

1. 下載 [VirtualBox 6.1.32 platform packages](#) [Windows hosts](#)
點擊 <https://download.virtualbox.org/virtualbox/6.1.32/VirtualBox-6.1.32-149290-Win.exe>

執行 VirtualBox-6.1.32-149290-Win.exe 安裝

2. arch linux 官網 - <https://www.archlinux.org/> <https://archlinux.org/download/> 812.3 MB
安裝硬體需求說明 https://wiki.archlinux.org/index.php/Installation_guide

Requirements for installing Arch Linux:

A x86_64 (i.e. 64 bit) compatible machine

Minimum 512 MB of RAM (recommended 2 GB)

At least 1 GB of free disk space (recommended 20 GB for basic usage)

An active internet connection

A USB drive with minimum 2 GB of storage capacity

Familiarity with Linux command line

Step 1: Download the ISO <https://www.archlinux.org/download/> **Taiwan** 區下載點 下載
http://mirror.archlinux.tw/ArchLinux/iso/2022.02.01/x86_64.iso 下載 [archlinux-2022.02.01-x86_64.iso](#)

http://archlinux.cs.nctu.edu.tw/iso/2022.02.01/x86_64.iso 下載 [archlinux-2022.02.01-x86_64.iso](#)

目前版本 **Current Release:** 2022.02.01

選擇地區下載可加快速度 Taiwan 地區

- [archlinux.tw](#) [archlinux.tw](#) [ccns.ncku.edu.tw](#) [free.nchc.org.tw](#) [free.nchc.org.tw](#)
- [nctu.edu.tw](#) [ntou.edu.tw](#) [ntou.edu.tw](#) [tku.edu.tw](#) [yzu.edu.tw](#) [yzu.edu.tw](#)

China 地區 [163.com](#) [bfsu.edu.cn](#) [bfsu.edu.cn](#) [cqu.edu.cn](#) [cqu.edu.cn](#) [dgut.edu.cn](#)
[dgut.edu.cn](#)

點擊 **新增** 建立虛擬機 設定名稱, 機器資料夾 可設你要的位置例如

c:\VirtualBoxVms

圖中為 d:\ VirtualBoxVms ← **學校電腦 要改在 c 碟**



建立 虛擬機器 **RAM 設 1G**

← 建立虛擬機器

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 **1024MB**。






名稱和作業系統

請為新的虛擬機器選擇描述性名稱和目的地資料夾，並選取其上安裝的作業系統類型。您選擇的名稱將在整個 VirtualBox 使用，以標識這部電腦。

名稱:

機器資料夾:  g:\VirtualBoxVMs

類型(T): Linux

版本(V): Arch Linux (64-bit)

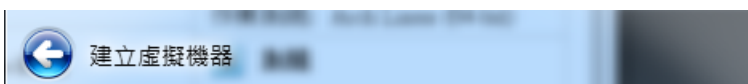


記憶體大小

選取配置到虛擬機器的記憶體體量 (RAM)，單位 MB。

建議的記憶體大小為 **1024MB**。

 1024 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)

 usb2.vmdk (標準, 不可存取) 



硬碟檔類型

請選擇新的虛擬硬碟希望使用的檔案類型。如果不需要用在其它虛擬化軟體，您可以保留這個設定不變更。

- ☒ VDI (VirtualBox 磁碟映像)
- ☐ VHD (虛擬硬碟)
- ☐ VMDK (虛擬機器磁碟)

存放裝置在實體硬碟

請選擇新的虛擬硬碟檔是否應根據使用來成長 (動態分配)，或建立為其最大大小 (固定大小)。

動態分配的硬碟檔將只在填滿時使用實體硬碟的空間 (直到最大的**固定大小**)，儘管它的空間釋放時不會再次自動縮小。

固定大小硬碟檔在某些系統需要比較長的時間建立，但通常用起來比較快。

- ☒ 動態分配(D)
☐ 固定大小(F)

檔案位置和大小

請在以下的方塊中輸入新虛擬硬碟檔的名稱，或按一下資料夾圖示以選擇建立檔案的其它資料夾。

G:\VirtualBoxVMs\arch\arch.vdi



選擇虛擬硬碟的大小 (以 MB 位元組為單位)。這個大小是對虛擬機器將能夠存儲在硬碟上的檔案資料量的限制。

4.00 MB 100 GB 2.00 TB



新增(N) 設定(S) 捨棄 啟動(T)

- 一般
- 系統
- 顯示
- 存放裝置

一般

基本(C) 進階(D) 描述(E) 磁碟加密(R)

快照資料夾(N): G:\VirtualBoxVMs\arch\Snapshots

共用剪貼簿(S): 雙向

拖放(R): 雙向

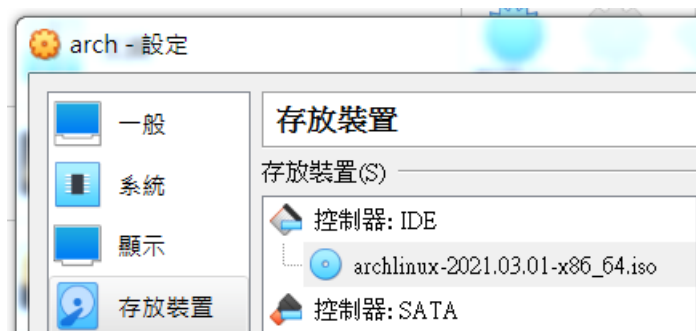
- 一般
- 系統
- 顯示
- 存放裝置
- 音訊
- 網路

系統

主機板(M) 處理器(P) 加速(L)


基本記憶體(M): 4 MB

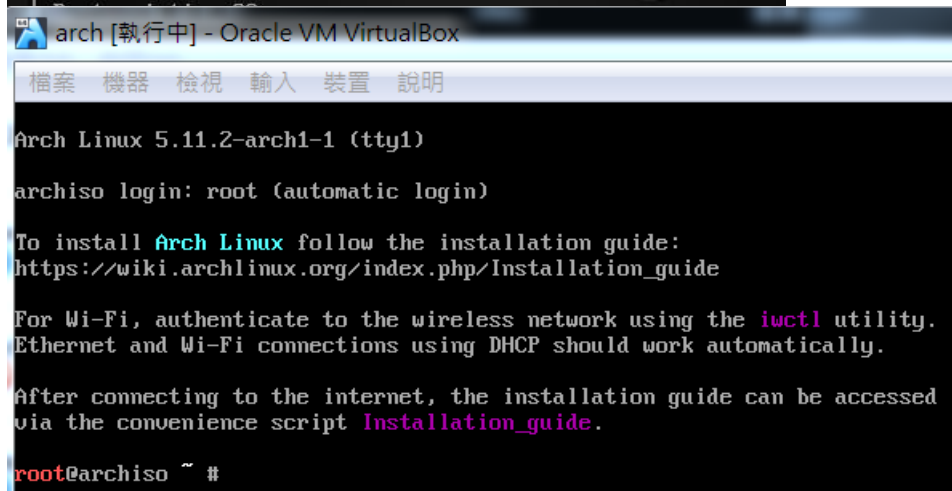
開機順序(B):
☐ 軟碟
☒ 光碟
☒ 硬碟
☐ 網路



請選取虛擬光碟檔或包含磁碟的實體光碟機，以便從其啟動新的虛擬機器。

該磁碟應適合從電腦啟動，並且包含您希望在虛擬機器上安裝的作業系統 (如果您現在要如此做)。下次關閉虛擬機器時，磁碟將自動從虛擬磁碟機中退出，但如果需要，也可以使用「裝置」功能表自行執行此操作。

archlinux-2021.03.01-x86_64.iso (734.32 MB) 



>> **fdisk -l**

```
root@archiso ~ # fdisk -l
Disk /dev/sda: 100 GiB, 107374182400 bytes, 209715200 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop0: 602.61 MiB, 631881728 bytes, 1234144 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ #
```

>> **fdisk /dev/sda**

Select label type

gpt

dos

sgi

sun

Device

Boot

>> Free space

[New]

Free Space New Enter 512M Enter Primary Enter Bootable Enter

Partition size: 512M

Enter Primary

[primary]

Partition type

[Bootable]

Device	Boot	Start	End	Sectors	Size	Id	Type
>> /dev/sda1	*	2048	1050623	1048576	512M	83	Linux
Free space		1050624	209715199	208664576	99.5G		

Free Space New Enter 4G Enter Primary Enter Type Enter 選 82 Enter

Device

Boot

>> Free space

[New]

Partition size: 4G

[Type]

82 Linux swap / Solaris

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	1050623	1048576	512M	83	Linux
>> /dev/sda2		1050624	9439231	8388608	4G	82	Linux swap / Solaris

Free Space New Enter 95.5G Enter Primary Enter Write Enter Yes Enter Quit Enter

Partition size: 95.5G

Write Enter Yes Enter Quit Enter

[Write]

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	1050623	1048576	512M	83	Linux
/dev/sda2		1050624	9439231	8388608	4G	82	Linux swap / Solaris
>> /dev/sda3		9439232	209715199	200275968	95.5G	83	Linux

[Quit]

>> lsblk

```
root@archiso ~ # lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
loop0 7:0 0 602.6M 1 loop /run/archiso/sfs/airootfs
sda 8:0 0 100G 0 disk
├─sda1 8:1 0 512M 0 part
├─sda2 8:2 0 4G 0 part
└─sda3 8:3 0 95.5G 0 part
sr0 11:0 1 734.3M 0 rom /run/archiso/bootmnt
```

>> mkfs.ext4 /dev/sda1

>> mkfs.ext4 /dev/sda3

```
root@archiso ~ # mkfs.ext4 /dev/sda1
mke2fs 1.46.1 (9-Feb-2021)
Creating filesystem with 131072 4k blocks and 32768 inodes
Filesystem UUID: 0c51ef05-4a6d-4da7-87df-8a458f2dd3d7
Superblock backups stored on blocks:
    32768, 98304

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
root@archiso ~ # mkfs.ext4 /dev/sda3
mke2fs 1.46.1 (9-Feb-2021)
Creating filesystem with 25034496 4k blocks and 6258688 inodes
Filesystem UUID: 782e98c2-b405-47e6-afae-b57058224fdc
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done
```

>> mkswap /dev/sda2

```
root@archiso ~ # mkswap /dev/sda2
Setting up swapspace version 1, size = 4 GiB (4294963200 bytes)
no label, UUID=f7014e88-cf3f-433f-883a-73e7b7277330
root@archiso ~ #
```

>> swapon /dev/sda2

```
root@archiso ~ # swapon /dev/sda2
root@archiso ~ #
```

>> nano /etc/pacman.d/mirrorlist ← 至少加入前 4 個 server

Taiwan

Server = http://archlinux.cs.nctu.edu.tw/\$repo/os/\$arch

Server = http://free.nchc.org.tw/arch/\$repo/os/\$arch

Server = http://ftp.tku.edu.tw/Linux/ArchLinux/\$repo/os/\$arch

Server = http://ftp.yzu.edu.tw/Linux/archlinux/\$repo/os/\$arch

Server = http://archlinux.ccns.ncku.edu.tw/archlinux/\$repo/os/\$arch

Server = https://free.nchc.org.tw/arch/\$repo/os/\$arch
Server = http://shadow.ind.ntou.edu.tw/archlinux/\$repo/os/\$arch
Server = https://shadow.ind.ntou.edu.tw/archlinux/\$repo/os/\$arch
Server = https://ftp.yzu.edu.tw/Linux/archlinux/\$repo/os/\$arch

```
# With:      reflector @/etc/xdg/reflectord/reflectord.conf
# When:      2021-03-12 00:51:51 UTC
# From:      https://www.archlinux.org/mirrors/status/json/
# Retrieved: 2021-03-12 00:51:51 UTC
# Last Check: 2021-03-11 23:28:28 UTC

Server = http://archlinux.cs.nctu.edu.tw/$repo/os/$arch
Server = http://free.nchc.org.tw/arch/$repo/os/$arch
Server = http://ftp.tku.edu.tw/Linux/ArchLinux/$repo/os/$arch
Server = http://ftp.yzu.edu.tw/Linux/ArchLinux/$repo/os/$arch
```

ctrl + O

出現 File Name to Write: /etc/pacman.d/mirrorlist ← 按 Enter

```
Server = https://mirrors.neusoft.edu.cn/archlinux/$repo/os/$arch
Server = https://arch.mirror.constant.com/$repo/os/$arch
Server = https://mirror.pseudoform.org/$repo/os/$arch
File Name to Write: /etc/pacman.d/mirrorlist
^G Help      ^I-D DOS Format  ^I-A Append     ^I-B Backup File
^C Cancel    ^I-M Mac Format  ^I-P Prepend    ^T Browse
```

Ctrl+X 離開

```
Server = https://arch.mirror.constant.com/$repo/os/$arch
Server = https://mirror.pseudoform.org/$repo/os/$arch
[ Wrote 84 lines ]
^G Help      ^O Write Out     ^W Where Is     ^K Cut          ^T Execute      ^C Location     ^U Undo
^X Exit      ^R Read File     ^N Replace      ^U Paste        ^J Justify      ^G Go To Line    ^E Redo
```

>>

在本機同步鏡像來源資料庫，輸入：**pacman -Syy**

pacman -Syy

```
root@archiso ~ # pacman -Syy
:: Synchronizing package databases...
core               131.1 KiB   397 KiB/s 00:00 [#####]
extra              1646.1 KiB  381 KiB/s 00:04 [#####]
community          5.4 MiB    495 KiB/s 00:11 [#####]
root@archiso ~ #
```

到現在為止，我們的所有操作都是在光碟（iso）啟動的 Linux 系統中進行的。下面要將真正的 Linux 系統檔安裝到虛擬機器的硬碟中。前文中進行了分區與格式化之後的硬碟分區，首先要經過“掛載”（mount）操作，才能夠供我們訪問。

首先掛載根目錄：

mount /dev/sda3 /mnt

在根目錄中創建啟動目錄（boot），並掛載它：

mkdir -p /mnt/boot

```
root@archiso / # mount /dev/sda3 /mnt
root@archiso / #
root@archiso / # mkdir -p /mnt/boot
root@archiso / #
```

mount /dev/sda1 /mnt/boot

```
root@archiso / # mount /dev/sda1 /mnt/boot
```



```

root@archiso ~ # mount /dev/sda3 /mnt
root@archiso ~ # mkdir -p /mnt/boot
root@archiso ~ # mount /dev/sda1 /mnt/boot
root@archiso ~ #

```

現在終於可以把系統檔安裝到我們剛才創建的硬碟分區中去了。

Arch Linux 團隊為此準備了專門的腳本工具 `pacstrap`，執行下面的命令以安裝系統檔：

安裝系統 於 `/mnt` 目錄下：

輸入下面的命令來引導系統啟動：

```

# pacstrap /mnt base
# pacstrap /mnt base-devel
# pacstrap /mnt linux
# pacstrap /mnt linux-firmware
# pacstrap /mnt grub
# pacstrap /mnt os-prober
# pacstrap /mnt dhcp
# pacstrap /mnt netctl
# pacstrap /mnt intel-ucode
# pacstrap /mnt dhcpcd
# pacstrap /mnt pacman
# pacstrap /mnt nano

```

上述指令也可 打'成一行 但不要打'錯字 不然軟體沒裝完全

```

# pacstrap /mnt base base-devel linux linux-firmware grub os-prober
# pacstrap /mnt dhcp netctl intel-ucode dhcpcd pacman
nano

```

可以看到系統正在同步數據包。

```

root@archiso ~ # pacstrap /mnt base base-devel linux linux-firmware grub os-prober
==> Creating install root at /mnt
==> Installing packages to /mnt
:: Synchronizing package databases...
core               131.1 KiB   410 KiB/s  00:00 [#####] 100%
extra              1040.8 KiB  380 KiB/s  00:01 [#####] 63%

```

```

root@archiso ~ # pacstrap /mnt dhcp netctl intel-ucode dhcpcd pacman nano

```

其中 `/mnt` 是當前根目錄所在的掛載點，`base` 包中是最基本的 Linux 系統檔，`base-devel` 包中包含了一些常用的開發工具，建議同時安裝兩個包。

生成 `fstab` 檔 系統基本軟體安裝完成後，為了讓系統能夠在每次重啟之後自動掛載各個分區（包括交換分區）我們需要把當前的分區掛載方式寫入到一個叫做 `fstab` 的檔中去。執行下面的指令碼 命令，生成 Linux 系統需要的 `fstab` 檔：

```

# genfstab -U -p /mnt >> /mnt/etc/fstab

```

```

root@archiso ~ # genfstab -U -p /mnt >> /mnt/etc/fstab

```

記得檢查 `fstab` 檔的內容，以確保萬無一失。

```

# more /mnt/etc/fstab

```

（ 看內容是否有 `/boot` `swap` ） `/dev/sda3`，`/dev/sda1`，`/dev/sda2`

```

root@archiso ~ # more /mnt/etc/fstab
# Static information about the filesystems.
# See fstab(5) for details.

# <file system> <dir> <type> <options> <dump> <pass>
# /dev/sda3
UUID=782e98c2-b405-47e6-afae-b57058224fdc / ext4 rw,relatime 0 1

# /dev/sda1
UUID=0c51ef05-4a6d-4da7-87df-8a458f2dd3d7 /boot ext4 rw,relatime 0 2

# /dev/sda2
UUID=f7014e88-cf3f-433f-883a-73e7b7277330 none swap defaults 0 0
root@archiso ~ #

```

配置系統 選項

因為還沒有正確安裝 Boot Loader，所以現在我們還不能重新開機系統。好在 Arch Linux 團隊提供的 arch-chroot 工具可以讓我們暫時登錄到已經安裝好的 Linux 系統中去開始系統組態。執行命令：

```
# arch-chroot /mnt /bin/bash | 使用 bash
```

注意提示符變為 **sh-4.3#**，我們已經成功登陸剛剛安裝的 Arch Linux 系統，執行 **ls -al** 試試：

```

root@archiso ~ # arch-chroot /mnt /bin/bash
[root@archiso /]#

```

```
# pacman -S nano | 裝 nano
```

```

[root@archiso /]# pacman -S nano
warning: nano-5.6.1-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) nano-5.6.1-1

Total Installed Size: 2.33 MiB
Net Upgrade Size: 0.00 MiB

:: Proceed with installation? [Y/n] y

```

現在來更改語言配置：

```
# nano /etc/locale.gen
```

移除一行之前的 **"#"** 符號（即取消注釋）來選擇你需要的 locale。我的選擇是：

```

en_US.UTF-8 UTF-8
zh_CN.GBK GBK
zh_CN.UTF-8 UTF-8
zh_TW.UTF-8 UTF-8
zh_TW BIG5

```

包含了美語英文 UTF-8 編碼，簡體中文 GBK 與 UTF-8 編碼，以及繁體中文的 UTF-8 編碼。通過刪除 **#** 以及按 **control + O**，按 **control + X**，然後再按 **return** 來選擇下面的語言配置：

```
GNU nano 2.5.3 File: /etc/locale.gen

#en_CA.UTF-8 UTF-8
#en_CA ISO-8859-1
#en_DE.UTF-8 UTF-8
#en_DE ISO-8859-1
#en_GB.UTF-8 UTF-8
#en_GB ISO-8859-1
#en_HK.UTF-8 UTF-8
#en_HK ISO-8859-1
#en_IE.UTF-8 UTF-8
#en_IE ISO-8859-1
#en_IE@euro ISO-8859-15
#en_IN UTF-8
#en_MG UTF-8
#en_MZ.UTF-8 UTF-8
#en_MZ ISO-8859-1
#en_PH.UTF-8 UTF-8
#en_PH ISO-8859-1
#en_SG.UTF-8 UTF-8
#en_SG ISO-8859-1
#en_US.UTF-8 UTF-8

Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos
Exit      Read File  Replace   Uncut Text  To Spell  Go To Line
```

使用 **locale-gen** 命令來生成 locale 資訊：更新語言環境 以生成 locale

locale-gen

```
[root@archiso /]#
[root@archiso /]# locale-gen
Generating locales...
  en_US.UTF-8... done
  zh_CN.GBK... done
  zh_CN.UTF-8... done
  zh_TW.UTF-8... done
  zh_TW.BIG5... done
Generation complete.
[root@archiso /]#
```

創建 **locale.conf** 並選擇要使用的當地語系化選項

使用下面的命令來創建 **/etc/locale.conf** 配置文件：

[可參考網址 Localization/Traditional Chinese \(正體中文\) - ArchWiki](#)

nano /etc/locale.conf

```
Generation complete.
[root@archiso /]# nano /etc/locale.conf_
```

然後按 **return**。現在你就可以在配置文件中輸入下面一行內容來為系統添加語言：

LANG=en_US.UTF-8

```
arch [執行中] - Oracle VM VirtualBox
檔案 機器 檢視 輸入 裝置 說明
GNU nano 5.6.1 /etc/locale.conf
LANG=en_US.UTF-8
```

然後按 **ctrl+O**，**ctrl+x**，再按 **return**。

完全支持中文環境，但是以英文作為用戶界面

/etc/locale.conf 內容：

LANG=zh_TW.UTF8

LC_MESSAGES=en_US.UTF8

若不使用 **nano /etc/locale.conf** 也可使用 **echo** 方法

（使用 **echo xxx > yyy** 的命令，可以把 **echo** 命令的輸出內容即 **xxx** 重定向到 **yyy** 檔中去，這與使用 **nano** 編輯器對 **yyy** 檔進行編輯效果相同，只是更快捷）：以下命令設置默認 locale

```
# echo LANG=en_US.UTF-8 > /etc/locale.conf 或
```

```
# echo LANGUAGE=en_US >> /etc/locale.conf
```

```
# echo LC_ALL=C >> /etc/locale.conf
```

官方建議：將系統 locale 設置為 en_US.UTF-8，系統的 Log 會以英文顯示，這樣更容易判斷問題和處理。

設置時區, Set Timezone

輸入下面的命令來同步時區：

```
# ls /usr/share/zoneinfo 你將看到整個世界的時區列表。
```

輸入下面的命令來選擇你所在的時區：

```
# ln -sf /usr/share/zoneinfo/Asia/Taipei /etc/localtime <==台北 或
```

```
# ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime ← 上海時區
```

或者你可以從下面的列表中選擇其它名稱。

```
[root@archiso /]# ls /usr/share/zoneinfo
Africa      Chile      Factory    Iceland    Mexico     posixrules US
America     CST6CDT   GB          Indian     MST        PRC        UTC
Antarctica  Cuba      GB-Eire     Iran       MST7MDT    PST8PDT    MET
Arctic      EET       GMT         iso3166.tab Navajo     right      W-SU
Asia        Egypt     GMT0        Israel     NZ         ROC        zone1970.tab
Atlantic    Eire      GMT-0       Jamaica    NZ-CHAT    ROK        zone.tab
Australia   EST       GMT+0       Japan      Pacific    Singapore  Zulu
Brazil      EST5EDT   Greenwich   Kwajalein  Poland     Turkey
Canada      Etc       Hongkong    Libya      Portugal   UCT
CET          Europe    HST         MET        posix      Universal
```

```
[root@archiso /]# ln -sf /usr/share/zoneinfo/Asia/Taipei /etc/localtime
[root@archiso /]# hwclock -w -u
```

```
# hwclock -w -u 硬體時鐘已同步
```

設定主機 名稱：

```
>> echo 你的主機名稱 > /etc/hostname
```

也可使用 nano /etc/hostname 再加入你的主機名稱 輸入你想設置 的主機名稱，按 control + x，按 y，再按 return。

```
範例: echo "job" > /etc/hostname
```

```
範例: echo "nice" > /etc/hostname
```

```
[root@archiso /]# echo nice > /etc/hostname
[root@archiso /]# _
```

接著向 /etc/hosts 文件添加 hosts 條目

```
# nano /etc/hosts
```

```
#<ip-address> <hostname.domain.org><hostname>
```

```
127.0.0.1 localhost.localdomain localhost
```

```
::1 localhost.localdomain localhost
```

```
127.0.1.1 <主機名>.localdomain <主機名>
```

例如 假設主機名為 nice：

```
127.0.0.1 localhost.localdomain localhost
```

```
::1          localhost.localdomain    localhost
127.0.1.1   nice.localdomain        nice
```

```
GNU nano 5.6.1 /etc/hosts
# Static table lookup for hostnames.
# See hosts(5) for details.
127.0.0.1 localhost.localdomain localhost
::1 localhost.localdomain localhost
127.0.0.1 nice.localdomain nice
```

```
>> systemctl enable dhcpd
```

這樣在下一次系統啟動時， dhcpd 將會自動啟動，並自動獲取一個 IP 地址：

```
[root@archiso /]# systemctl enable dhcpd
Created symlink /etc/systemd/system/multi-user.target.wants/dhcpd.service → /usr/lib/systemd/system/dhcpd.service.
[root@archiso /]# _
```

若出現: Failed to enable unit, unit dhcpd.service does not exist.

修正..先安裝 dhcpd 再執行

```
# pacman -S dhcpd
```

```
# dhcpd//有線網路
```

Enable dhcpd (internet connection) 啟用 dhcpd (互聯網連接)

```
# systemctl enable dhcpd
```

```
>> mkinitcpio -p linux
```

```
[root@archiso /]# mkinitcpio -p linux
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'default'
-> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux.img
==> Starting build: 5.11.5-arch1-1
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [autodetect]
-> Running build hook: [modconf]
-> Running build hook: [block]
-> Running build hook: [filesystems]
-> Running build hook: [keyboard]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating zstd-compressed initcpio image: /boot/initramfs-linux.img
==> Image generation successful
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'fallback'
-> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux-fallback.img -S autodetect
==> Starting build: 5.11.5-arch1-1
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [modconf]
-> Running build hook: [block]
==> WARNING: Possibly missing firmware for module: aic94xx
==> WARNING: Possibly missing firmware for module: wd719x
==> WARNING: Possibly missing firmware for module: xhci_pci
-> Running build hook: [filesystems]
-> Running build hook: [keyboard]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating zstd-compressed initcpio image: /boot/initramfs-linux-fallback.img
==> Image generation successful
[root@archiso /]# _
```

重點在最後一句 Image generation successful 映像檔 建立成功

```
==> Creating zstd-compressed initcpio image: /boot/initramfs-linux-fallback.img
==> Image generation successful
[root@archiso /]#
```

安裝 Intel-ucode (非 Intel CPU 可以跳過此步驟)

```
# pacman -S intel-ucode //若 install-ucode 已裝就不用裝
```

```
[root@archiso /]# pacman -S intel-ucode
warning: intel-ucode-20210216-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) intel-ucode-20210216-1

Total Installed Size: 3.46 MiB
Net Upgrade Size: 0.00 MiB

:: Proceed with installation? [Y/n] y_
```

=====

你通常不需創建 `initramfs`，因為在執行 `pacstrap` 安裝 `linux` 套件時，`mkinitcpio` 即會被自動執行。

For special configurations, modify the `mkinitcpio.conf(5)` file and recreate the `initramfs` image: 如有特殊配置的需求，可更改 `mkinitcpoi.conf(5)`[dead link 2018-09-22] 並執行以下命令重新創建一個 `Initramfs`：

mkinitcpio -p linux

```
[root@archiso /]# mkinitcpio -p linux
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'default'
-> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux.img
==> Starting build: 5.11.5-arch1-1
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [autodetect]
-> Running build hook: [modconf]
-> Running build hook: [block]
-> Running build hook: [filesystems]
-> Running build hook: [keyboard]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating zstd-compressed initcpio image: /boot/initramfs-linux.img
==> Image generation successful
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'fallback'
-> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux-fallback.img -S autodetect
==> Starting build: 5.11.5-arch1-1
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [modconf]
-> Running build hook: [block]
==> WARNING: Possibly missing firmware for module: aic94xx
==> WARNING: Possibly missing firmware for module: wd719x
==> WARNING: Possibly missing firmware for module: xhci_pci
-> Running build hook: [filesystems]
-> Running build hook: [keyboard]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating zstd-compressed initcpio image: /boot/initramfs-linux-fallback.img
==> Image generation successful
[root@archiso /]#
```

=====

設置 root 密碼： Set password for root

>> passwd ← 使用 **root** 的密碼 要記住密碼

```
[root@archiso etc]# echo beta > /etc/hostname
[root@archiso etc]# passwd
New password:
Retype new password:
passwd: password updated successfully
[root@archiso etc]#
```

[新增使用者]

`useradd -m 使用者名稱 # 新增使用者(-m 在 Home 建立目錄)`

passwd 使用者名稱 # 設定密碼

新建一個用戶， 加入"wheel" group 使用 bash

```
#useradd -m -g users -G wheel -s /bin/bash wang
```

或

```
# useradd -m -g users -G wheel -s /usr/bin/fish MYUSERNAME
```

```
[root@archiso /]# useradd -m -g users -G wheel -s /bin/bash wang
[root@archiso /]#
[root@archiso /]# useradd -m -g users -G wheel -s /bin/bash wa
[root@archiso /]#
```

並為新用戶設置密碼

```
#passwd username
```

例如:

```
#passwd wang
```

輸入密碼 記下你輸入的密碼

```
[root@archiso /]# passwd wang
New password:
Retype new password:
passwd: password updated successfully
[root@archiso /]# passwd wa
New password:
Retype new password:
passwd: password updated successfully
[root@archiso /]#
```

為你剛才創建的用戶 添加 sudo 權限

nano /etc/sudoers

```
[root@archiso /]# nano /etc/sudoers_
```

找到 # %wheel ALL=(ALL)ALL 這行，去掉之前的#註釋符

找到 root ALL=(ALL) ALL 再下一行 加入你使用者的權限

newuser ALL=(ALL) ALL

例如

wang ALL=(ALL) ALL

```
##
## User privilege specification
##
root ALL=(ALL) ALL

## Uncomment to allow members of group wheel to execute any command
%wheel ALL=(ALL) ALL
wang ALL=(ALL) ALL
wa ALL=(ALL) ALL
```

ctrl+o 寫入檔案 , **ctrl+x** 離開

這裡的%wheel 就是代表 wheel 組，意味著 wheel 組中的所有用戶都可以使用 sudo 命令

```
# ALL ALL=(ALL) ALL # WARNING: only use this together with 'Defaults targetpw'
File Name to Write: /etc/sudoers
^G Help          ^M-D DOS Format  ^M-A Append      ^M-B Backup File
^C Cancel        ^M-M Mac Format  ^M-P Prepend     ^M-T Browse
```

設定 DNS

nano /etc/resolv.conf

或

vim /etc/resolv.conf

將所有設定前方加上 # 作註解，添加以下 DNS (最少 1 種，看個人選擇)

```
nameserver 163.23.1.66          # dyu1
nameserver nameserver 163.23.1.77 # dyu2
nameserver 8.8.8.8              #Google
nameserver 8.8.4.4              #Google
nameserver 168.95.192.1         #中華電信
nameserver 168.95.1.1          #中華電信
```

```
# See man:systemd-resolved.service(8) for
# operation for /etc/resolv.conf.
nameserver 8.8.8.8
nameserver 8.8.4.4
nameserver 163.23.1.66
nameserver 163.23.1.77
nameserver 168.95.192.1
nameserver 168.95.1.1
```

```
^G Help      ^O Write Out  ^W Where Is
^X Exit      ^R Read File  ^_ Replace
```

=====

安裝必要網路工具

```
# pacman -S net-tools
# pacman -S wireless_tools
# pacman -S dhclient
# pacman -S wpa_supplicant
```

```
[root@archiso /]# pacman -S net-tools wireless_tools dhclient wpa_supplicant
resolving dependencies...
looking for conflicting packages...

Packages (5) run-parts-4.8.6.1-2 dhclient-4.4.2-2 net-tools-2.10-1 wireless_tools-30.pre9-3
wpa_supplicant-2:2.9-8

Total Download Size: 2.69 MiB
Total Installed Size: 9.44 MiB

:: Proceed with installation? [Y/n] y_
```

安裝 工具

```
# pacman -S dhcp
# pacman -S vim pkgfile bash-completion sudo
# pacman -S nano          已裝就不用裝
# pacman -S unzip ctags
```

```
[root@archiso /]# pacman -S dhcp vim pkgfile bash-completion sudo nano unzip ctags
warning: dhcp-4.4.2-2 is up to date -- reinstalling
warning: sudo-1.9.5.p2-1 is up to date -- reinstalling
warning: nano-5.6.1-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (12) gpm-1.20.7.r38.ge82d1a6-3 jansson-2.13.1-1 libyaml-0.2.5-1 vim-runtime-8.2.2489-1
bash-completion-2.11-1 ctags-1:r20210106+g08b1c490-1 dhcp-4.4.2-2 nano-5.6.1-1
pkgfile-21-2 sudo-1.9.5.p2-1 unzip-6.0-14 vim-8.2.2489-1

Total Download Size: 9.26 MiB
Total Installed Size: 53.67 MiB
Net Upgrade Size: 37.27 MiB

:: Proceed with installation? [Y/n] y_
```

pacman 說明:

pacman -S 套件名 pacman

-Sy 同步列表 pacman -Su 更新套

件版本 pacman -Syu 同步更新

8. 安裝 Boot Loader 安裝引導加載程序 電腦開機之後，處理器會到固定的位置去讀取磁片內容，這個“固定的位置”一般就在磁片的最開始

位置。放在這個位置的軟體，就叫做 Boot Loader，它的作用是為處理器指明作業系統的安裝位置，使得處理器可以進一步的讀取並執行作業系統。

下載 Grub Boot Loader，將其安裝到 Arch Linux 被安裝到的硬碟中 (sda)，然後生成啟動所需要的 grub 檔。

```
>> pacman -S grub os-prober //會下載相關程序
```

```
[root@archiso /]# pacman -S grub os-prober
warning: grub-2:2.04-10 is up to date -- reinstalling
warning: os-prober-1.78-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (2) grub-2:2.04-10  os-prober-1.78-1

Total Installed Size: 33.00 MiB
Net Upgrade Size:      0.00 MiB

:: Proceed with installation? [Y/n] y
```

安裝 grub-bios 套件(Install grub-bios package)

```
# pacman -S grub-bios
```

```
[root@archiso /]# pacman -S grub-bios
warning: grub-2:2.04-10 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) grub-2:2.04-10

Total Installed Size: 32.94 MiB
Net Upgrade Size:      0.00 MiB

:: Proceed with installation? [Y/n] y
```

Note: 僅安裝套件並不會更新 /boot/grub/i386-pc/core.img 檔案和 GRUB(2) 在 /boot/grub/i386-pc 的模組。你必須如下解釋般使用 grub-install 來更新他們

使用下面的命令來將啟動加載程序安裝到硬碟上：

```
# grub-install --target=i386-pc --recheck --debug --force /dev/sda
```

```
[root@archiso /]# grub-install --target=i386-pc --recheck --debug --force /dev/sda
```

```
grub-install: info: saving <101,0,512>.
grub-install: info: saving <102,0,512>.
grub-install: info: saving <103,0,512>.
grub-install: info: saving <104,0,512>.
Installation finished. No error reported.
```

```
# grub-install --target=i386-pc --recheck /dev/sda
```

[https://wiki.archlinux.org/index.php/GRUB_\(%E6%AD%A3%E9%AB%94%E4%B8%AD%E6%96%87\)](https://wiki.archlinux.org/index.php/GRUB_(%E6%AD%A3%E9%AB%94%E4%B8%AD%E6%96%87))

產生 GRUB2 BOIS 設置檔案(Generate GRUB2 BIOS Config file)

最後產生一個 GRUB2 的設定檔 (this is explained in greater detail in the Configuration section):

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

注意： 檔案路徑是 /boot/grub/grub.cfg, 不是 /boot/grub/i386-pc/grub.cfg.

```
>> grub-mkconfig - /boot/grub/grub.cfg
```

```
[root@archiso /]# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/intel-ucode.img /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot: initramfs-linux-fallback.img
done
[root@archiso /]# _
```

os-prober 可以用以偵測其他系統的存在，並在之後加入 grub 選單中。

os-prober

```
[root@archiso /]# os-prober
[root@archiso /]#
```

下面從安裝的 Linux 中退出，卸載剛才掛載的檔案系統，然後關機。這樣原始 Arch Linux 的安裝就完成了。

=====

在離開 chroot 以前記得把 dhcp 服務加入 systemd 的起動選項中

systemctl enable dhcpd.service

```
[root@archiso /]# systemctl enable dhcpd.service
[root@archiso /]#
```

exit //使用命令 exit 或 [CTRL+D] 退回安裝環境

```
[root@archiso /]# exit
exit
arch-chroot /mnt /bin/bash 24.05s user 8.12s system 0% cpu 2:40:00.03 total
root@archiso ~ #
```

reboot 重開機 poweroff

```
root@archiso ~ #
root@archiso ~ #
root@archiso ~ # reboot
```

重開機

選項要選 **Boot existing OS** (不是 boot arch linux)

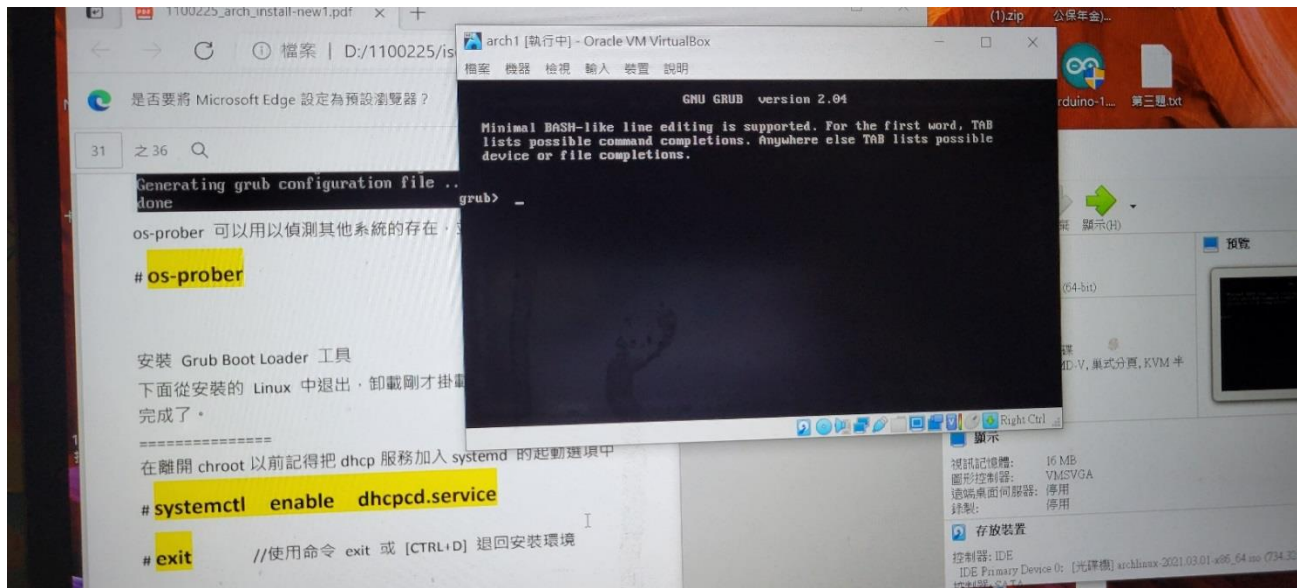
```
Arch Linux
Arch Linux install medium (x86_64, BIOS)
Arch Linux install medium (x86_64, BIOS) with speech
Boot existing OS
```

然後以新的 user login

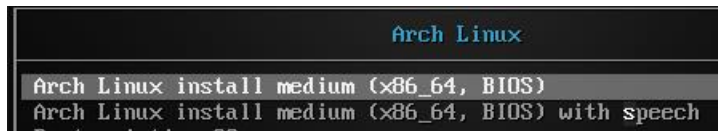
login in 後 ping 網路是否有通

ping 8.8.8.8

=====



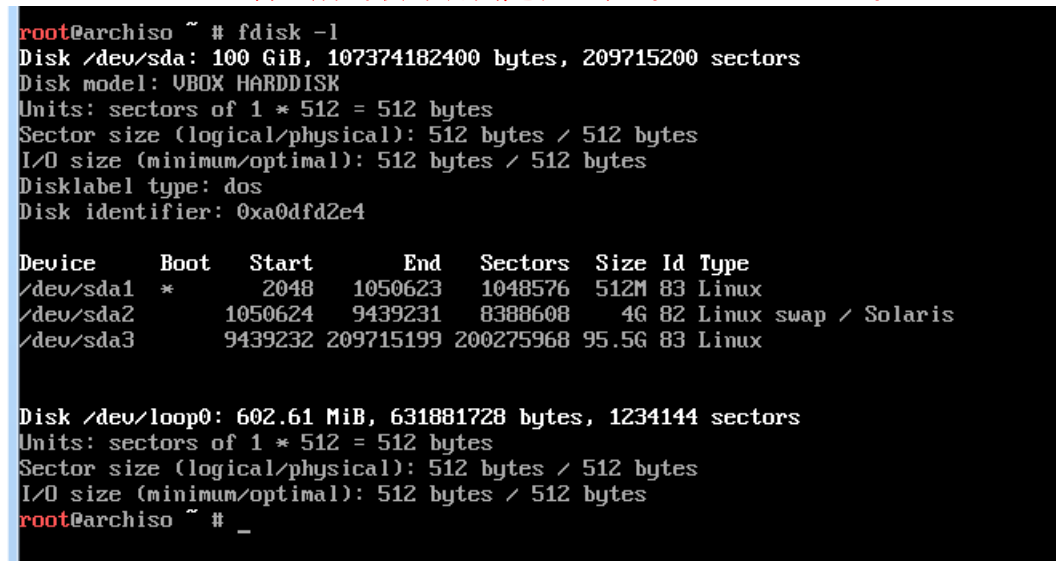
若開不了機無法出現 login 畫面 則要重新使用光碟開機 選第一項
Arch Linux install medium (x86_64, BIOS)
然後再重新設定



1. 先用光碟的 iso 檔開機



2. fdisk -l ←看之前的硬碟切割是否還在 (sda1, sda2, sda3)



3. mount 硬碟到 /mnt

mount /dev/sda3 /mnt

4. mount 硬碟到 /mnt/boot 當然之前已經 "mkdir -p /mnt/boot"

mount /dev/sda1 /mnt/boot/

```
root@archiso ~ # mount /dev/sda3 /mnt
root@archiso ~ # mount /dev/sda1 /mnt/boot
```

5.

>> arch-chroot /mnt /bin/bash

>> pacman -S grub-bios

>> grub-install --target=i386-pc --recheck --debug --force /dev/sda

```
[root@archiso /]# grub-install --target=i386-pc --recheck --debug --force /dev/sda
grub-install: info: saving <99,0,512>.
grub-install: info: saving <100,0,512>.
grub-install: info: saving <101,0,512>.
grub-install: info: saving <102,0,512>.
grub-install: info: saving <103,0,512>.
grub-install: info: saving <104,0,512>.
Installation finished. No error reported.
```

或

>> grub-install --target=i386-pc --recheck /dev/sda

>> grub-mkconfig - /boot/grub/grub.cfg

```
[root@archiso /]#
[root@archiso /]#
[root@archiso /]# grub-mkconfig /boot/grub/grub.cfg
```

```
### END /etc/grub.d/10_linux ###

### BEGIN /etc/grub.d/20_linux_xen ###
### END /etc/grub.d/20_linux_xen ###

### BEGIN /etc/grub.d/30_os-prober ###
### END /etc/grub.d/30_os-prober ###

### BEGIN /etc/grub.d/40_custom ###
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
### END /etc/grub.d/40_custom ###

### BEGIN /etc/grub.d/41_custom ###
if [ -f ${config_directory}/custom.cfg ]; then
    source ${config_directory}/custom.cfg
elif [ -z "${config_directory}" ] -a -f $prefix/custom.cfg ]; then
    source $prefix/custom.cfg;
fi
### END /etc/grub.d/41_custom ###
done
[root@archiso /]#
```

os-prober 可以用以偵測其他系統的存在，並在之後加入 grub 選單中。

os-prober

```
[root@archiso /]# os-prober
[root@archiso /]#
```

下面從安裝的 Linux 中退出，卸載剛才掛載的檔案系統，然後關機。這樣原始 Arch Linux 的安裝就完成了。

=====

在離開 chroot 以前記得把 dhcp 服務加入 systemd 的起動選項中

systemctl enable dhcpcd.service

```
[root@archiso /]# systemctl enable dhcpcd.service  
[root@archiso /]#
```

exit //使用命令 `exit` 或 `[CTRL+D]` 退回安裝環境

```
[root@archiso /]# exit  
exit  
arch-chroot /mnt /bin/bash 24.05s user 8.12s system 0% cpu 2:40:00.03 total  
root@archiso ~ #
```

reboot 重開機 `poweroff`

```
root@archiso ~ #  
root@archiso ~ #  
root@archiso ~ # reboot
```

重開機

