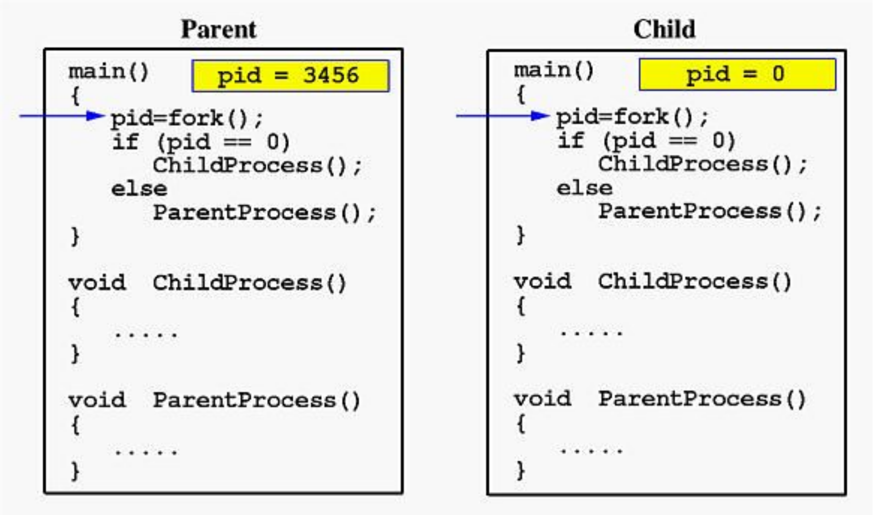
OS HW1

fork + shared memory

Part1. fork exercise (20%)

- System call fork() is used to create processes.
- It takes no arguments and returns a process ID.
    - It returns a positive value to parent process.
    - It returns a zero to child process.
    - It returns a negative value if the creation of a child process was fail.



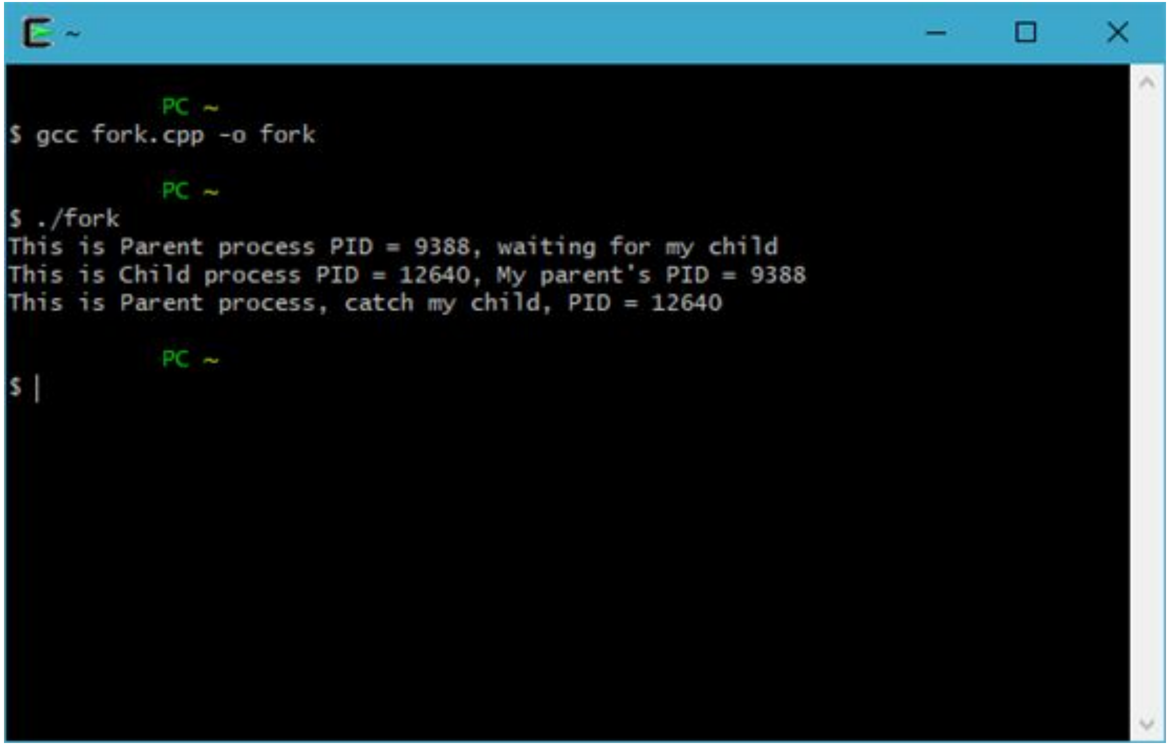
```
main()
{
    pid=fork();
    if (pid == 0)
        ChildProcess();
    else
        ParentProcess();
}

void  ChildProcess()
{
    .....
}

void  ParentProcess()
{
    .....
}
```

```
main()
{
    pid=fork();
    if (pid == 0)
        ChildProcess();
    else
        ParentProcess();
}

void  ChildProcess()
{
    .....
}

void  ParentProcess()
{
    .....
}
```

Requirements

- use fork() to create processes
- parent process
  - print parent process id and wait children process
  - when children process done, print children process id
- children process
  - print children process id and parent porcess id
- upload your code in New e3 named studentID_hw1-2.cpp /.c
- violating any requirement above will get score penalty

example output

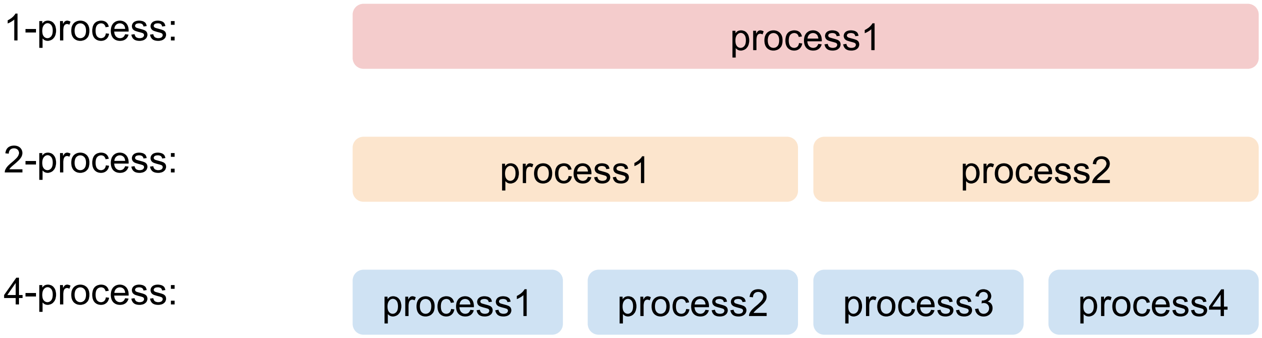


```
          PC ~
$ gcc fork.cpp -o fork

          PC ~
$ ./fork
This is Parent process PID = 9388, waiting for my child
This is Child process PID = 12640, My parent's PID = 9388
This is Parent process, catch my child, PID = 12640

          PC ~
$ |
```

Part2. calculate determinant (80%)

using multi-process to calculate determinant

$$\begin{vmatrix} 2 & -5 & 4 & 1 \\ 1 & 0 & 3 & -2 \\ -4 & 5 & 3 & 0 \\ -2 & 1 & 1 & 2 \end{vmatrix} = 2\begin{vmatrix} 0 & 3 & -2 \\ 5 & 3 & 0 \\ 1 & 1 & 2 \end{vmatrix} - (-5)\begin{vmatrix} 1 & 3 & -2 \\ -4 & 3 & 0 \\ -2 & 1 & 2 \end{vmatrix} + 4\begin{vmatrix} 1 & 0 & -2 \\ -4 & 5 & 0 \\ -2 & 1 & 2 \end{vmatrix} - 1\begin{vmatrix} 1 & 0 & 3 \\ -4 & 5 & 3 \\ -2 & 1 & 1 \end{vmatrix}$$



1-process:

process1

2-process:

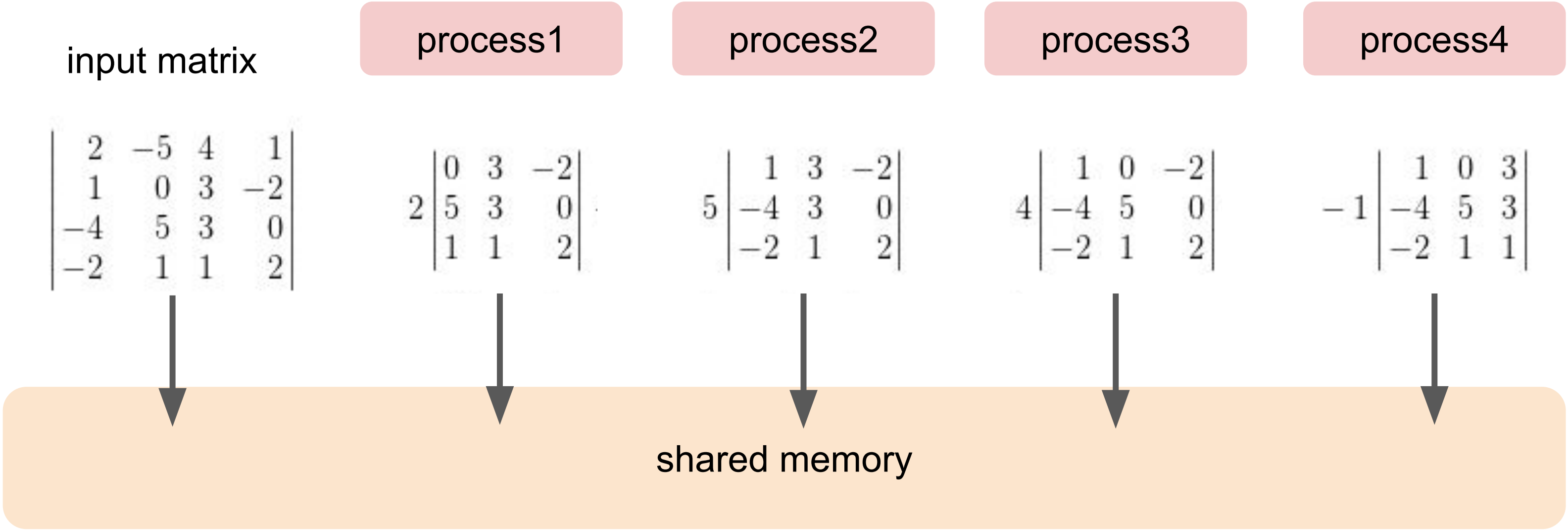
process1 process2

4-process:

process1 process2 process3 process4

Part2. calculate determinant

store the input matrix and the multi-process result in shared memory



input matrix

$$\begin{vmatrix} 2 & -5 & 4 & 1 \\ 1 & 0 & 3 & -2 \\ -4 & 5 & 3 & 0 \\ -2 & 1 & 1 & 2 \end{vmatrix}$$

process1

$$2\begin{vmatrix} 0 & 3 & -2 \\ 5 & 3 & 0 \\ 1 & 1 & 2 \end{vmatrix}$$

process2

$$5\begin{vmatrix} 1 & 3 & -2 \\ -4 & 3 & 0 \\ -2 & 1 & 2 \end{vmatrix}$$

process3

$$4\begin{vmatrix} 1 & 0 & -2 \\ -4 & 5 & 0 \\ -2 & 1 & 2 \end{vmatrix}$$

process4

$$-1\begin{vmatrix} 1 & 0 & 3 \\ -4 & 5 & 3 \\ -2 & 1 & 1 \end{vmatrix}$$

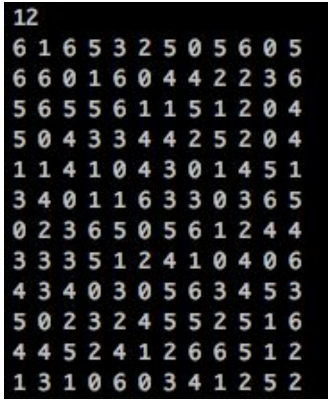
shared memory

Requirements

- read the file input.txt to get dimension and matrix
- create multi-process using fork()
  - 4 cases, process parallelism from 1 to 4
  - store input matrix and each process result in shared memory
- for each case, print elapsed time and determinant
  - the final determinant must be correct
  - matrix elements and the determinant are 32-bit unsigned int
  - do not care about integer overflows
- TAs will test your program using any matrix dimension between 10*10 and 15*15
  - your 2-process and 3-process must be noticeably faster than 1-process
- upload your code in New e3 named studentID_hw1-2.cpp /.c
- violating any requirement above will get score penalty

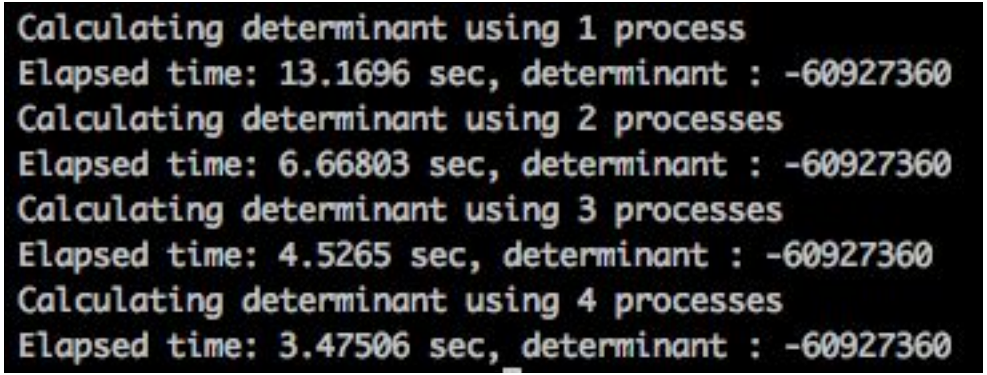
example

input.txt:



```
12
6 1 6 5 3 2 5 0 5 6 0 5
6 6 0 1 6 0 4 4 2 2 3 6
5 6 5 5 6 1 1 5 1 2 0 4
5 0 4 3 3 4 4 2 5 2 0 4
1 1 4 1 0 4 3 0 1 4 5 1
3 4 0 1 1 6 3 3 0 3 6 5
0 2 3 6 5 0 5 6 1 2 4 4
3 3 3 5 1 2 4 1 0 4 0 6
4 3 4 0 3 0 5 6 3 4 5 3
5 0 2 3 2 4 5 5 2 5 1 6
4 4 5 2 4 1 2 6 6 5 1 2
1 3 1 0 6 0 3 4 1 2 5 2
```

output:



```
Calculating determinant using 1 process
Elapsed time: 13.1696 sec, determinant : -60927360
Calculating determinant using 2 processes
Elapsed time: 6.66803 sec, determinant : -60927360
Calculating determinant using 3 processes
Elapsed time: 4.5265 sec, determinant : -60927360
Calculating determinant using 4 processes
Elapsed time: 3.47506 sec, determinant : -60927360
```

number in the first row is matrix dimension
the rest rows are the matrix rows

header files

- unistd.h
- sys/ipc.h
- sys/shm.h
- sys/wait.h
- sys/time.h

APIs

- fork() - creates a child process by duplicating the calling process
  - http://man7.org/linux/man-pages/man2/fork.2.html
- wait() family - wait for process to change state
  - http://man7.org/linux/man-pages/man2/waitpid.2.html
- gettimeofday() - get the time when start / end calculating to compute the elapsed time
  - http://man7.org/linux/man-pages/man2/gettimeofday.2.html

APIs

- shmget() - create a block of shared memory
  - http://man7.org/linux/man-pages/man2/shmget.2.html
- shmat() - attach shared memory to the current process’s address space
  - http://man7.org/linux/man-pages/man2/shmat.2.html
- shmdt() - detach shared memory from the current process’s address space
  - http://man7.org/linux/man-pages/man2/shmdt.2.html
- shmctl() - control shared memory
  - http://man7.org/linux/man-pages/man2/shmctl.2.html

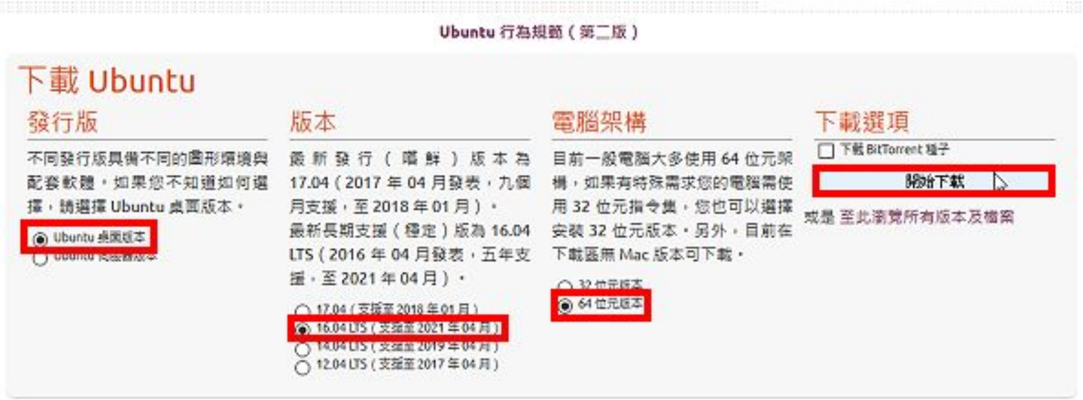
testing environment

- ubuntu 16.04
- ubuntu 14.04
- CS linux work station

your code should compile successfully in one of the above environments!

virtualbox

ubuntu download: http://www.ubuntu-tw.org/modules/tinyd0/



Ubuntu 行為規範（第二版）

下載 Ubuntu

發行版

不同發行版具備不同的圖形環境與配套軟體，如果您不知道如何選擇，請選擇 Ubuntu 桌面版本。

● Ubuntu 桌面版本

○ Ubuntu 伺服器版本

版本

最新發行（嚐鮮）版本為 17.04（2017 年 04 月發表，九個月支援，至 2018 年 01 月）。最新長期支援（穩定）版為 16.04 LTS（2016 年 04 月發表，五年支援，至 2021 年 04 月）。

○ 17.04（支援至 2018 年 01 月）

● 16.04 LTS（支援至 2021 年 04 月）

○ 14.04 LTS（支援至 2019 年 04 月）

○ 12.04 LTS（支援至 2017 年 04 月）

電腦架構

目前一般電腦大多使用 64 位元架構，如果有特殊需求您的電腦需使用 32 位元指令集，您也可以選擇安裝 32 位元版本。另外，目前在下載區無 Mac 版本可下載。

○ 32 位元版本

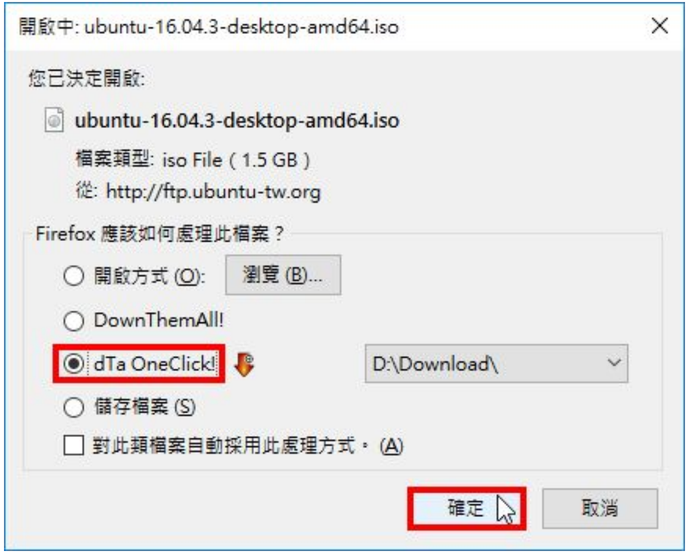
● 64 位元版本

下載選項

□ 下載 BitTorrent 種子

開始下載

或是 至此瀏覽所有版本及檔案



開啟中: ubuntu-16.04.3-desktop-amd64.iso

您已決定開啟:

ubuntu-16.04.3-desktop-amd64.iso

檔案類型: iso File (1.5 GB)

從: http://ftp.ubuntu-tw.org

Firefox 應該如何處理此檔案？

○ 開啟方式 (O): 瀏覽 (B)...

○ DownThemAll!

● dTa OneClick! D:\Download\

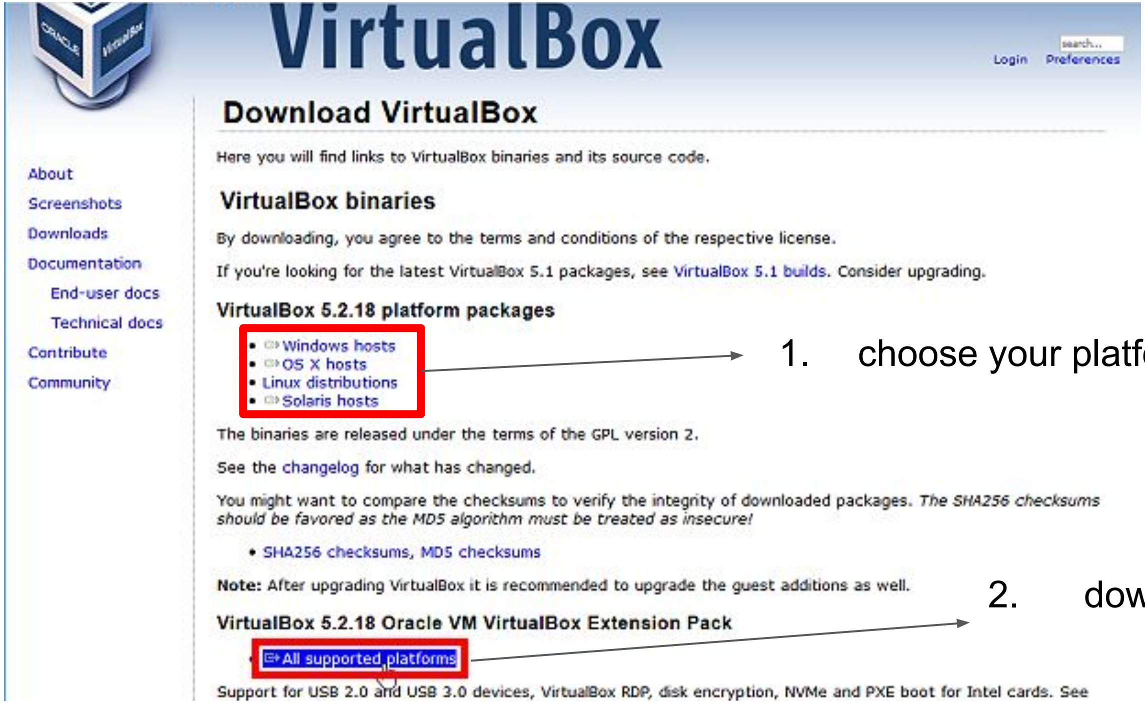
○ 儲存檔案 (S)

□ 對此類檔案自動採用此處理方式。 (A)

確定 取消

virtualbox

virtualbox download : https://www.virtualbox.org/wiki/Downloads



Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 5.1 packages, see VirtualBox 5.1 builds. Consider upgrading.

VirtualBox 5.2.18 platform packages

- Windows hosts
- OS X hosts
- Linux distributions
- Solaris hosts

The binaries are released under the terms of the GPL version 2.

See the changelog for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

- SHA256 checksums, MD5 checksums

Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

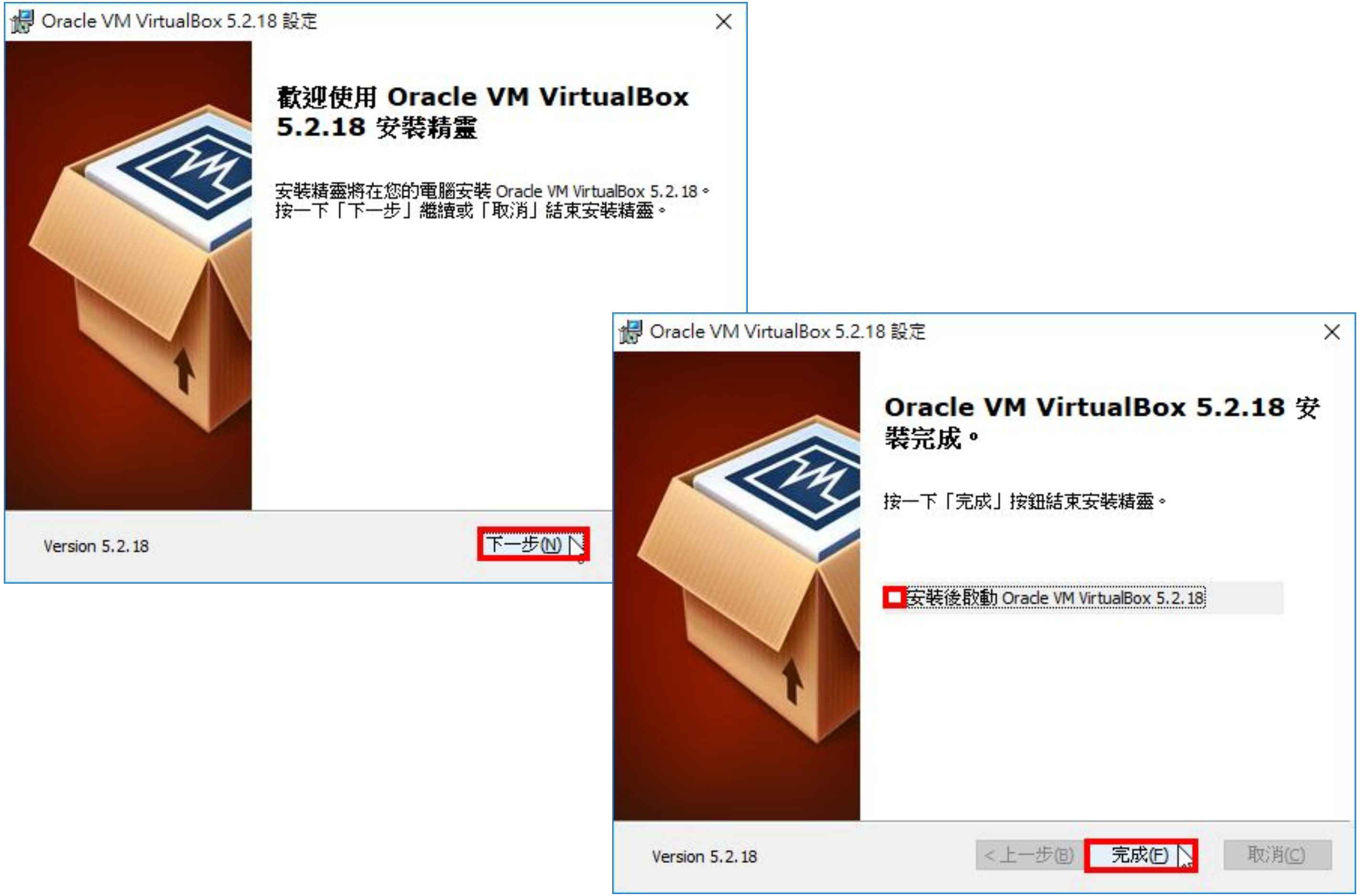
VirtualBox 5.2.18 Oracle VM VirtualBox Extension Pack

- All supported platforms

Support for USB 2.0 and USB 3.0 devices, VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See

1. choose your platform to download packages
2. download

virtualbox



Oracle VM VirtualBox 5.2.18 設定

歡迎使用 Oracle VM VirtualBox 5.2.18 安裝精靈

安裝精靈將在您的電腦安裝 Oracle VM VirtualBox 5.2.18。
按一下「下一步」繼續或「取消」結束安裝精靈。

Version 5.2.18

Oracle VM VirtualBox 5.2.18 設定

Oracle VM VirtualBox 5.2.18 安裝完成。

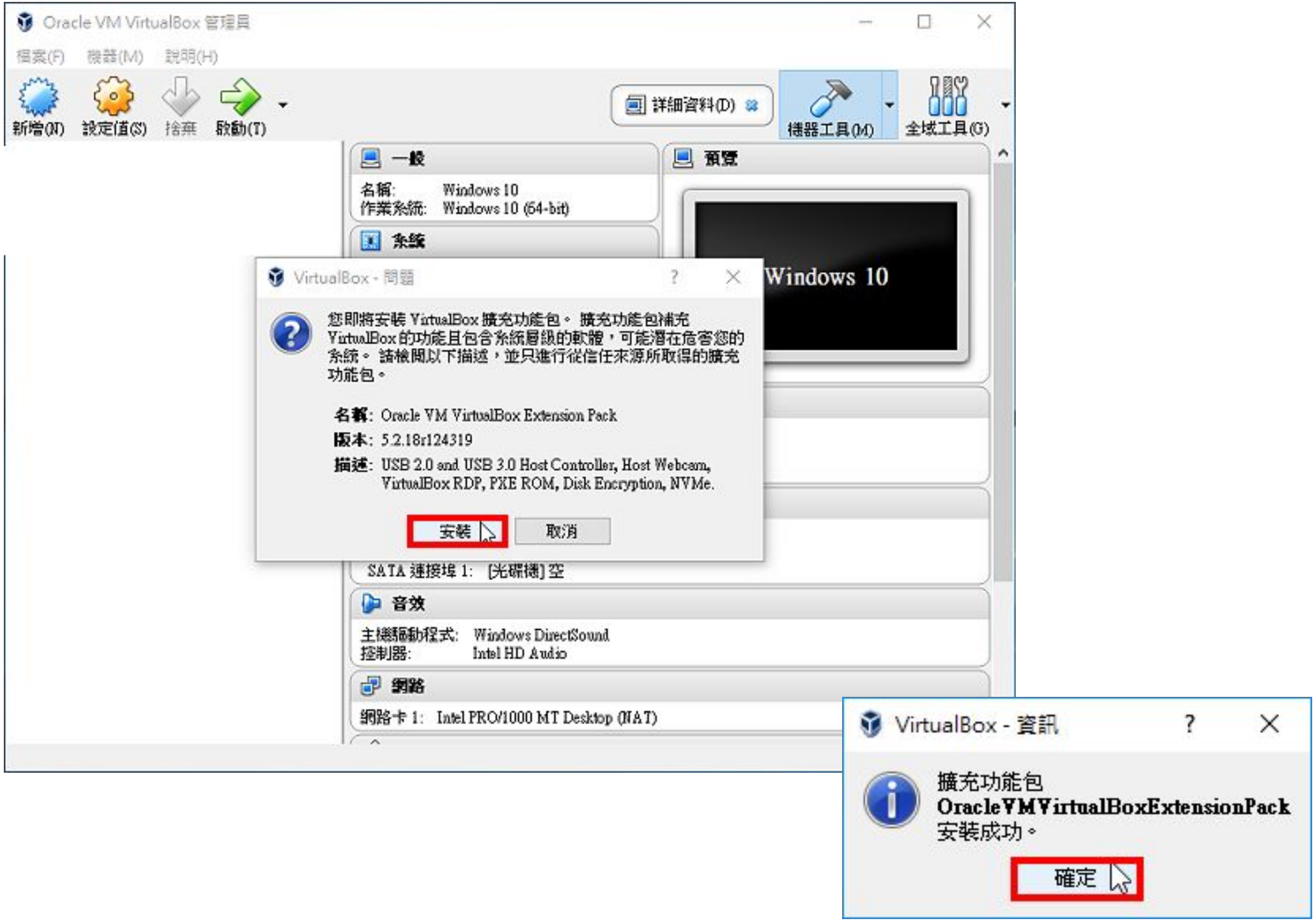
按一下「完成」按鈕結束安裝精靈。

□ 安裝後啟動 Oracle VM VirtualBox 5.2.18

Version 5.2.18

virtualbox

Oracle_VM_VirtualBox_Extension_Pack-5.2.18.vbox-extpack



名稱: Windows 10
作業系統: Windows 10 (64-bit)

SATA 連接埠 1: [光碟機] 空

主機驅動程式: Windows DirectSound
控制器: Intel HD Audio

網路卡 1: Intel PRO/1000 MT Desktop (NAT)

VirtualBox - 問題

您即將安裝 VirtualBox 擴充功能包。 擴充功能包補充 VirtualBox 的功能且包含系統層級的軟體，可能潛在危害您的系統。 請檢閱以下描述，並只進行從信任來源所取得的擴充功能包。

名稱: Oracle VM VirtualBox Extension Pack

版本: 5.2.18r124319

描述: USB 2.0 and USB 3.0 Host Controller, Host Webcam, VirtualBox RDP, PXE ROM, Disk Encryption, NVMe.

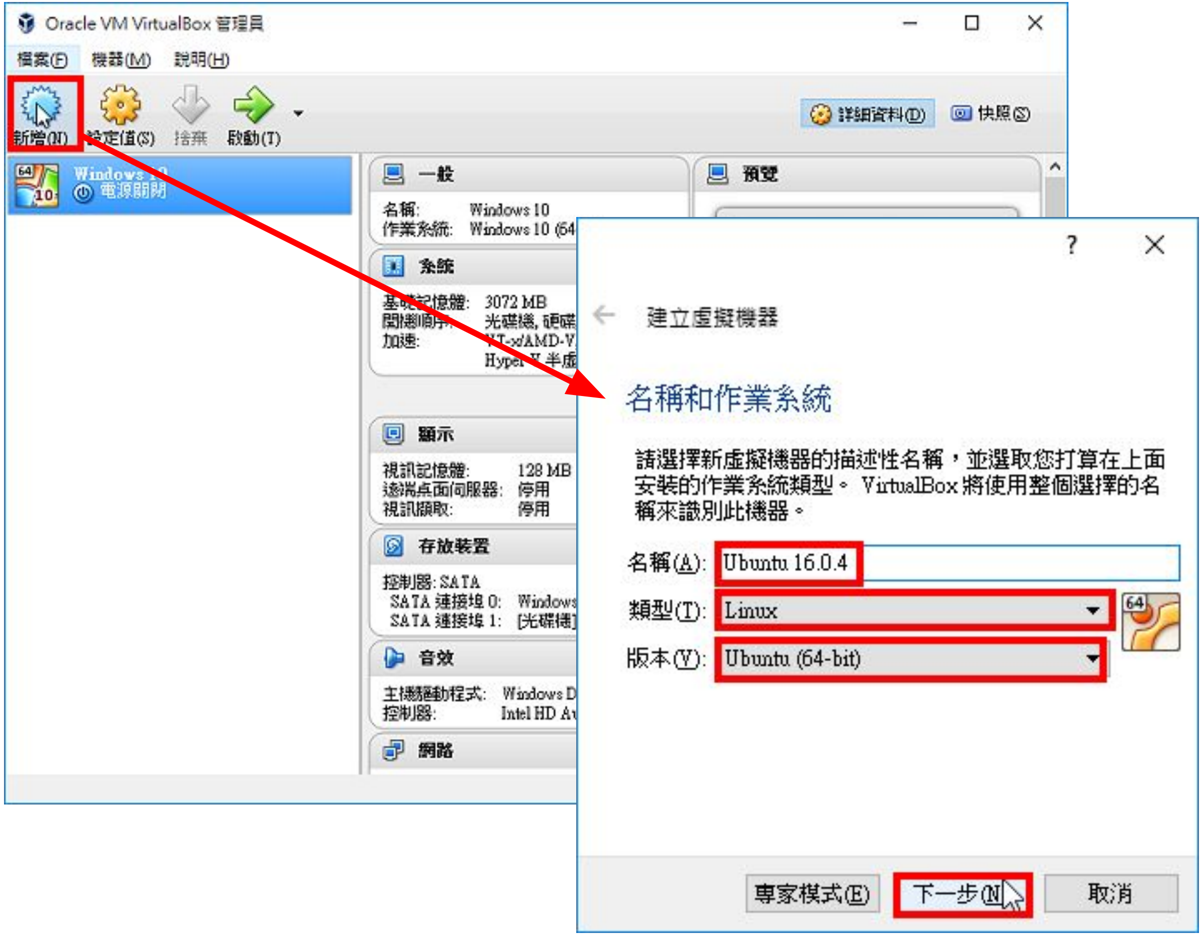
安裝 取消

VirtualBox - 資訊

擴充功能包
OracleVMVirtualBoxExtensionPack
安裝成功。

確定

virtualbox



名稱: Windows 10
作業系統: Windows 10 (64

基本記憶體: 3072 MB
開機順序: 光碟機, 硬碟
加速: VT-x/AMD-V
Hyper-V 半虛

視訊記憶體: 128 MB
遠端桌面伺服器: 停用
視訊擷取: 停用

控制器: SATA
SATA 連接埠 0: Windows
SATA 連接埠 1: [光碟機]

主機驅動程式: Windows D
控制器: Intel HD Au

建立虛擬機器

名稱和作業系統

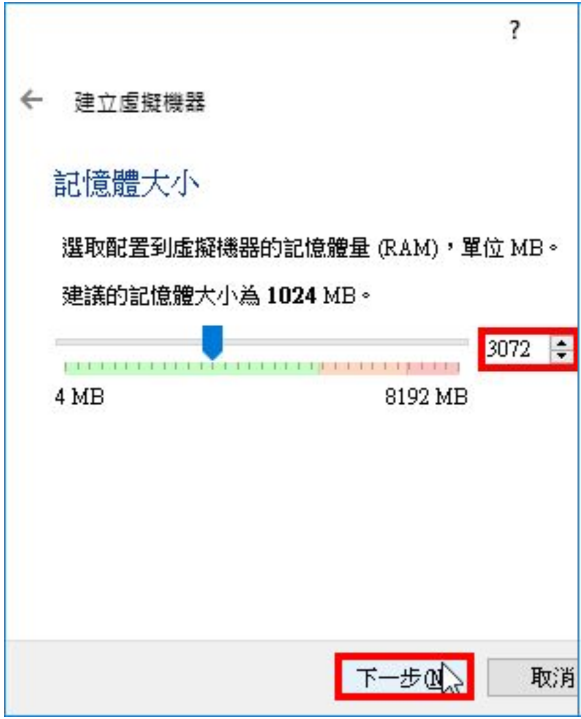
請選擇新虛擬機器的描述性名稱，並選取您打算在上面安裝的作業系統類型。 VirtualBox 將使用整個選擇的名稱來識別此機器。

名稱(A): Ubuntu 16.0.4

類型(T): Linux

版本(V): Ubuntu (64-bit)

virtualbox



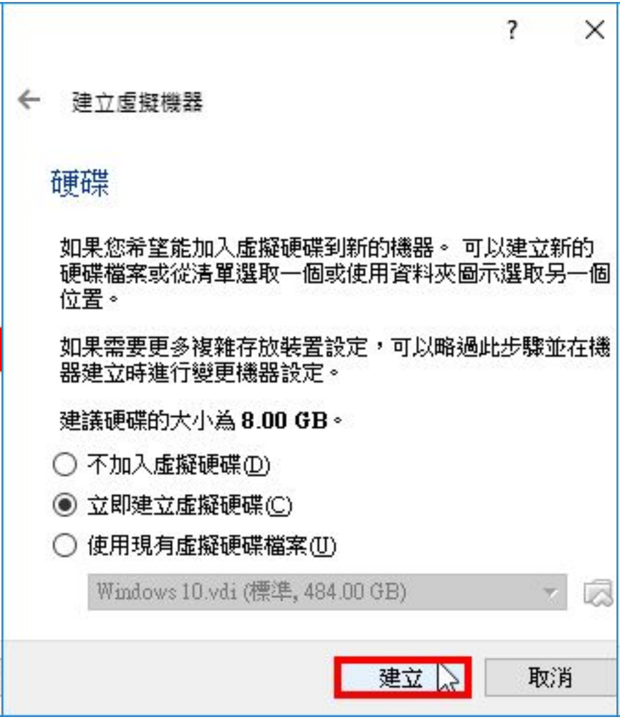
記憶體大小

選取配置到虛擬機器的記憶體量 (RAM)，單位 MB。

建議的記憶體大小為 **1024** MB。

3072

4 MB 8192 MB



硬碟

如果您希望能加入虛擬硬碟到新的機器。 可以建立新的硬碟檔案或從清單選取一個或使用資料夾圖示選取另一個位置。

如果需要更多複雜存放裝置設定，可以略過此步驟並在機器建立時進行變更機器設定。

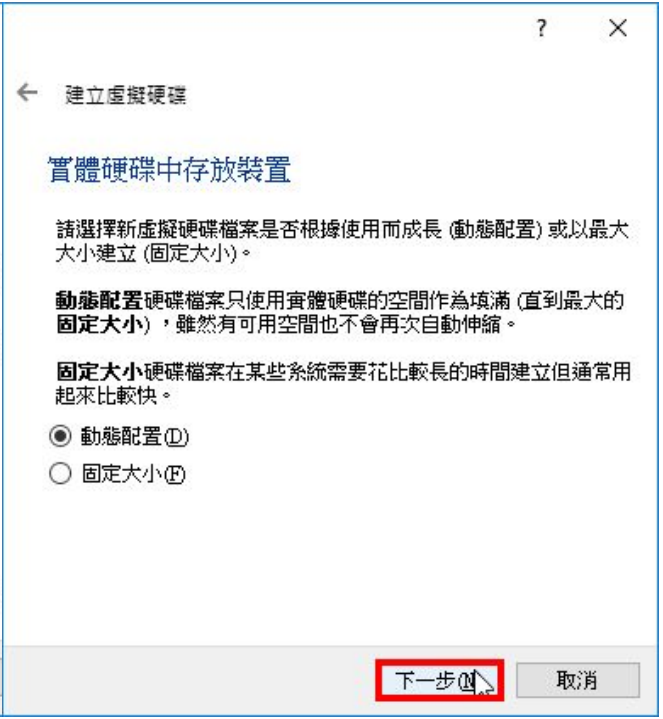
建議硬碟的大小為 **8.00 GB**。

○ 不加入虛擬硬碟(D)

● 立即建立虛擬硬碟(C)

○ 使用現有虛擬硬碟檔案(U)

Windows 10.vdi (標準, 484.00 GB)



實體硬碟中存放裝置

請選擇新虛擬硬碟檔案是否根據使用而成長 (動態配置) 或以最大大小建立 (固定大小)。

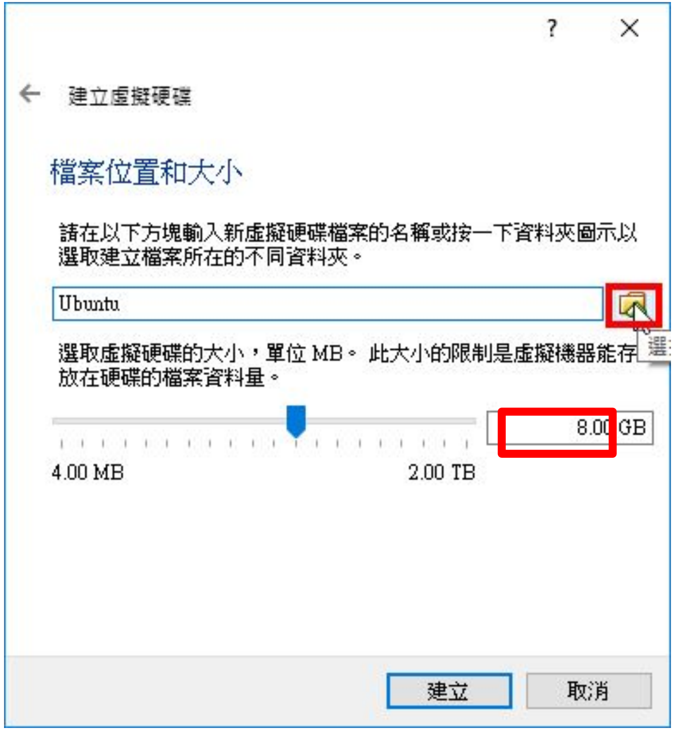
動態配置硬碟檔案只使用實體硬碟的空間作為填滿 (直到最大的**固定大小**)，雖然有可用空間也不會再次自動伸縮。

固定大小硬碟檔案在某些系統需要花比較長的時間建立但通常用起來比較快。

● 動態配置(D)

○ 固定大小(F)

virtualbox



建立虛擬硬碟

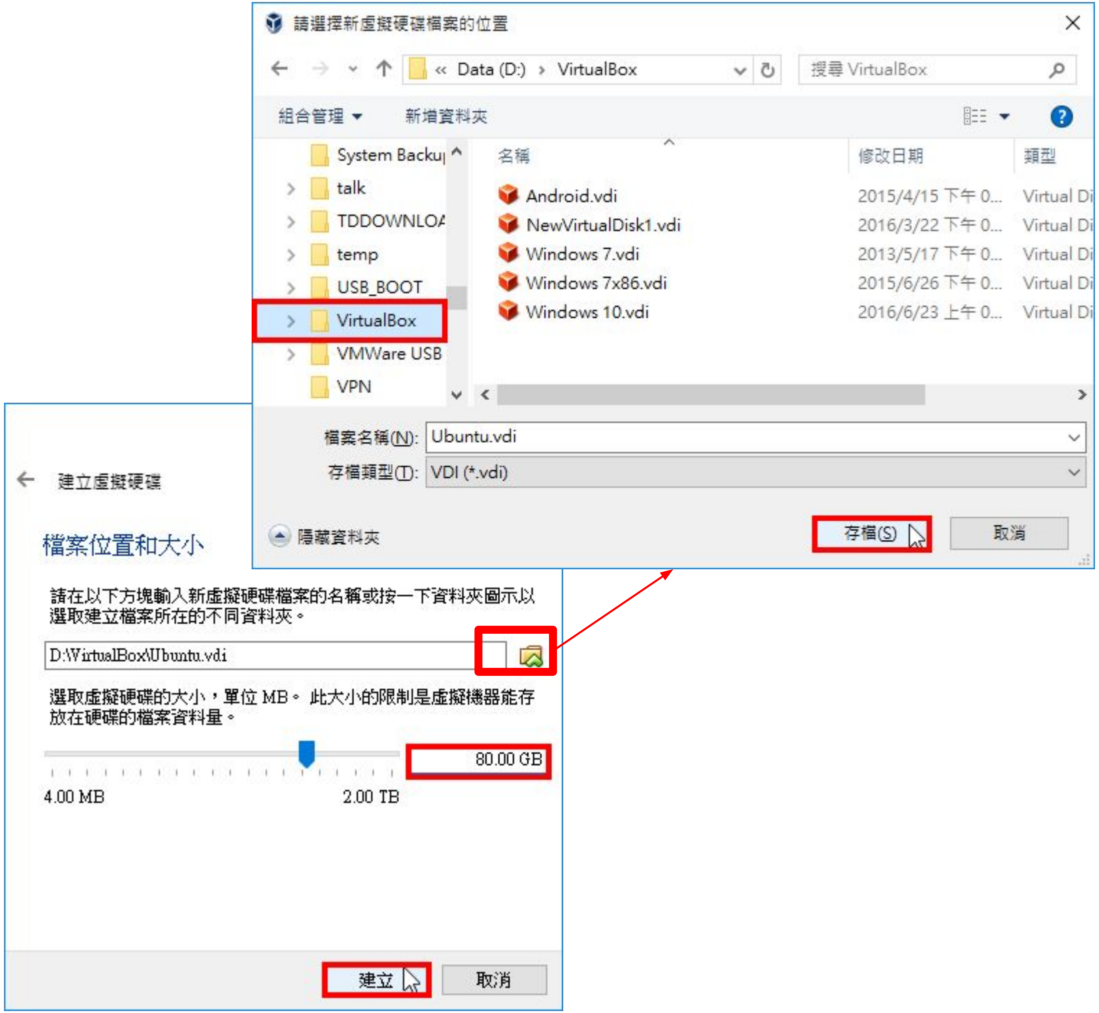
檔案位置和大小

請在以下方塊輸入新虛擬硬碟檔案的名稱或按一下資料夾圖示以選取建立檔案所在的不同資料夾。

Ubuntu

選取虛擬硬碟的大小，單位 MB。 此大小的限制是虛擬機器能存放在硬碟的檔案資料量。

8.00 GB



請選擇新虛擬硬碟檔案的位置

« Data (D:) › VirtualBox

名稱	修改日期	類型
Android.vdi	2015/4/15 下午 0...	Virtual Di
NewVirtualDisk1.vdi	2016/3/22 下午 0...	Virtual Di
Windows 7.vdi	2013/5/17 下午 0...	Virtual Di
Windows 7x86.vdi	2015/6/26 下午 0...	Virtual Di
Windows 10.vdi	2016/6/23 上午 0...	Virtual Di

檔案名稱(N): Ubuntu.vdi

存檔類型(T): VDI (*.vdi)

建立虛擬硬碟

檔案位置和大小

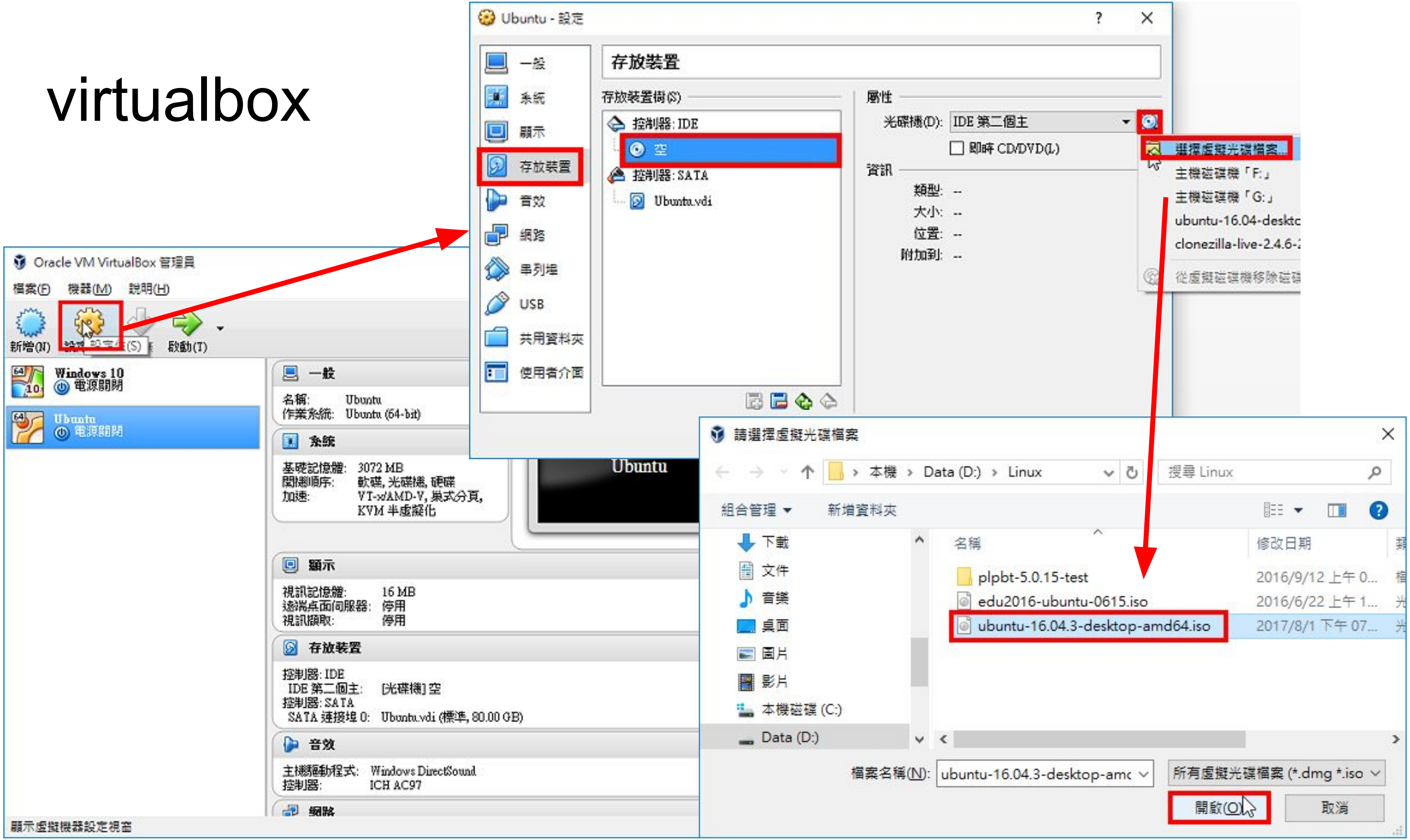
請在以下方塊輸入新虛擬硬碟檔案的名稱或按一下資料夾圖示以選取建立檔案所在的不同資料夾。

D:\VirtualBox\Ubuntu.vdi

選取虛擬硬碟的大小，單位 MB。 此大小的限制是虛擬機器能存放在硬碟的檔案資料量。

80.00 GB

virtualbox



Windows 10 電源關閉

Ubuntu 電源關閉

一般

名稱: Ubuntu
作業系統: Ubuntu (64-bit)

系統

基礎記憶體: 3072 MB
開機順序: 軟碟, 光碟機, 硬碟
加速: VT-x/AMD-V, 巢式分頁, KVM 半虛擬化

顯示

視訊記憶體: 16 MB
遠端桌面伺服器: 停用
視訊擷取: 停用

存放裝置

控制器: IDE
IDE 第二個主: [光碟機] 空
控制器: SATA
SATA 連接埠 0: Ubuntu.vdi (標準, 80.00 GB)

音效

主機驅動程式: Windows DirectSound
控制器: ICH AC97

網路

存放裝置

存放裝置樹(S)

- 控制器: IDE
  - 空
- 控制器: SATA
  - Ubuntu.vdi

屬性

光碟機(D): IDE 第二個主

□ 即時 CD/DVD(L)

資訊

類型: --
大小: --
位置: --
附加到: --

本機 › Data (D:) › Linux

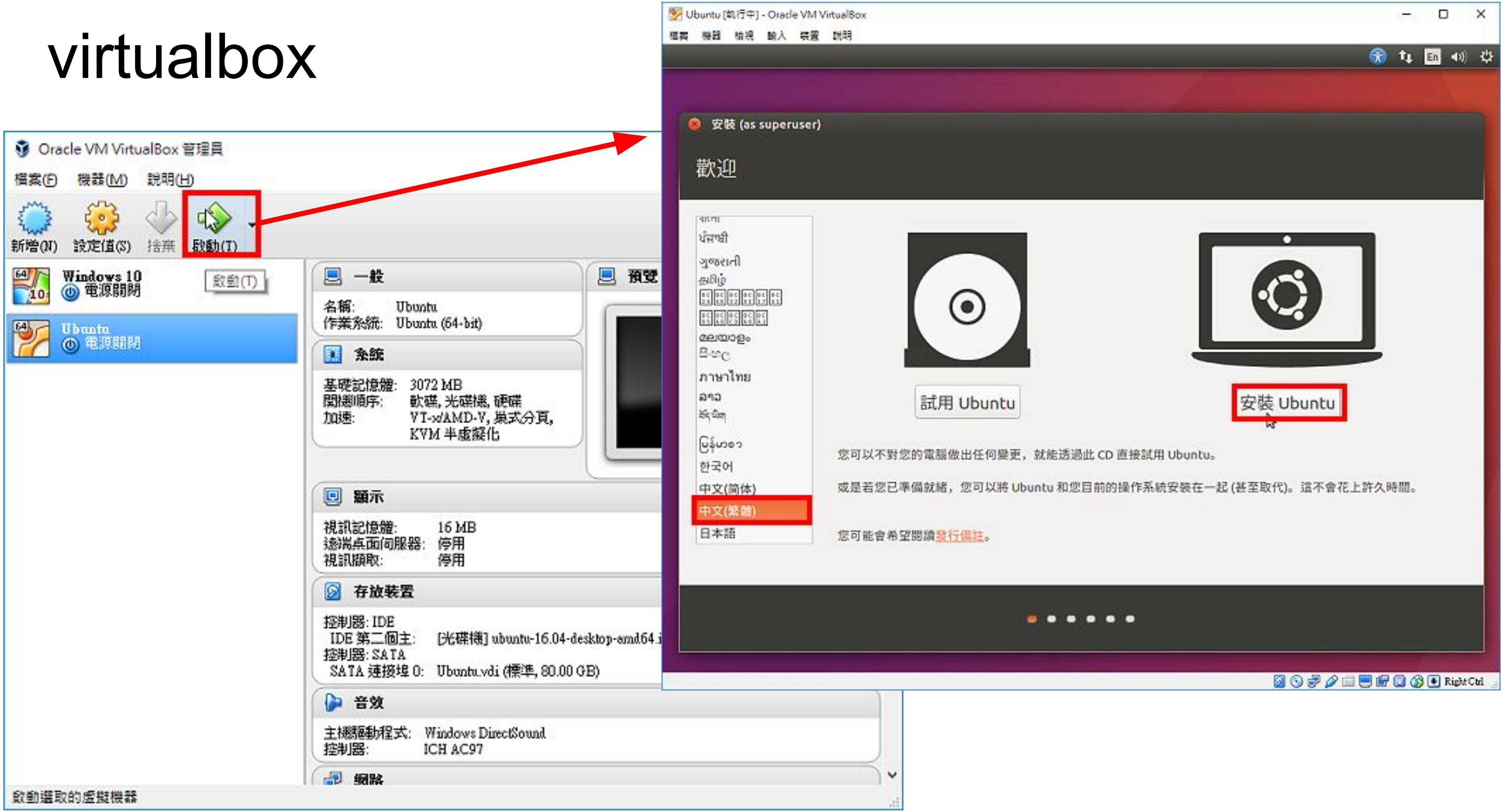
搜尋 Linux

名稱	修改日期
plpbt-5.0.15-test	2016/9/12 上午 0...
edu2016-ubuntu-0615.iso	2016/6/22 上午 1...
ubuntu-16.04.3-desktop-amd64.iso	2017/8/1 下午 07...

檔案名稱(N): ubuntu-16.04.3-desktop-amd

所有虛擬光碟檔案 (*.dmg *.iso

virtualbox



名稱: Ubuntu
作業系統: Ubuntu (64-bit)

基礎記憶體: 3072 MB
開機順序: 軟碟, 光碟機, 硬碟
加速: VT-x/AMD-V, 巢式分頁, KVM 半虛擬化

視訊記憶體: 16 MB
遠端桌面伺服器: 停用
視訊擷取: 停用

控制器: IDE
IDE 第二個主: [光碟機] ubuntu-16.04-desktop-amd64.i
控制器: SATA
SATA 連接埠 0: Ubuntu.vdi (標準, 80.00 GB)

主機驅動程式: Windows DirectSound
控制器: ICH AC97

啟動選取的虛擬機器

您可以不對您的電腦做出任何變更，就能透過此 CD 直接試用 Ubuntu。

或是若您已準備就緒，您可以將 Ubuntu 和您目前的作業系統安裝在一起 (甚至取代)。這不會花上許久時間。

您可能會希望閱讀發行備註。