that Windows allows a drive to be divided into partitions, and one partition—identified as the boot partition—contains the operating system and device drivers. The Windows system places its boot code in the first logical block on the hard disk or first page of the NVM device, which it terms the master boot



Chapter 11 Mass-Storage Structure

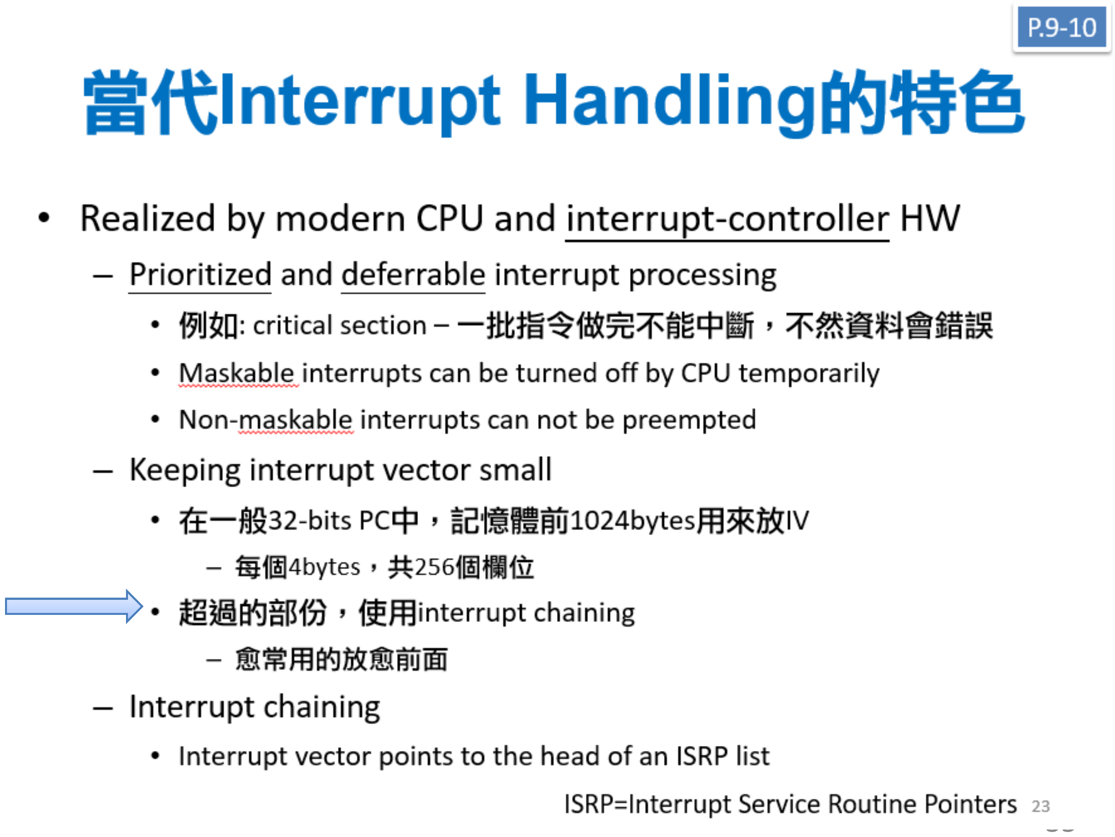
boot
code
MBR
partition
table
partition 1
partition 2
boot partition
partition 3
partition 4

Figure 11.10 Booting from a storage device in Windows.

record, or MBR. Booting begins by running code that is resident in the system's firmware. This code directs the system to read the boot code from the MBR, understanding just enough about the storage controller and storage device to load a sector from it. In addition to containing boot code, the MBR contains a

30

- C
- Which of the following mechanism is used to keep the interrupt vector small? (a) DMA (b) Interrupt request line (c) Interrupt chaining (d) Interrupt context



P.9-10

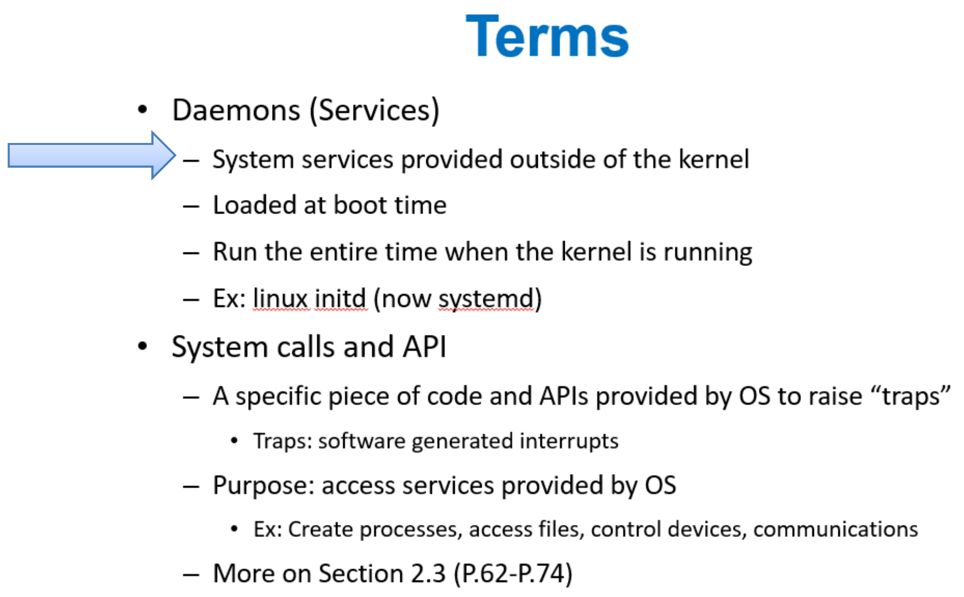
當代Interrupt Handling的特色

- Realized by modern CPU and interrupt-controller HW
  - Prioritized and deferrable interrupt processing
    - 例如: critical section – 一批指令做完不能中斷，不然資料會錯誤
    - Maskable interrupts can be turned off by CPU temporarily
    - Non-maskable interrupts can not be preempted
  - Keeping interrupt vector small
    - 在一般32-bits PC中，記憶體前1024bytes用來放IV
      - 每個4bytes，共256個欄位
    - 超過的部份，使用interrupt chaining
      - 愈常用的放愈前面
  - Interrupt chaining
    - Interrupt vector points to the head of an ISRP list

ISRP=Interrupt Service Routine Pointers 23

31

- A
- System services provided outside of the kernel are called (a) daemons (b) system call (c) interrupt (d) middleware



Terms

- Daemons (Services)
  - System services provided outside of the kernel
  - Loaded at boot time
  - Run the entire time when the kernel is running
  - Ex: linux initd (now systemd)
- System calls and API
  - A specific piece of code and APIs provided by OS to raise “traps”
    - Traps: software generated interrupts
  - Purpose: access services provided by OS
    - Ex: Create processes, access files, control devices, communications
  - More on Section 2.3 (P.62-P.74)

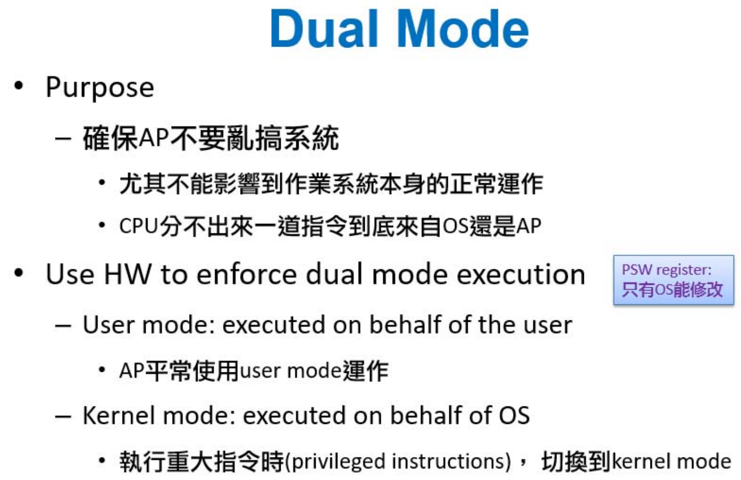
32

- A
- About the LKM assignment, which of the following is false? (a) dmesg is used to ~~delete~~ a message from the kernel ring buffer (b) insmod is used to install the kernel module (c) raw_copy_to_user() is used to copy the contents of kernel buffer to user space usr_buf (d) proc_create is used to create an entry in /proc

33

- D
- Which of the following is false: (a) privileged instructions are executed only in kernel mode (b) PSW is the register used to support dual mode operations (c) The use of VM and VMM may interfere multi-mode operations of the OS (d) The purpose of a dual mode operation is to ~~enhance the performance~~ of CPU

security



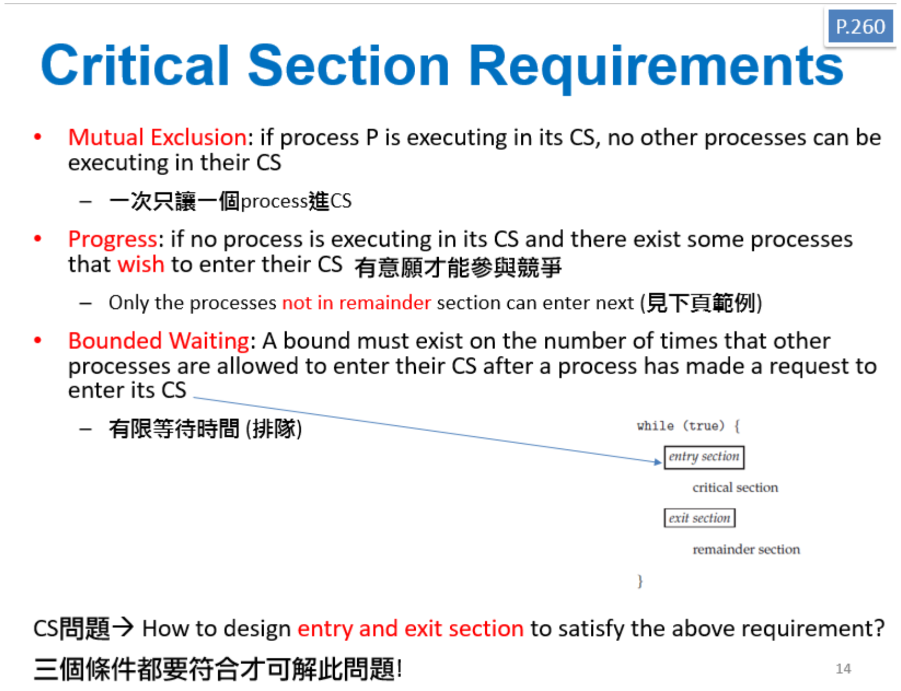
Dual Mode

- Purpose
  - 確保AP不要亂搞系統
    - 尤其不能影響到作業系統本身的正常運作
    - CPU分不出來一道指令到底來自OS還是AP
- Use HW to enforce dual mode execution
  - User mode: executed on behalf of the user
    - AP平常使用user mode運作
  - Kernel mode: executed on behalf of OS
    - 執行重大指令時(privileged instructions)，切換到kernel mode

PSW register: 只有OS能修改

34

- C
- Which of the following critical-section problem’s requirements limits the amount of time a program will wait before it can enter its critical section? (a) mutual exclusion (b) progress (c) bounded waiting (d) none of the above



P.260

Critical Section Requirements

- Mutual Exclusion: if process P is executing in its CS, no other processes can be executing in their CS
  - 一次只讓一個process進CS
- Progress: if no process is executing in its CS and there exist some processes that wish to enter their CS 有意願才能參與競爭
  - Only the processes not in remainder section can enter next (見下頁範例)
- Bounded Waiting: A bound must exist on the number of times that other processes are allowed to enter their CS after a process has made a request to enter its CS
  - 有限等待時間 (排隊)

```
while (true) {
    entry section
        critical section
    exit section
        remainder section
}
```

CS問題→ How to design entry and exit section to satisfy the above requirement?

三個條件都要符合才可解此問題!

14

Q & A