Operating System

1

圖形 使用者介面

Graphical User Interface
(GUI)

Terminal-based

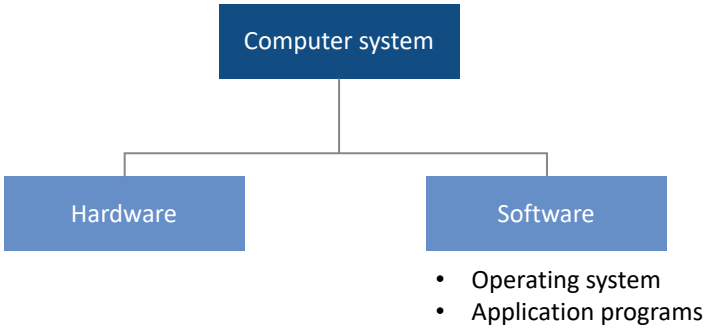
```
C:\Windows>dir
 磁碟區 C 中的磁碟沒有標籤。
 磁碟區序號:  E032-1298

 C:\Windows 的目錄

2020/10/14  上午 03:14    <DIR>          .
2020/10/14  上午 03:14    <DIR>          ..
2018/04/12  上午 07:38    <DIR>          addins
2018/10/17  下午 07:19    <DIR>          appcompat
2020/07/07  下午 02:53    <DIR>          apppatch
2020/10/14  上午 03:15    <DIR>          AppReadiness
2020/10/15  上午 03:17    <DIR>          assembly
2020/10/14  上午 03:14    <DIR>          bcastdvr
2020/09/30  下午 04:59            67,072 bfsvc.exe
2018/04/12  上午 07:38    <DIR>          Boot
2018/04/12  上午 07:38    <DIR>          Branding
2020/10/14  上午 02:50    <DIR>          CbsTemp
2019/08/29  下午 05:32             2,052 comsetup.log
2018/10/09  下午 05:08    <DIR>          Containers
2018/10/09  下午 04:13    <DIR>          CSC
2018/04/12  上午 07:38    <DIR>          Cursors
2018/10/09  下午 04:32    <DIR>          debug
2019/08/29  下午 04:34            17,148 diagerr.xml
2018/04/12  上午 07:38    <DIR>          diagnostics
2019/08/29  下午 04:34            17,148 diagwrn.xml
2018/04/12  下午 11:52    <DIR>          DigitalLocker
2020/04/06  下午 02:55    <DIR>          Downloaded Installations
2019/08/29  下午 04:34             2,139 DtcInstall.log
2018/04/12  上午 07:33            36,540 Education.xml
2018/10/26  下午 06:11    <DIR>          en-US
2018/04/12  上午 07:33            36,580 Enterprise.xml
```

2

Operating system

Operating system: Control the access to hardware by users

Application programs: Use the computer hardware to assist or solve users' tasks



3

What is OS?

Operating system (OS) : A program which manages the complete operation of your computer or mobile device and let you interact with it

- a general manager supervises the activities of each component in the system
- a program (or a set of programs) that helps to execute other programs
- Interface between computer computer and user
- Coordinate tasks and configure devices
- Monitor performance and provide file management

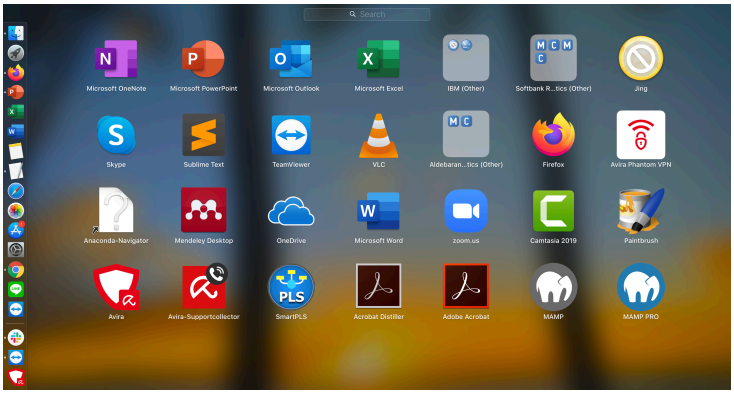
Goals : Easy to use resources and efficient to use hardware

4

Graphical User Interface

GUI (Graphical User Interface)

A form of user interface which allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation



https://en.wikipedia.org/wiki/Graphical_user_interface

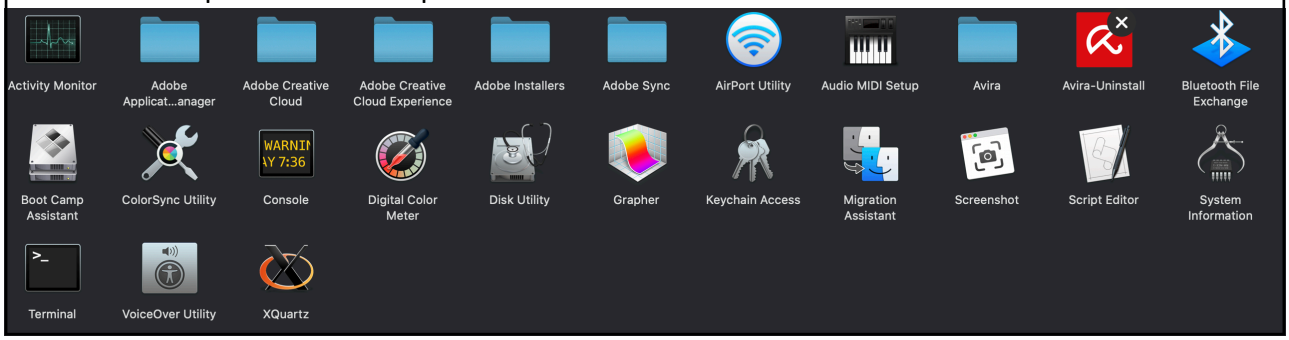
5

Utilities

實用程式

Perform maintenance-type tasks related to managing the device

- Maintain computer and devices
- Manage files
- Search for content or programs
- View images
- Install and uninstall programs and apps
- Compress and back up files



Bootstrap process

啟動程序

The OS itself needs to be loaded into the memory and run to load other programs into memory for execution

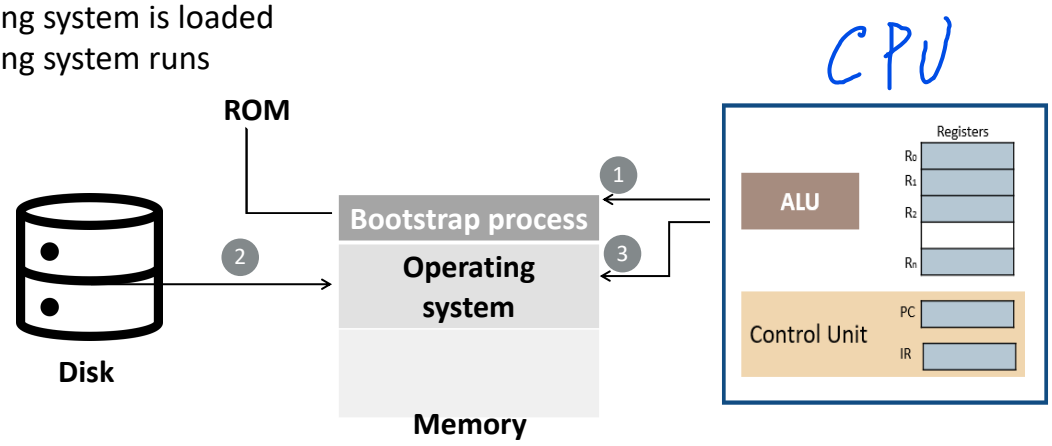
ROM holds a small program called the bootstrap program

- When the computer is turned on, the CPU counter is set to the first instruction of this bootstrap program and executes the instructions in this program
- Once finished, the program counter is set to the first instruction of the operating system in RAM

7

Bootstrap Process

1. Bootstrap program runs
2. Operating system is loaded
3. Operating system runs

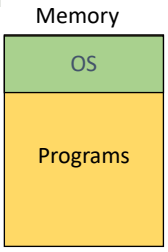


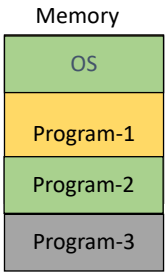
8

Memory Management

Monoprogramming (belongs to the past)

- Memory capacity is dedicated to a single program
  - Only little part is needed to hold the operating system
- When one program is running, no other program be executed
  - Speed: CPU >> Input & Output
  - CPU is idle when receiving data from or sending data to devices
- If size of program > size of memory: cannot be run





Multiprogramming (current approach)

Multiple programs are stored in memory and executed concurrently

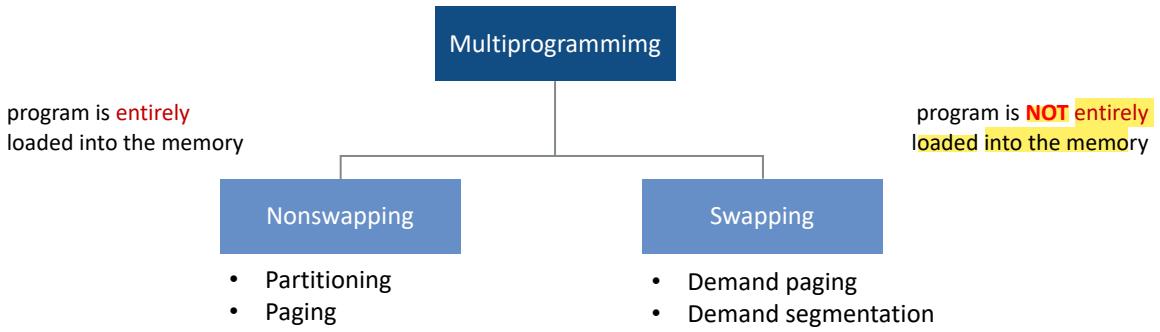
- CPU switch rapidly between programs

9

Memory Management- Multiprogramming

Nonswapping: program keeps in memory for the duration of execution

Swapping: programs can be swapped between memory and disk



10

Nonswapping- Partitioning

Memory is divided into variable-length sections

- Each partition holds one program
- CPU switches between programs
  - Execute instructions of the program, until an I/O operation is encountered or the time allocated for the program expires
- Each program is entirely loaded into the memory, requiring **contiguous** locations
  - Small partition size: programs cannot be loaded into memory
  - Large partition size: holes (unused locations) in memory
    - Memory manager can compact the partitions to remove holes and create new partitions, but it takes extra costs

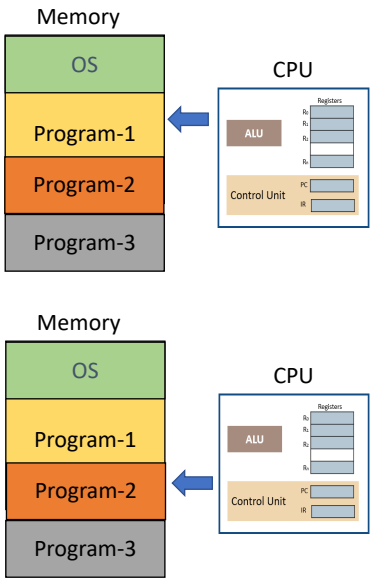


Image credit: Foundations Of Computer Science 4th edition, Behrouz Forouzan, (ISBN:1473751047)

11

Nonswapping- Paging

Programs are divided into equally sized sections: pages

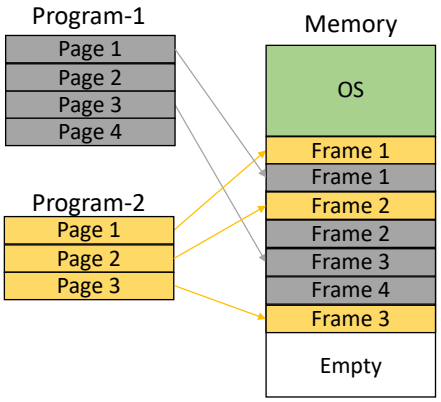
Memory is divided into equally sized sections: frames

- The size of a page/frame is the same and equal to the size of the block used by the system

Programs do not have to be contiguous in memory

- Two consecutive pages can occupy **noncontiguous** frames in memory

Paging can improve efficiency, but the entire program needs to be loaded into memory before execution



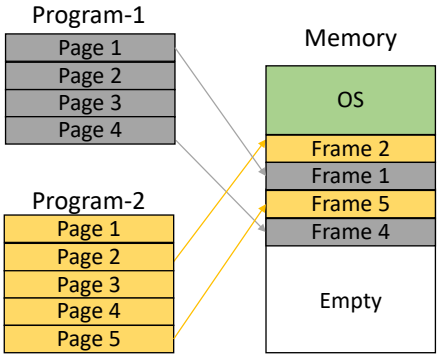
12

Swapping- Demand Paging

A program is divided into pages, and these pages can be loaded into memory one by one (not entirely), and can be executed and replaced by another page

Memory can hold pages from multiple programs at the same time

- Pages can be loaded into any free frame



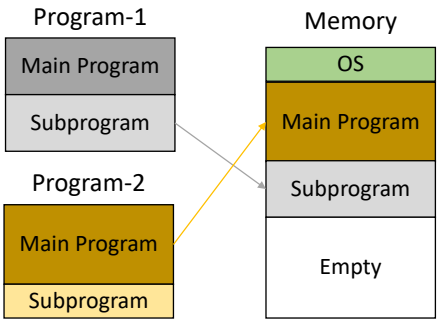
13

Swapping- Demand Segmentation

A program is usually made up of a main program and subprograms

A program is divided into multiple segments, and the segments are loaded into memory, executed and replaced by another module in the same or different program

- While segments in memory are of equal size, part of a segment may remain empty




```
Main() {
  …
  …}

Add() {
  …}

Times() {
  … }
```

14

Virtual Memory

Virtual memory means demand paging and demand segmentation

When a program is being executed, part of the program is in memory and part is on disk

- A memory size of 10 MB can execute ten programs (each: 3 MB, total: 30 MB)
- 10 MB of the ten programs are in memory and 20 MB are on disk
  - Actual memory size : 10MB; Virtual memory size: 30MB

15

OS uses three terms that refer to a set of instructions

1 Program: a nonactive set of instructions stored on disk

- A program may or may not become a job

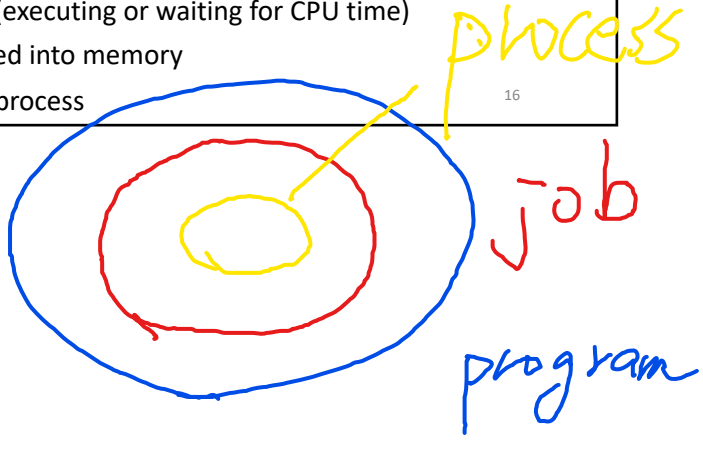
2 Job: a program becomes a job when it is selected for execution until it has finished and becomes a program again

- A job may or may not be executed
  - Located on disk waiting to be loaded to memory; loaded into memory waiting for execution by CPU; on disk or in memory waiting for input/output events
- When finished executing, a job becomes a program again
- Every job is a program, but not every program is a job

3 Process: a program in execution (has started but has not finished)

- As long as a job is in memory, it is a process (executing or waiting for CPU time)
- Selected among other waiting jobs and loaded into memory
- Every process is a job, but not every job is a process

16

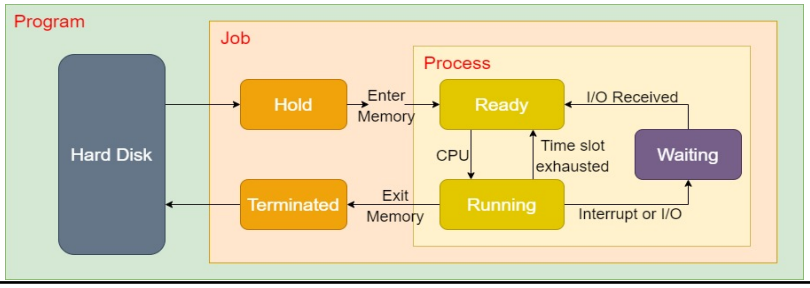


Process Manager

1. Program becomes Job when selected by OS and bring to Hold state
2. Once being loaded to memory, the Job moves to Ready state and becomes Process
3. When the CPU can execute the Job, it moves to Running state

In Running state, three things can happen:

- Process execution until I/O are needed→ move to Waiting state until I/O is finished
- Process exhausts its allocated time slot → move to Ready state
- Process terminates → move to Terminated state
- Process can move b/t the Running, Waiting and Ready states many times



17

Schedulers

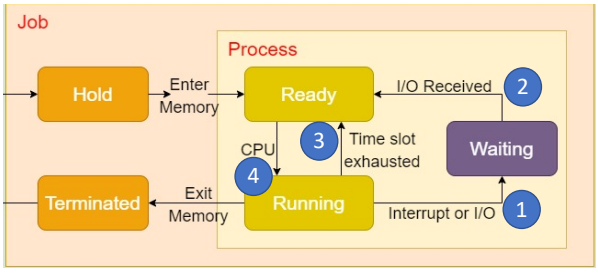
Move a Job or Process from one state to another

Job scheduler

- Create Process from Job: move Job from Hold to Ready state
- Terminate a process: move Job from Running to Terminated state

Process scheduler

1. Move a process from Running to Waiting state
   - When the process is waiting for some (I/O) events
2. Move a process from Waiting to Ready state
   - When the event is satisfied
3. Move a process from Running to Ready state
   - When the process' time allotment has expired
4. Move a process from Ready to Running state
   - When the CPU is ready to run the process



Queuing

Process manager uses queues (waiting lists) to store information

- A block of memory that stores information about jobs or processes
- Process manager stores the job or process control block instead of the job or the process itself (representing the job or process that is waiting) in the queue

An OS can have several queues

- Job queue: hold jobs that are waiting for memory
- Ready queue: hold processes (in memory, ready to be run, waiting for CPU)
- I/O queue: hold processes waiting for I/O device(s)

How to select the next job

- FIFO (first in first out)
- LIFO (last in first out)
- Shortest length first
- Highest priority first

Input and Output

19

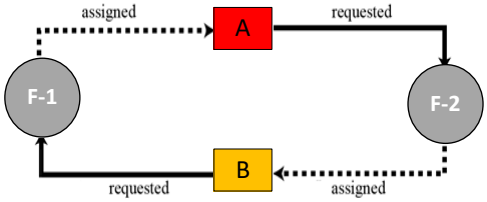
Deadlock I

Deadlock occurs when the OS fails to put resource restrictions on processes

- If the OS allows the process to start running without first checking whether the required resources are ready
  - To avoid: cannot start running until the required resources are free
- If the OS allows the process to reserve resources as needed without restrictions
  - To avoid: limit the time a process can hold a resource

When resources are accessed by multiple users

- File-1 is assigned to process-A and cannot release until it acquires File-2
- File-2 is assigned to process-B and cannot release until it acquires File-1



20

Deadlock II

The following four conditions are necessary for deadlock to occur

- Mutual exclusion: only one process can hold a resource (cannot be shared by multiple processes)
- Hold and wait: the process owns a resource, even if it cannot use it before other resources are available (still waiting for resources)
- No preemption: OS cannot temporarily reallocate a resource
- Circular waiting: all processes and resources involved form a loop

21

Starvation

Starvation is the opposite of deadlock

- When OS puts too many resource restrictions on a process

Process-A needs file-1 & file-2

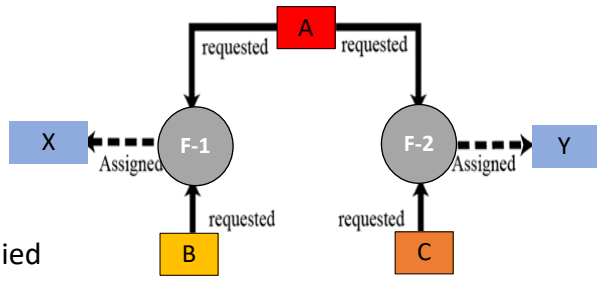
File-1 is being used by process-X

File-2 is being used by process-Y

Process-X terminates and release File-1

- Process-A cannot start as File-2 is still occupied

Process-Y needs only File-1 and is allowed to run



22

Device Manager

Device manager is responsible for access to input/output devices

- Monitor every input/output device constantly to ensure that the device is functioning properly
- When a device is ready to serve the next process in the queue
  - Device manager can maintain one or more queues for input/output devices
    - Two queues for two printers
- Control different policies for accessing input/output devices
  - FIFO for one device and LIFO for another

23

Components of OS

Shell is a user interface for access to an OS's services

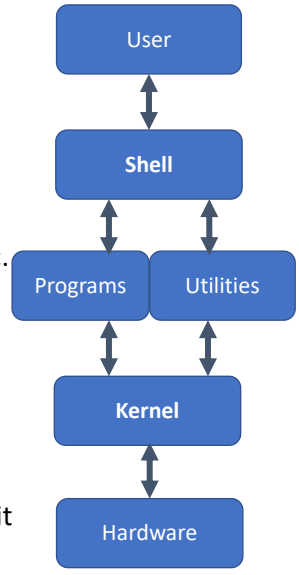
- User give shell a command
- Command-line interface or graphical user interface GUI
- OS can have several different shells

Utilities provide a support process for users

- Common utilities: text editors, search programs, sort programs, etc.

Kernel is the heart of an OS with complete control over everything in the system

- Contain the most basic parts of OS
  - Memory management, process management, device/file management
- Other components of the system call on the kernel to perform services
- If the command requires an application, the shell requests kernel to run it



24

Mobile Operating System

- Android
- iOS
- Focused on the needs of a mobile user and the capabilities of the device
- Work especially well with mobile device features
  - Touchscreens, voice recognition, wi-fi networks
  - Video and photo cameras, media players. Speech recognition, GPS, wireless capabilities, rotating screen displays (switch orientation)
- Run with limited memory of mobile devices and the display works with smaller screen size

25

Software as a Service

SaaS (Software as a Service)

- Instead of releasing a new complete version of the program, software that is distributed on a monthly subscription or an annual fee
  - Including problem fixes or other functions
- Updates: provide other features or solve any security or other issues
  - Patches: updates that address a single issue
- Upgrades: a new version of the program or application may require additional payment to install the upgrade

26

Closed source vs. Open source

Closed source: keep all or some code hidden

- Programs have standard features and can only be customized using the operating system tools
- Windows and macOS

Open source: the copyright holder has no restrictions on modification and redistribution

- Users can add features and sell their versions or give away to others
- Linux
- Security issues : Unscrupulous programmers added malicious code, which may damage the user's system or be used to collect data without the user's knowledge

27

Open source

- **Shareware**: free copyright and distribution during the trial period, after that, you must pay to continue using the program
- **Free software**: protected by copyright and provided for free, but the developer retains all rights to the product
- **Open source**: provide use, modification and redistribution without restriction of copyright owner
  - Users can add features and sell their versions or give away to others

28

Intellectual Property

- **Intellectual property** (IP) refers to ***unique and original works***
  - Ideas, inventions, art, writings, processes, product names, logos, etc.
- **Copyright** is the exclusive legal right to protect any form of tangible expression
- **Digital Rights Management** (DRM) is a strategy designed to prevent illegal distribution of movies, music and other digital content
- The license agreement determines the number of devices on which the product can be installed, any expiration dates or other restrictions
- **Pricy** is common in software
  - Illegal copy of software, movie, music, and other digital materials

29

Arithmetic operations

All arithmetic operations such as addition, subtraction, multiplication, and division can be applied to integers

A + B

A − B → A + ($\overline{B}$ + 1)

5 – 3 → 5 + (-3)

(a) 3: 0011

(b) Transfer to two's complement: 1100 → 1101

(c) 5: 0101

-3: 1101

2: 0010

30

Example A+B

A=17, B=22; A+B= ?

A = $(00010001)_2$ B = $(00010110)_2$

	0	0	0	1	0	0	0	1	A
+	0	0	0	1	0	1	1	0	B
	0	0	1	0	0	1	1	1	

31

Example A+(-B)

A=24, B=-17; A+(-B)= ?

A = $(00011000)_2$ B = $(11101111)_2$

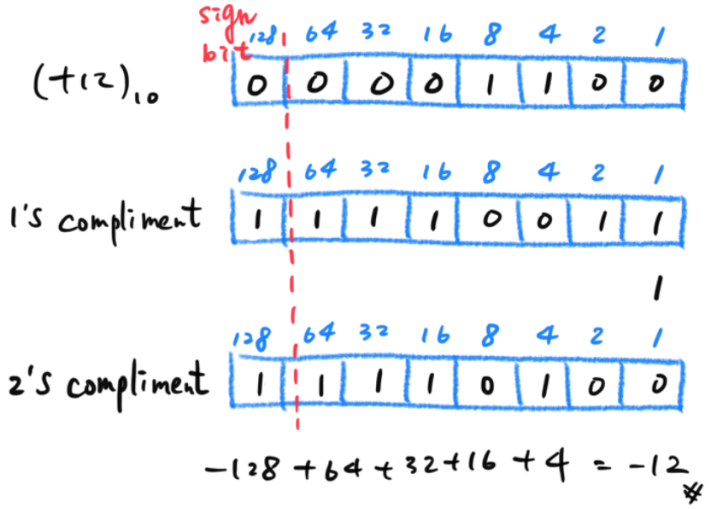
17= $(00010001)_2$

	0	0	0	1	1	0	0	0	A
+	1	1	1	0	1	1	1	1	B
	0	0	0	0	0	1	1	1	

32

2's complement

$(-12)_{10}$: ?



33

12 - 7

$(12)_{10}$ 0 0 0 0 1 1 0 0

$-(7)_{10}$ $-$ 0 0 0 0 0 1 1 1

$(5)_{10}$ 0 0 0 0 0 1 0 1

(+7) 0 0 0 0 0 1 1 1

↓

(-7) 1 1 1 1 1 0 0 1

-128+64+32+16+8+1 = -7

$(12)_{10}$ 0 0 0 0 1 1 0 0

$+(-7)_{10}$ + 1 1 1 1 1 0 0 1

$(5)_{10}$ (1) 0 0 0 0 0 1 0 1

ignore overflow value

7 - 12

8 4 2 1

$(7)_{10}$ 0 0 0 0 0 1 1 1

$-(12)_{10}$ $-$ 0 0 0 0 1 1 0 0

$(-5)_{10}$ (1) 0 1 1

-8+2+1 = -5

$(12)_{10}$ 0 0 0 0 1 1 0 0

↓

$(-12)_{10}$ 1 1 1 1 0 0 1 1 + 1

1 1 1 1 0 1 0 0

128 64 32 16 8 4 2 1

$(7)_{10}$ 0 0 0 0 0 1 1 1

$+(-12)_{10}$ + 1 1 1 1 0 1 0 0

$(-5)_{10}$ 1 1 1 1 1 0 1 1

-128+64+32+16+8+2+1 = -5

Eight-bit two's complement

Binary value	Two's complement interpretation	Unsigned interpretation
00000000	0	0
00000001	1	1
⋮	⋮	⋮
01111110	126	126
01111111	127	127
10000000	−128	128
10000001	−127	129
10000010	−126	130
⋮	⋮	⋮
11111110	−2	254
11111111	−1	255

https://en.wikipedia.org/wiki/Signed_number_representations

35