

NAR Labs

國家實驗研究院

國網中心臺灣衫三號 介紹與使用

高銓澤

資料網址：

<https://ppt.cc/fkJKDx>

2022/07/26

www.narlabs.org.tw

綱要

一、國網中心與臺灣衫三號簡介

二、Linux 系統使用入門

三、SLURM 操作與環境設定

國網中心與臺灣衫三號簡介

國家高速網路與計算中心

國網中心沿革

1991 年成立 行政院國家科學委員會 國家高速電腦中心

2003 年改制 財團法人國家實驗研究院 國家高速網路與計算中心



ISO 9001
品質管理



ISO 27001
資訊安全管理



BS 10012
個人資料管理



ISO 50001
能源管理



新竹本部

30076 新竹市東區研發六路7號

電話：886-3-5776085

傳真：886-3-5776082

1993 年



臺南分部

74147 臺南市新市區南科三路28號

電話：886-6-5050940

傳真：886-6-5050945

2005 年



臺中分部

40763 臺中市西屯區科園路22號

電話：886-4-2462-0202、886-4-2465-0818

傳真：886-4-2462-7373

2008 年

國家高速網路與計算中心

國網中心超級電腦建置

2002 年 IBM Cluster 1350, No. **60** in TOP500

2007 年 IRIS, No. **35** in TOP500

2011 年 御風者, No. **42** in TOP500

2012 年 Formosa 5, 自主研發, No. **232** in TOP500

2017 年 臺灣衫一號, No. **95** in TOP500, No. **31** in GREEN500

2018 年 臺灣衫二號, No. **20** in TOP500, No. **10** in GREEN500

2020 年 臺灣衫三號, No. **181** in TOP500, No. **69** in GREEN500

Floating-point operations per second (FLOPS)
K, M, G, T, P, E



台灣衫一號
TAIWANIA 1



2017/11

台灣衫一號TAIWANIA 1

計算力 1.7 PFLOPS
TOP500 #95
Green500 #31



2018/11

台灣衫二號TAIWANIA 2

計算力 9 PFLOPS
TOP500 #20
Green500 #10

台灣衫三號
TAIWANIA 3



2020/11

台灣衫三號TAIWANIA 3

計算力 2.7 PFLOPS
TOP500 #181
Green500 #69

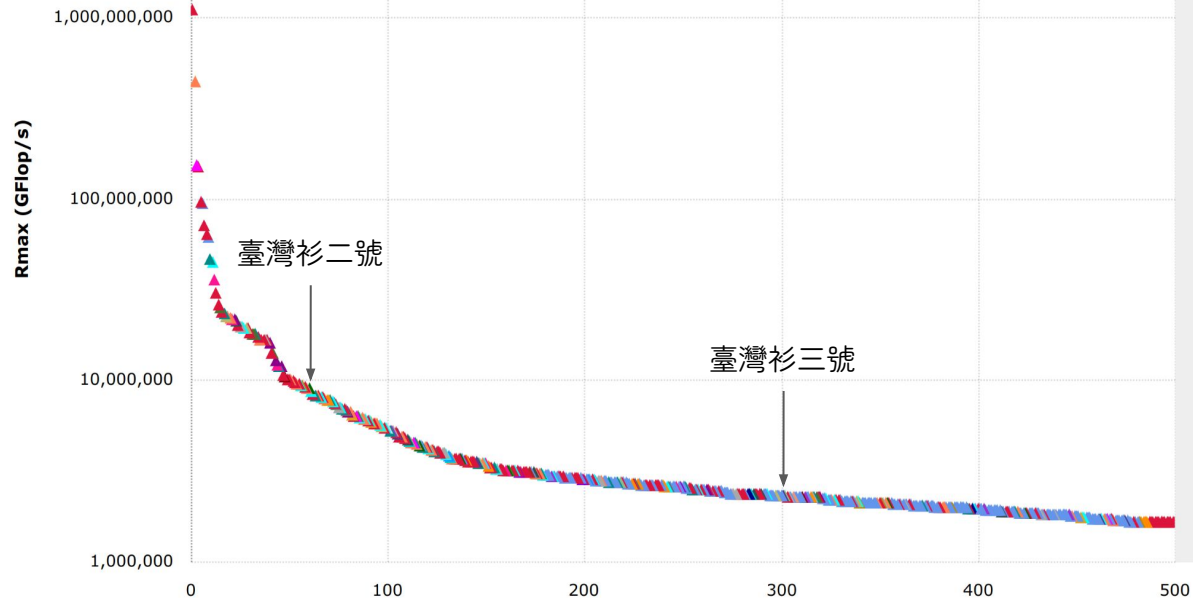
Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	Frontier - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	LUMI - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	1,110,144	151.90	214.35	2,942

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	Frontier - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
59	Taiwania 2 - QCT QuantaGrid D52G-4U/LC, Xeon Gold 6154 18C 3GHz, Mellanox InfiniBand EDR, NVIDIA Tesla V100 SXM2, Quanta Computer / Taiwan Fixed Network / ASUS Cloud National Center for High Performance Computing Taiwan	170,352	9.00	15.21	798

~4x Energy
efficiency

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	Frontier - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
302	Taiwania 3 - QCT QuantaPlex T42D-2U/4N, Xeon Platinum 8280 28C 2.7GHz, InfiniBand HDR 100, Quanta Computer / Taiwan Fixed Network / ASUS Cloud National Center for High Performance Computing Taiwan	50,400	2.30	4.35	564

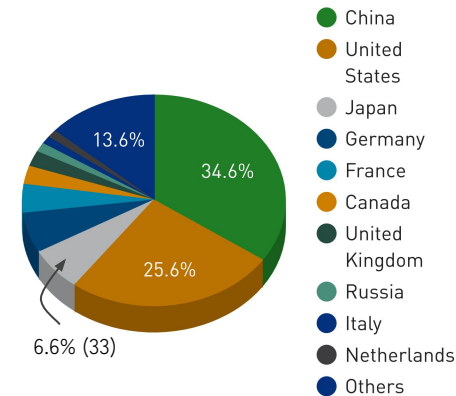
TOP500



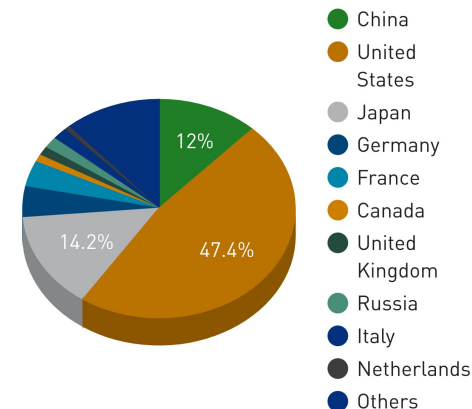
Legend:

Japan, China, United States, Germany, Ireland, France, Australia, United Kingdom, Brazil, Netherlands, Saudi Arabia, Russia, South Korea, Canada, Sweden, Luxembourg, Taiwan, Austria, Poland, Switzerland, Norway, India, Singapore, Italy, Hungary, Czechia, Slovenia, Morocco, Bulgaria, Finland, Spain, United Arab Emirates,

Countries System Share



Countries Performance Share



臺灣衫三號

CPU 計算節點數量：

900

CPU 計算節點規格：

CPU：2 顆 Intel® Xeon® Platinum 8280 (2.7 GHz, 28 cores per CPU)

主記憶體：192 GiB

網路卡：NVIDIA/Mellanox InfiniBand HDR100 (100 Gbps)

作業系統：CentOS 7

排程軟體：SLURM

臺灣衫三號

GPU 計算節點數量：

12

GPU 計算節點規格：

CPU：2 顆 Intel® Xeon® Platinum 8280 (2.7 GHz, 28 cores per CPU)

主記憶體：768 GB

圖形加速卡：8 張 NVIDIA V100 32 GB

網路卡：NVIDIA/Mellanox InfiniBand HDR100 (100 Gbps)

作業系統：CentOS 7

排程軟體：SLURM

臺灣衫三號

大記憶體計算節點數量：

4

大記憶體計算節點規格：

CPU：2 顆 Intel® Xeon® Platinum 8280 (2.7 GHz, 28 cores per CPU)

主記憶體：6 TB

網路卡：NVIDIA/Mellanox InfiniBand HDR100 (100 Gbps)

作業系統：CentOS 7

排程軟體：SLURM

臺灣衫三號

登入節點數量：

4

登入節點網址：

t3-c1.nchc.org.tw

t3-c2.nchc.org.tw

t3-c3.nchc.org.tw

t3-c4.nchc.org.tw

帳號取得 ⇒ iService

iService 網站：[會員中心](#) [服務介紹](#) [操作說明](#) [常見問題](#) [註冊](#) [登入](#) [EN](#)

計算資源服務網，提供用戶建立國網中心使用者與使用者計算專案的網頁界面

iService 帳號：

以電子信箱為帳號名，用於登入 iService 網站
計算資源服務網

主機帳號：

以小寫英文字母、數字或底線的組合為帳號名，用於登入國網中心計算主機

帳號取得 ⇒ iService

iService 網址：

https://iservice.nchc.org.tw/nchc_service/index.php



帳號取得 ⇒ iService

iService 網址：

https://iservice.nchc.org.tw/nchc_service/index.php



帳號取得 ⇒ iService

iService 網址：

https://iservice.nchc.org.tw/nchc_service/index.php

計畫管理

▶ 我的計畫



學術計畫

長庚大學資工暑期課程 大數據應用

建立者名稱：高銓澤 計畫系統代號：ACD111078 計畫編號：

成員數：1，未認證：0

計畫執行期間：2022-07-04 ~ 2022-08-31

申請通過

核可額度：16,000.0 已用額度：0.0 剩餘額度：16,000.0 可用額度：16,000.0

帳號取得 ⇒ iService

iService 網址：

https://iservice.nchc.org.tw/nchc_service/index.php



帳號取得 ⇒ iService

iService 網址：

https://iservice.nchc.org.tw/nchc_service/index.php

修改主機帳號基本資料

主機帳號

主機密碼

OTP 認證碼顯示

最近一次變更主機密碼日期：2022-07-08 12:20:36

說明：

1. 以上是您未來登入主機的帳號資訊，此帳號如建立後，不提供更名之服務。
2. 為了讓您體驗及熟悉主機之環境，特別貼心的為初次申請者，自動提供台灣杉一號(Taiwania 1)主機免費使用的額度。
3. 未來如額度不敷使用時，敬請透過本服務網提出計畫申請及購買使用額度。
4. 登入台灣杉一號(Taiwania 1)主機時，需要輸入您的主機密碼加上本服務網提供的一次性密碼 (OTP)
5. 要在電腦或行動裝置上快速查看 OTP，可以安裝下列推薦的 Authenticator，安裝後啟動應用程式並掃描上方認證碼旁 QR Code 即可快速查看。

安裝 Authenticator (查看安裝與操作說明)

行動裝置版本
(使用行動裝置瀏覽本網頁時請點我安裝)

電腦版本
Authenticator for Firefox
(Firefox 擴充套件)
Authenticator for Google Chrome
(Google Chrome 擴充功能)
Authenticator for Microsoft Edge
(僅支援 windows 10，由 Microsoft Store 下載安

帳號取得 ⇒ iService

iService 網址：

https://iservice.nchc.org.tw/nchc_service/index.php

修改主機帳號基本資料

主機帳號

主機密碼

OTP 認證碼隱藏

569835

03

最近一次變更主機密碼日期：2022-07-08 12:20:36

說明：

1. 以上是您未來登入主機的帳號資訊，此帳號如建立後，不提供更名之服務。

2. 為了讓您體驗及熟悉主機之環境，特別貼心的為初次申請者，自動提供台灣衫一號(Taiwania 1)主機免費使用的額度。

3. 未來如額度不敷使用時，敬請透過本服務網提出計畫申請及購買使用額度。

4. 登入台灣衫一號(Taiwania 1)主機時，需要輸入您的主機密碼加上本服務網提供的一次性密碼 (OTP)

5. 要在電腦或行動裝置上快速查看 OTP，可以安裝下列推薦的 Authenticator，安裝後啟動應用程式並掃描上方認證碼旁 QR Code 即可快速查看。

安裝 Authenticator (查看安裝與操作說明)

行動裝置版本

(使用行動裝置瀏覽本網頁時請點我安裝)

電腦版本

Authenticator for Firefox

(Firefox 擴充套件)

Authenticator for Google Chrome

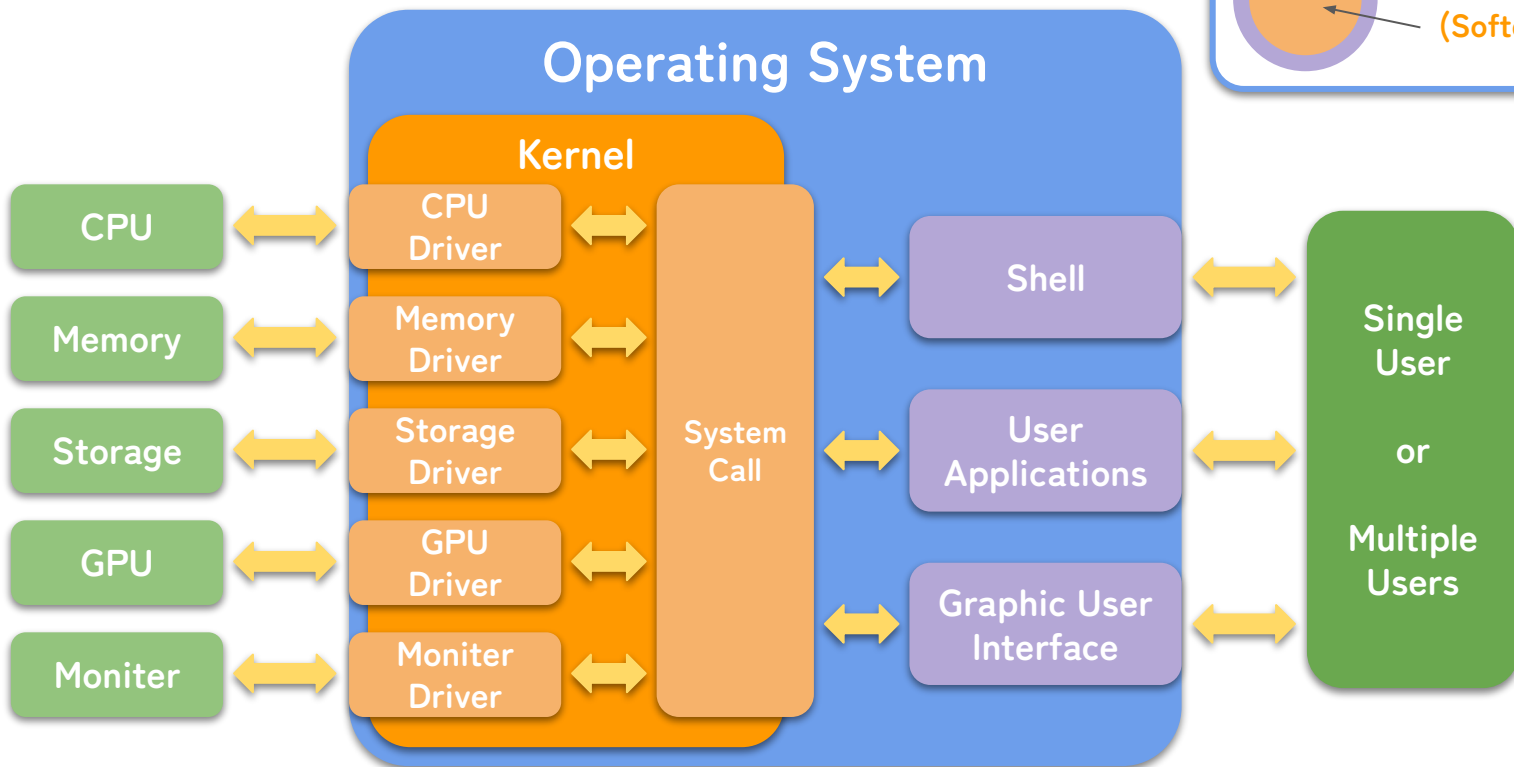
(Google Chrome 擴充功能)

Authenticator for Microsoft Edge

Linux 系統使用入門

什麼是 Linux 系統？

使用 Linux kernel 的作業系統 (Operating System)



常見的 Linux 系統

Linux Distribution: 公司或是團體以 Linux kernel 為基礎，搭配挑選過的應用程式套件、特有的應用程式與特有的修改，來發行的作業系統。

Debian, Ubuntu

Red Hat Enterprise Linux (RHEL),
Fedora Linux, CentOS

SUSE Linux Enterprise, openSUSE

Arch Linux

Gentoo Linux



Ubuntu



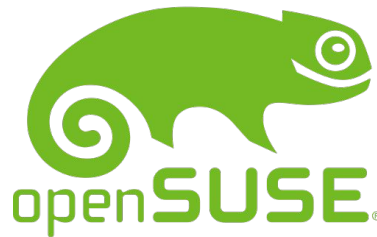
Red Hat
Enterprise Linux



fedora



CentOS



特殊的 Linux 系統

Android → 手機、智慧電視、Chromecast

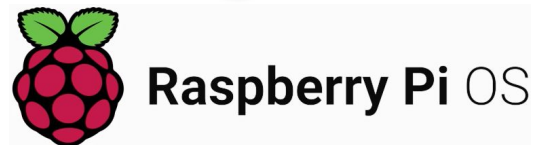
Chrome OS → Chromebooks, Chromeboxes

Chrome OS Flex → 舊型電腦

Tiny Core Linux → 舊型電腦, 極低規格電腦

Raspberry Pi OS → 特定小型裝置

OpenWrt → 部分無線路由器



各常見 Linux Distribution 的特色

deb 套件格式

Debian：安全嚴謹，套件豐富

Ubuntu：Debian 的分支，美觀，社群支援豐富

rpm 套件格式

Red Hat Enterprise Linux (RHEL)：企業版

Fedora Linux：RHEL 的家用版

CentOS：RHEL 的分支，免費的社群企業版

SUSE Linux Enterprise：企業版, yast

openSUSE：SUSE 的家用版

Arch Linux, Gentoo Linux



Ubuntu



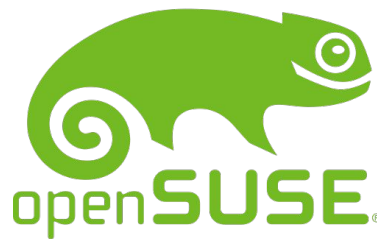
Red Hat
Enterprise Linux



fedora



CentOS



archlinux



gentoo linux™

Unix kernel

Unix, Darwin

Unix-like kernel

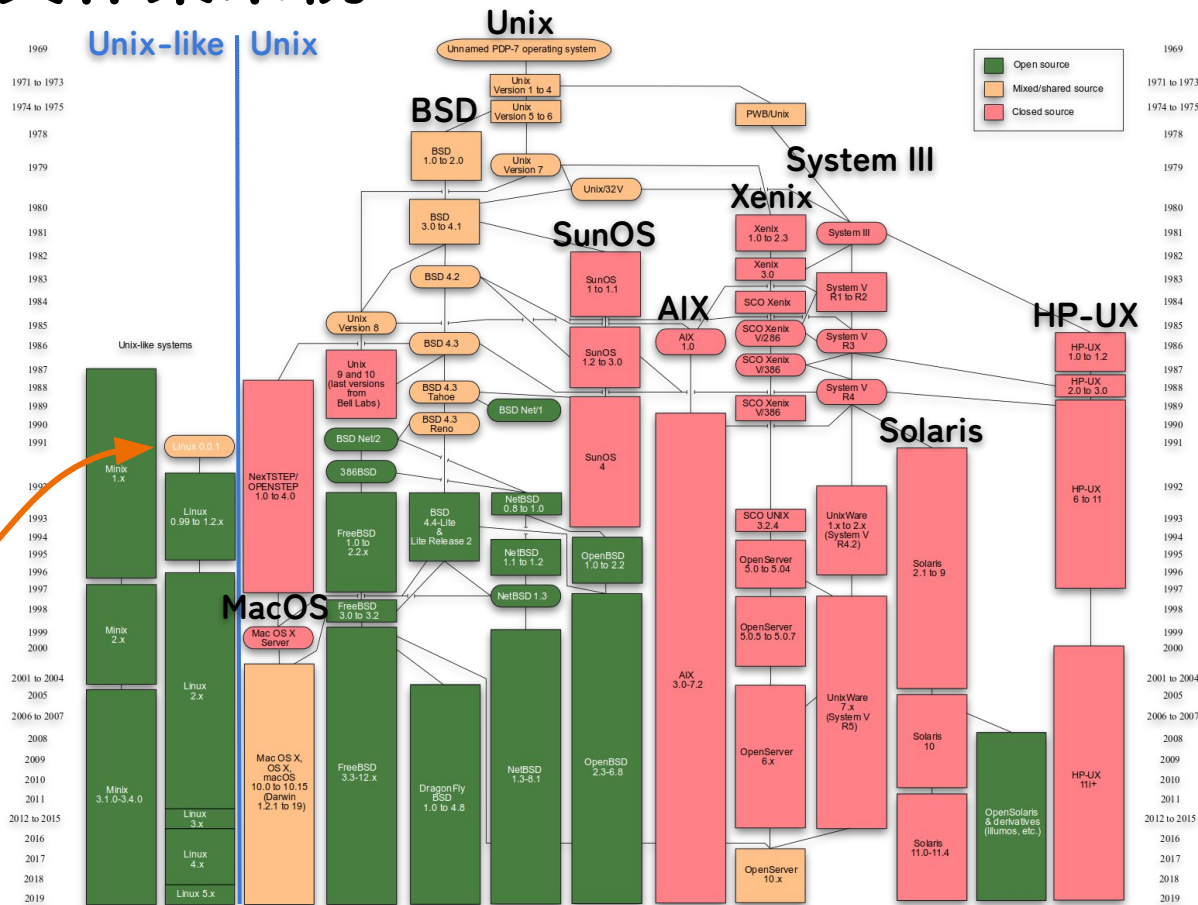
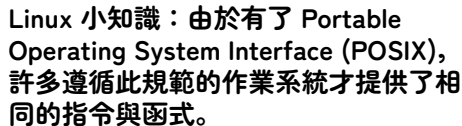
Minix, Linux

Windows kernel

Windows NT

Linux 小知識：1991 年，Linus Torvalds 撰寫的第一版 Linux kernel 被放上了網路

Linux 小知識：Linux kernel 的吉祥物是一隻叫做 Tux 的企鵝



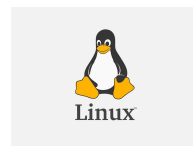
為什麼使用 Linux 系統做計算？



Windows



MacOS



Linux

Unix

硬體支援度



穩定度



安全性



能否自行修改

X

X

✓

✓ / X

軟體支援度



免費

X

X

✓

✓ / X

好上手（個人經驗）



連線使用 Linux

文字界面

通訊協定：Secure Shell (SSH)

程式：

Windows：

PuTTY (open source)

Bitvise (closed source)

MobaXterm (partial open source)

iOS：

termius (closed source)

Andriod：

termius (closed source)

ConnectBot (open source)



PuTTY



MobaXterm



termius



ConnectBot

連線使用 Linux

圖形界面 **！注意：臺灣衫三號並不提供此方式連線**

通訊協定：

Remote Desktop Protocol (RDP)

Remote FrameBuffer (RFB)

Web Real-Time Communication (WebRTC)

...

程式：

Windows：遠端桌面 (RDP) → Linux xrdp server

跨平臺：

TeamViewer (?)

RealVNC (RFB)

Chrome Remote Desktop (WebRTC)



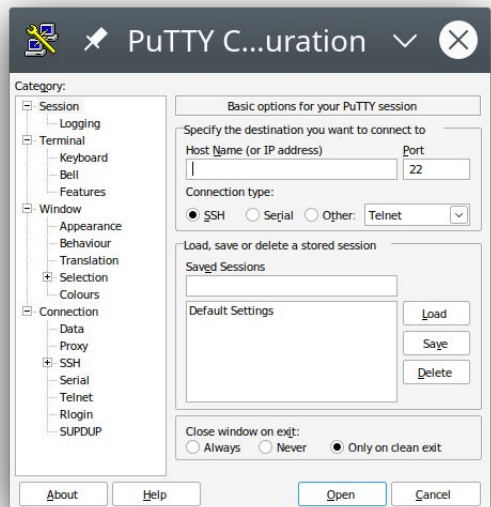
連線使用臺灣衫三號生醫專用節點

可以使用網址：t3-c1.nchc.org.tw

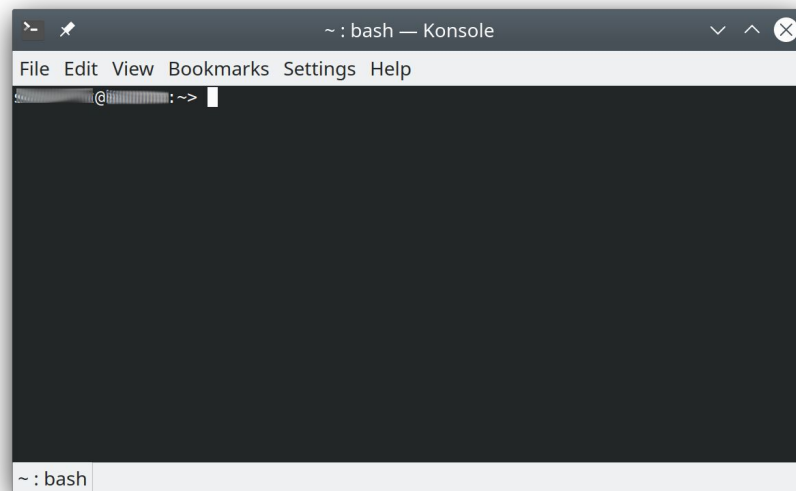
或者是 IP：203.145.216.51

連接埠 (port)：22 (ssh 的預設埠)

PuTTY



Konsole



以 PuTTY 爲範例連線

Google putty ← 1. google 搜尋 putty

全部 圖片 影片 新聞 購物 更多 工具

約有 133,000,000 項結果 (搜尋時間: 0.30 秒)

2. 點選此搜尋結果

<https://www.putty.org> ▾ 翻譯這個網頁

Download PuTTY - a free SSH and telnet client for Windows

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with ...

Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

Download PuTTY ← 3. 點選此連結

Below suggestions are independent of PuTTY. They are not endorsements by the PuTTY project.

Bitvise SSH Client

Bitvise SSH Client is an SSH and SFTP client for Windows. It is developed and supported professionally by Bitvise. The SSH Client is robust, easy to install, easy to use, and supports all features supported by PuTTY, as well as the following:

- graphical SFTP file transfer;
- single-click Remote Desktop tunneling;
- auto-reconnecting capability;
- dynamic port forwarding through an integrated proxy;
- an FTP-to-SFTP protocol bridge.

Bitvise SSH Client is **free to use**.

Download Bitvise SSH Client

Download PuTTY: latest release (0.77)

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)

Download: [Stable](#) | [Snapshot](#) | [Docs](#) | [Changes](#) | [Wishlist](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.77, released on 2022-05-27.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a [permanent link to the 0.77 release](#).

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the [development snapshots](#), to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm).

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

MSI ("Windows Installer")

64-bit x86: [putty-64bit-0.77-installer.msi](#) (signature)

64-bit Arm: [putty-arm64-0.77-installer.msi](#) (signature)

32-bit x86: [putty-0.77-installer.msi](#) (signature)

Unix source archive

.tar.gz: [putty-0.77.tar.gz](#) (signature)

4-1. 可以點選此連結下載安裝程式，或者使用下圖的免安裝程式

Alternative binary files

The installer packages above will provide versions of all of these (except PuTTYtel and pterm), but you can download standalone binaries one by one if you prefer.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

putty.exe (the SSH and Telnet client itself)

64-bit x86: [putty.exe](#) (signature)

64-bit Arm: [putty.exe](#) (signature)

32-bit x86: [putty.exe](#) (signature)

pscp.exe (an SCP client, i.e. command-line secure file copy)

64-bit x86: [pscp.exe](#) (signature)

64-bit Arm: [pscp.exe](#) (signature)

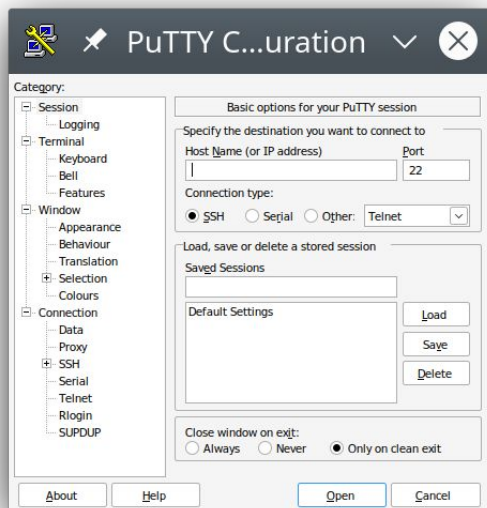
32-bit x86: [pscp.exe](#) (signature)

4-2. 可以點選此連結下載免安裝程式

本範例使用 4-2 的方式

以 PuTTY 爲範例連線

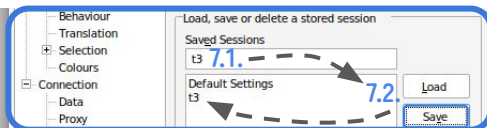
5. 點選下載好的 putty.exe 檔，得如下視窗



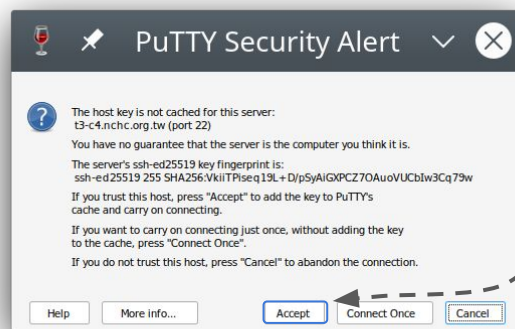
6. 填入網址或是 IP，確認埠號 22 與 SSH 協定



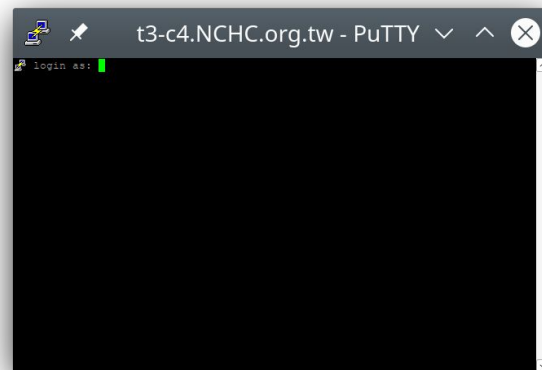
7. 在下圖中填入好記的名稱來儲存這次的設定



8. 第一次連線會出如下視窗，點選 Accept 即可

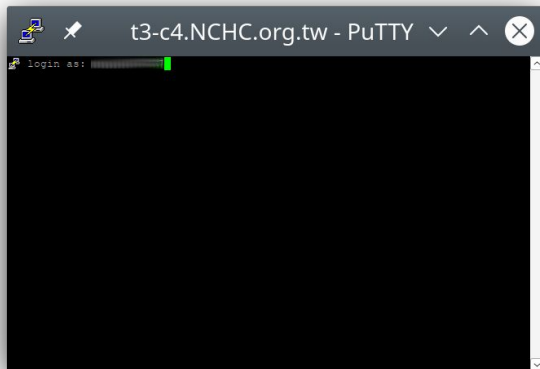


9. 看到下圖表示已經連線成功了

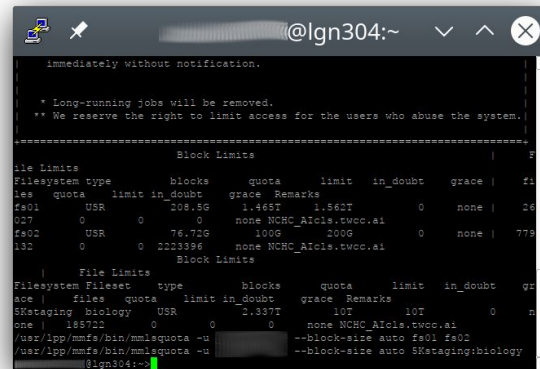


以 PuTTY 為範例連線

10. 輸入主機帳號



13. 登入成功會出現類似如下畫面

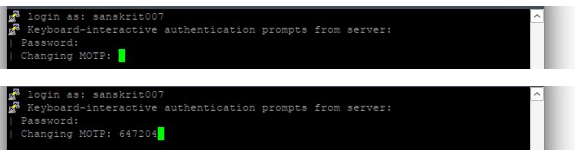


11. 輸入密碼 ! 注意：輸入密碼時，不會顯示任何字元



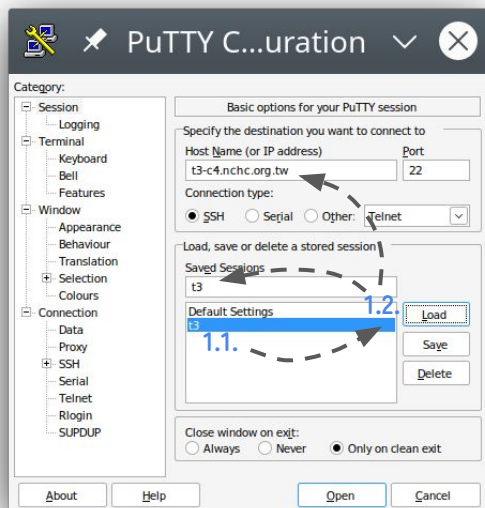
14. 輸入 exit 可以登出

12. 輸入 OTP

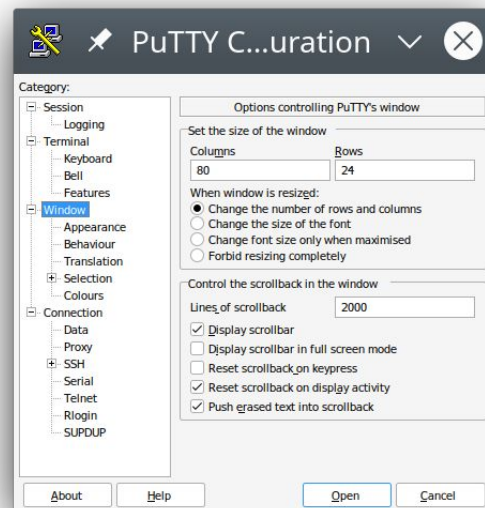


調整 PuTTY 畫面設定

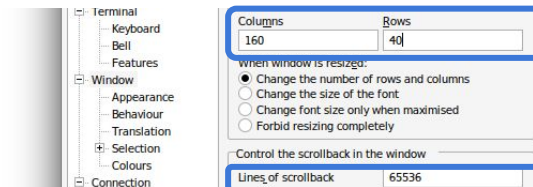
1. 開啓PuTTY，載入想修改的連線設定



2. 點選左側的 "Window" 可得如下畫面

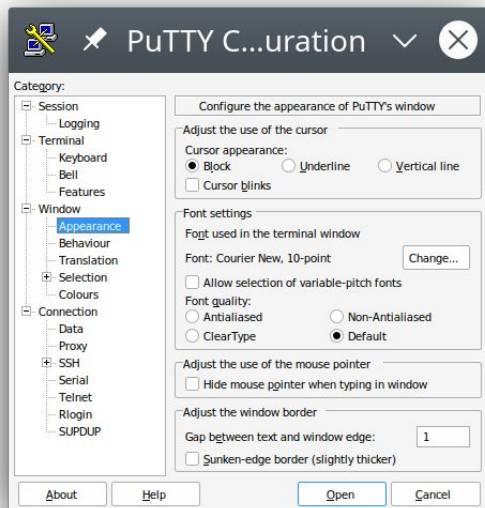


3. 可修改預設畫面大小，以及可向上捲動的行數

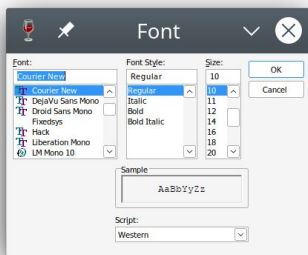


調整 PuTTY 畫面設定

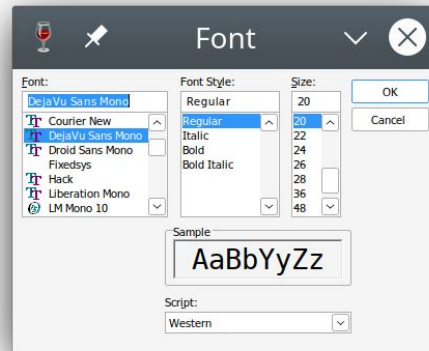
4. 點選左側的“Appearance”可得如下畫面



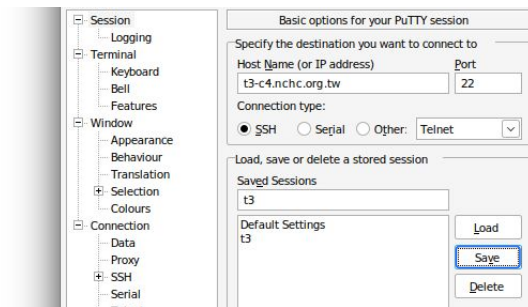
5. 點選上圖右側的“Change”可得如下畫面



6. 選擇想要的字型與大小後，點選“OK”



7. 點選左側的“Session”，再點選“Save”儲存設定



！注意：“Saved Sessions”下方的文字欄位會被當作要儲存的連線名稱，而不是更下方清單所點選的名稱

調整 PuTTY 畫面設定

修改前

```
@lgn304:~
immediately without notification.

* Long-running jobs will be removed.
** We reserve the right to limit access for the users who abuse the system.

=====
Block Limits
=====
Filesystem type blocks quota limit in_doubt grace | #
les quota limit in_doubt grace Remarks
fs01 USR 208.5G 1.465T 1.562T 0 none | 24
fs02 0 0 0 none NCHC_AICls.twcc.ai
fs02 USR 76.72G 100G 200G 0 none | 779
fs02 0 0 222396 none NCHC_AICls.twcc.ai

File Limits
Filesystem Fileset type blocks quota limit in_doubt gr
ace | files quota limit in_doubt grace Remarks
SKstaging biology USR 2.337T 10T 10T 0 n
one | 188725 0 0 none NCHC_AICls.twcc.ai
/usr/lpp/mmfs/bin/mmfsquota -u --block-size auto fs01 fs02
/usr/lpp/mmfs/bin/mmfsquota -u --block-size auto SKstaging:biology

lgn304:~
```

修改後

```
@lgn304:~
login as:
Keyboard-interactive authentication prompts from server:
Password:
Changing MOTP: 874230
End of keyboard-interactive prompts from server

Auth MOTP: PASS
Auth Password: PASS

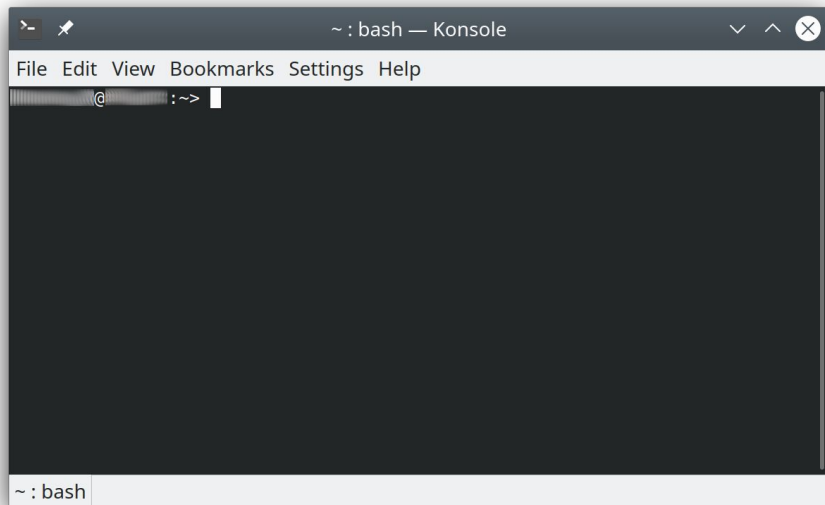
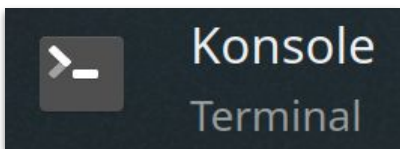
Last login: Thu Jul 7 09:21:55 2022 from 140.110.35.76

-----
Module Command Usage Information
-----
Command | Module Command Description
-----
$ module load [...] | load module(s)
$ module unload [...] | Remove module(s)
$ module purge | unload all modules
$ module list | List loaded modules
$ module avail | List available modules
$ module whatis module | Print whatis information about module
$ module keyword string | Search all name and whatis that contain string

-----
Resource Manager Command Usage Information
-----
Command | SLURM
-----
Submit batch job | sbatch [job script]
Request interactive shell | srun -pty /bin/bash
Delete batch job | scancel [job id]
```


以 Konsole 爲範例連線

1. 點選 Konsole 或是 terminal 來開啓文字命令視窗



2. 以如下指令來連線

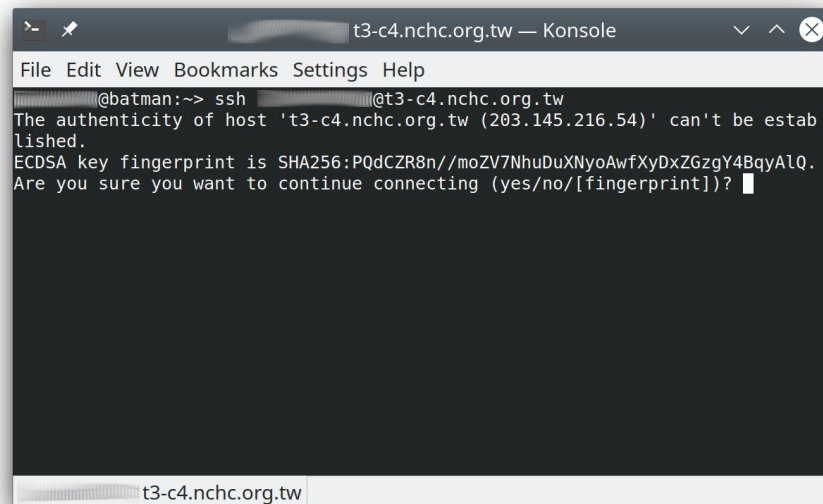
```
ssh [UserName]@[HostName]:[Port]
```

其中 [Port] 預設值爲 22，可不輸入

例：若主機帳號爲 neo1999，則連線指令爲：

```
ssh neo1999@t3-c1.nchc.org.tw
```

3. 第一次會出現如下問題，輸入 yes 即可



以 Konsole 爲範例連線

4. 輸入密碼與 OTP 進行登入，如下畫面爲登入成功

```
t3-c4.nchc.org.tw — Konsole
File Edit View Bookmarks Settings Help

Block Limits
File Limits
Filesystem type blocks quota limit in_doubt grace |
files quota limit in_doubt grace Remarks
fs01 USR 208.5G 1.465T 1.562T 0 none |
26027 0 0 0 none NCHC_AIcls.twcc.ai
fs02 USR 76.72G 100G 200G 0 none |
779132 0 0 2223396 none NCHC_AIcls.twcc.ai
Block Limits
File Limits
Filesystem Fileset type blocks quota limit in_doubt
grace | files quota limit in_doubt grace Remarks
5Kstaging biology USR 2.337T 10T 10T 0
none | 185722 0 0 0 none NCHC_AIcls.twcc.ai
/usr/lpp/mmfs/bin/mmlsquota -u --block-size auto fs01 fs02
/usr/lpp/mmfs/bin/mmlsquota -u --block-size auto 5Kstaging:biolo
gy
@lgn304:~>
t3-c4.nchc.org.tw
```

登入後，初始畫面說明

Last login: Thu Jul 7 15:36:43 2022 from 140.110.89.175 ← 上一次此帳號的登入時間與登入 IP

Module Command Usage Information	
Command	Module Comamnd Description
\$ module load [...]	load module(s)
\$ module unload [...]	Remove module(s)
\$ module purge	unload all modules
\$ module list	List loaded modules
\$ module avail	List available modules
\$ module whatis module	Print whatis information about module
\$ module keyword string	Search all name and whatis that contain string

← module 的指令說明，module 是一個程式執行環境管理套件

Resource Manager Command Usage Information	
Command	SLURM
Submit batch job	sbatch [job script]
Request interactive shell	srun -pty /bin/bash
Delete job	scancel [job id]
Queue status	sinfo
Job status	scontrol show job [job id]
Node status	scontrol show node [NodeID]

SLURM 的指令說明，SLURM 是一個電腦叢集管理系統，用於管理計算節點與工作調度。

Welcome to NCHC 2020 HPC System

1. All jobs, batch or interactive must be submitted to the scheduler.
2. DO NOT run computational jobs on login nodes.
Login nodes are meant for light editing or compilation and for submitting jobs. Any job that runs for more than an hour* or is using significant memory and CPU within an hour should be submitted to Slurm either as interactive or batch job.
3. DO NOT run SLURM query commands CONTINUOUSLY and FREQUENTLY on login nodes. Any execution of interval less than 30 seconds and lasting more than 10 minutes is regarded as a system attack and will be terminated immediately without notification.

* Long-running jobs will be removed.

** We reserve the right to limit access for the users who abuse the system.

臺灣衫三號的使用注意事項 →

臺灣衫三號的使用注意事項

Welcome to NCHC 2020 HPC System

1. All jobs, batch or interactive must be submitted to the scheduler.
 2. DO NOT run computational jobs on login nodes.
Login nodes are meant for light editing or compilation and for submitting jobs. Any job that runs for more than an hour* or is using significant memory and CPU within an hour should be submitted to Slurm either as interactive or batch job.
 3. DO NOT run SLURM query commands CONTINUOUSLY and FREQUENTLY on login nodes. Any execution of interval less than 30 seconds and lasting more than 10 minutes is regarded as a system attack and will be terminated immediately without notification.
- * Long-running jobs will be removed.
** We reserve the right to limit access for the users who abuse the system.

1. 所有的計算工作都必須要送到工作調度器。
2. 不要在登入節點上執行計算工作。登入節點是用於文件編輯、程式編譯、與任務接收。任何要執行超過一小時的工作、使用較多記憶體或是 CPU 資源的工作，都應該送到工作調度器。
3. 不要在登入節點太頻繁且持續地執行 SLURM 的指令。任何執行間隔短於三十秒且持續超過十分鐘的使用方式，會被視為系統攻擊，將會被無預警的終止（可理解為被登出）。

@lgn304:~>timed out waiting for input: auto-logout
Connection to t3-c4.nchc.org.tw closed.

注意事項沒寫的...小事：一段時間沒動作，會被登出。

認識登入節點—文字界面

```
t3-c4.nchc.org.tw — Konsole
File Edit View Bookmarks Settings Help

Block Limits
File Limits
Filesystem type blocks quota limit in_doubt grace |
files quota limit in_doubt grace Remarks
fs01 USR 208.5G 1.465T 1.562T 0 none |
26027 0 0 0 none NCHC_AIcls.twcc.ai
fs02 USR 76.72G 100G 200G 0 none |
779132 0 0 2223396 none NCHC_AIcls.twcc.ai

Block Limits
File Limits
Filesystem Fileset type blocks quota limit in_doubt
grace | files quota limit in_doubt grace Remarks
5Kstaging biology USR 2.337T 10T 10T 0
none | 185722 0 0 0 none NCHC_AIcls.twcc.ai
/usr/lpp/mmfs/bin/mmfsquota -u --block-size auto fs01 fs02
/usr/lpp/mmfs/bin/mmfsquota -u --block-size auto 5Kstaging:biolo
gy

@lgn304:~>
t3-c4.nchc.org.tw
```

命令提示字串

顯示在輸入字串之前的一段訊息，此訊息可自訂。

[UserName]@[HostName]:[WorkingDirectory]>

工作目錄，working directory

預設的資料存取目錄

目錄，directory

一份位置清單，記錄目錄位置與檔案位置，是前往目標目錄或目標檔案的中轉站。

Desktop/
Documents/
Downloads/
Music/
Pictures/
note.txt
alphafold
...

根目錄，root directory

前往所有目錄與檔案的起點

家目錄，home directory

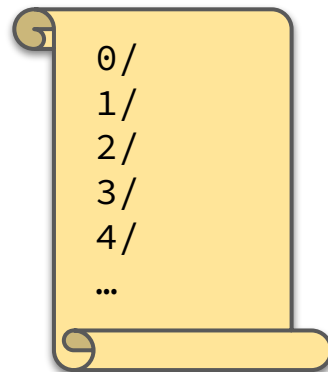
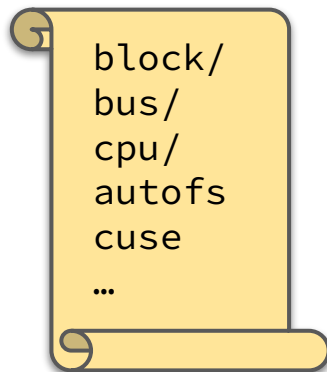
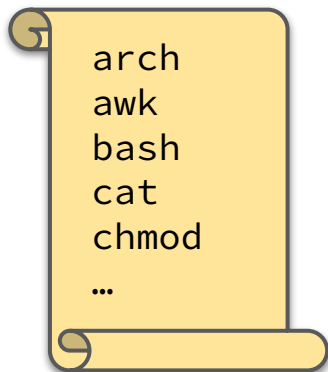
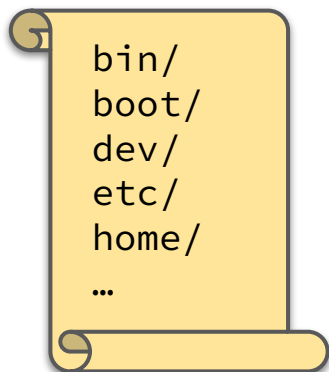
使用者登入後的預設工作目錄。通常，帳號與該帳號的家目錄是一對一，並且該帳號對於該目錄有完整的使用權限，包括讀、寫、與執行。

~ 為家目錄路徑的代號。

例，帳號名稱爲 neo1999，則通常對於 neo1999 來說，~ 就是 /home/neo1999。

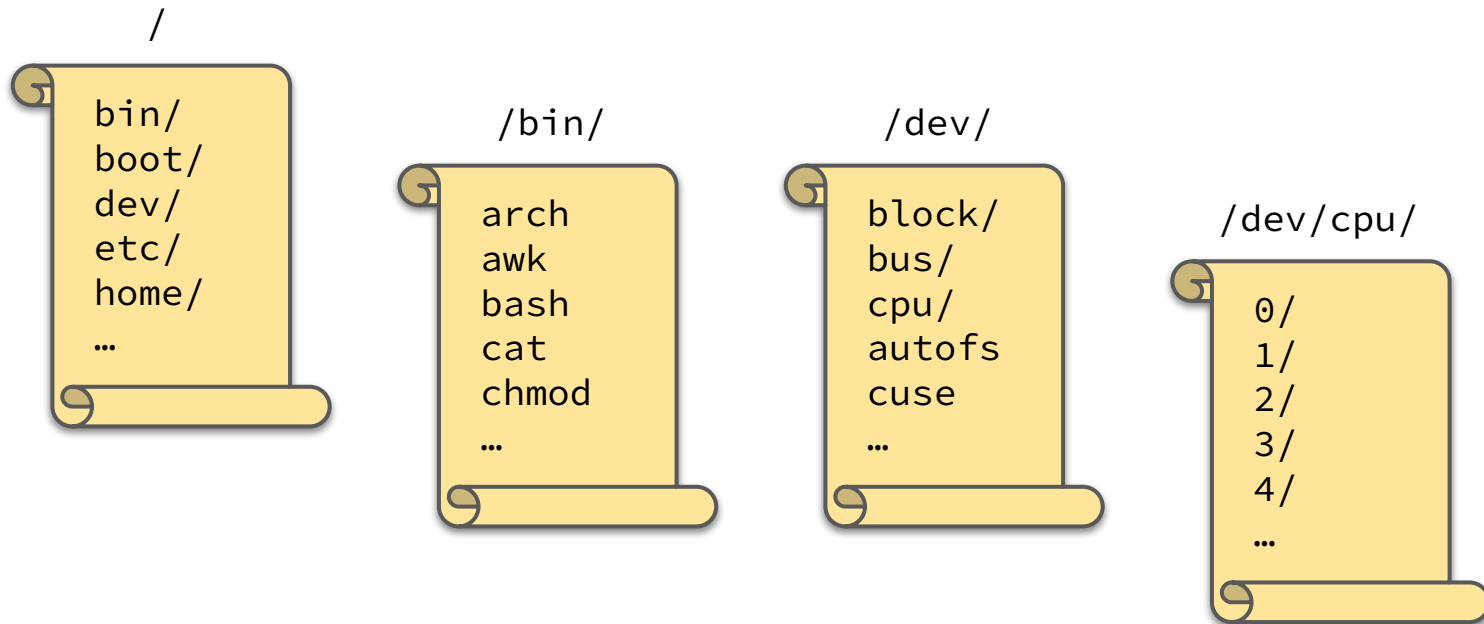
Linux 目錄配置

目錄是一份位置清單，記錄目錄位置與檔案位置。



Linux 目錄配置

所有的目錄與檔案的路徑，都是從根目錄(/)開始。



絕對路徑：從根目錄(/)開始寫起的路徑。例：

`/dev/disk/by-id/nvme-PC_SN730_NVMe_WDC_1024GB_210536803623-part2`

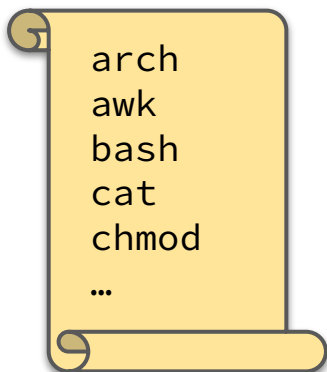
相對路徑：不從根目錄(/)開始寫起的路徑，相對於工作目錄。例：若工作目錄為/dev/disk/，則上例的相對路徑可以寫為

`by-id/nvme-PC_SN730_NVMe_WDC_1024GB_210536803623-part2`

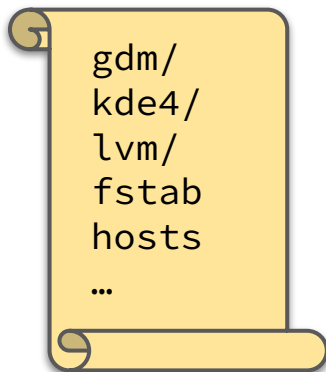
Linux 目錄配置

Linux 目錄配置標準：Filesystem Hierarchy Standard (FHS)

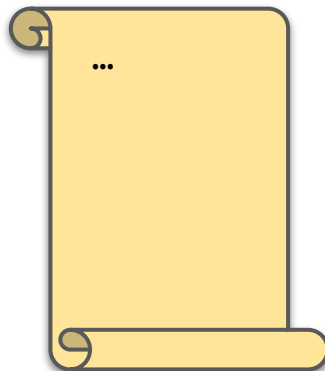
/bin/ 存放一般使用者也能使用的最基本指令。



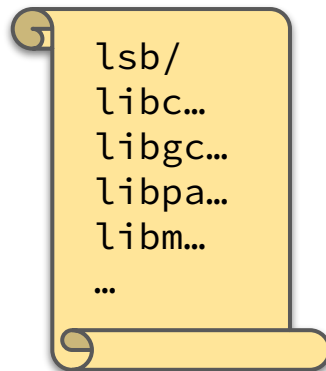
/etc/ 存放系統設定資料。



/home/ 存放一般使用者的家目錄資料。



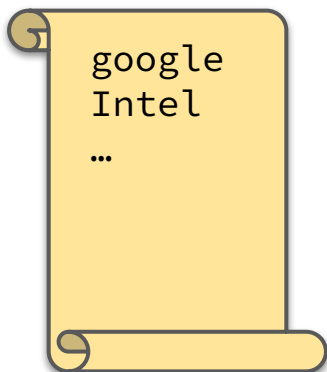
/lib/ 存放函式庫



Linux 目錄配置

Linux 目錄配置標準：Filesystem Hierarchy Standard (FHS)

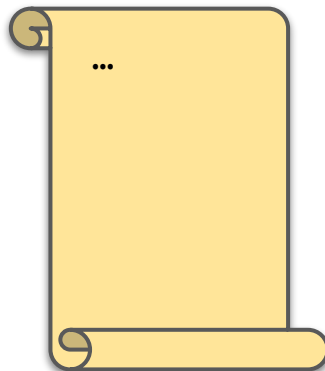
/opt/ 存放非原本 distribution 提供的軟體。



/proc/ 存放目前系統資訊。



/root/ 最大權限使用者 root 的家目錄。



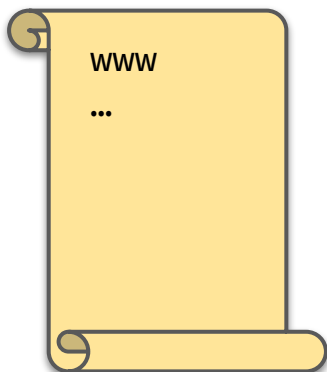
/sbin/ root 才能執行的程式



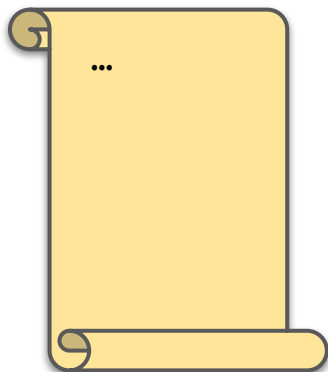
Linux 目錄配置

Linux 目錄配置標準：Filesystem Hierarchy Standard (FHS)

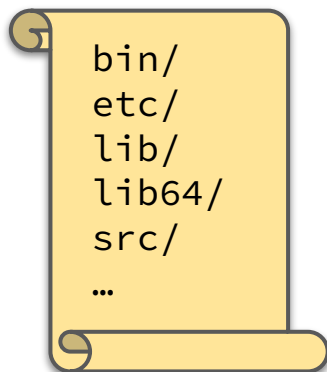
/srv/ 存放網路服務資料。



/tmp/ 存放暫存資料。

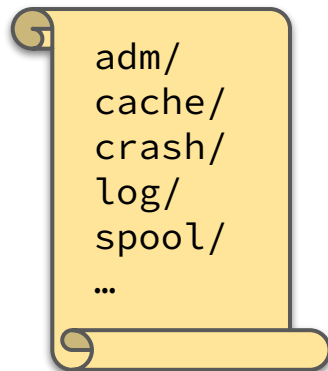


/usr/ 存放 distribution 安裝的軟體。



Unix Software Resource (usr)

/var/ 存放程式記錄檔與執行時資訊



認識登入節點—目錄配置

```
@lgn304:~>pwd
/home/
```

查詢工作目錄的路徑 → `pwd`

```
@lgn304:~>ls
20220426_tf2053py37s114824.yml  all_queuestat_temp  stgBio
bin                               queue_name_temp     work
fasta                             settings
```

顯示目錄的內容 → `ls`

`ls` 例 1：預設是顯示工作目錄的內容

```
@lgn304:~>ls /
bin  gpfs  media  opt  sbin  staging2  usr
boot home  misc  proc  scratch  starterkit  var
dev  lib  mnt  root  srv  sys  work
etc  lib64  net  run  staging  tmp  xcatpost
```

顏色設定？意義？

有些非預設的目錄？

`ls` 例 2：顯示根目錄的內容

```
ls /
```

Linux 環境配置— Bash Shell

```
@lgn301:~>env
XDG_SESSION_ID=132919
HOSTNAME=lgn301.nchc-2020-hpc
LMOD_REF_COUNT MODULEPATH=/opt/ohpc/Taiwania3/modulefiles:1
TERM=xterm-256color
SHELL=/bin/bash
ORI_LS_COLORS=no=00:fi=00:di=01;34:ln=00;4:pi=40;33:so=01;35:do=01;35:bd=40;
33;01:cd=40;33;01:or=41;33;01:ex=00;32:*.cmd=00;32:*.exe=01;32:*.com=01;32:*.bat=01;32:*.btm=01;32:*.dll=01;32:*.tar=00;31:*.tbz=00;31:*.tgz=00;31:*.rpm=00;31:*.deb=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.lzma=00;31:*.zip=00;31:*.zoo=00;31:*.z=00;31:*.Z=00;31:*.gz=00;31:*.bz2=00;31:*.tb2=00;31:*.tz2=00;31:*.tbz2=00;31:*.xz=00;31:*.avi=01;35:*.bmp=01;35:*.dl=01;35:*.fli=01;35:*.gif=01;35:*.gl=01;35:*.jpg=01;35:*.jpeg=01;35:*.mkv=01;35:*.mng=01;35:*.mov=01;35:*.mp4=01;35:*.mpg=01;35:*.pcx=01;35:*.pbm=01;35:*.pgm=01;35:*.png=
```

印出環境變數 → `env`

環境變數說明

- 用於設定部分指令的呈現方式
- 記錄使用者此次連線相關訊息
- 記錄部分主機訊息
- 記錄搜尋指令與函式庫的目錄順序
- 記錄自訂指令

命令提示字串 PS1

```
export PS1="\u@\h:\w>"
```

目錄指令顏色設定 LS_COLORS

```
export
LS_COLORS="no=00:fi=00:di=01;34:ln=36;4:pi=40;33:so=01;
35:do=01;35:bd=40;33;01:cd=40;33;01:or=41;33;01:ex=00;3
2:*.cmd=00;32:*.exe=01;32:*.com=01;32:*.bat=01;32:*.btm
=01;32:*.dll=01;32:*.tar=00;31:*.tbz=00;31:*.tgz=00;31:
*.rpm=00;31:*.deb=00;31:*.arj=00;31:*.taz=00;31:*.lzh=0
0;31:*.lzma=00;31:*.zip=00;31:*.zoo=00;31:*.z=00;31:*.Z
=00;31:*.gz=00;31:*.bz2=00;31:*.tb2=00;31:*.tz2=00;31:*.
tbz2=00;31:*.xz=00;31:*.avi=01;35:*.bmp=01;35:*.dl=01;
35:*.fli=01;35:*.gif=01;35:*.gl=01;35:*.jpg=01;35:*.jpe
g=01;35:*.mkv=01;35:*.mng=01;35:*.mov=01;35:*.mp4=01;35
:*.mpg=01;35:*.pcx=01;35:*.pbm=01;35:*.pgm=01;35:*.png=
01;35:*.ppm=01;35:*.svg=01;35:*.tga=01;35:*.tif=01;35:*.
webm=01;35:*.webp=01;35:*.wmv=01;35:*.xbm=01;35:*.xcf=
01;35:*.xpm=01;35:*.aiff=00;32:*.ape=00;32:*.au=00;32:*.
flac=00;32:*.m4a=00;32:*.mid=00;32:*.mp3=00;32:*.mpc=0
0;32:*.ogg=00;32:*.voc=00;32:*.wav=00;32:*.wma=00;32:*.
wv=00;32:"
```

Linux 環境配置— Bash Shell

語言顯示設定

```
#LANG=en_US.UTF-8
LANG=zh_TW.UTF-8
LC_CTYPE=$LANG
LC_NUMERIC=$LANG
LC_TIME=$LANG
LC_COLLATE=$LANG
LC_MONETARY=$LANG
LC_MESSAGES=$LANG
LC_PAPER=$LANG
LC_NAME=$LANG
LC_ADDRESS=$LANG
LC_TELEPHONE=$LANG
LC_MEASUREMENT=$LANG
LC_IDENTIFICATION=$LANG
LC_ALL=$LANG
```

指令別名 → alias

印出目前別名

```
alias
```

設定別名

```
alias o='less'
alias grep='grep --color=auto'
alias l='ls -alF'
```

印出字串 → echo

```
echo "YABE! Gura is so cute!"
echo "$PS1"
echo "我的帳號是 $USER"
echo "指令搜尋路徑是 : \"$PATH\""
echo '函式庫搜尋路徑是 : "$LD_LIBRARY_PATH"'
```

認識登入節點—目錄配置

```
@lgn304:~>pwd
/home/
```

查詢工作目錄的路徑 → `pwd`

```
@lgn304:~>ls
20220426_tf2053py37s114824.yml  all_queuestat_temp  stgBio
bin                               queue_name_temp     work
fasta                             settings
```

顯示目錄的內容 → `ls`

`ls` 例 1：預設是顯示工作目錄的內容

```
@lgn304:~>ls /
bin    gpfs    media  opt    sbin    staging2  usr
boot  home   misc   proc   scratch starterkit var
dev    lib    mnt    root   srv     sys      work
etc    lib64  net    run    staging tmp      xcatpost
```

顏色設定？意義？

有些非預設的目錄？

`ls` 例 2：顯示根目錄的內容

```
ls /
```

認識登入節點—硬碟空間

```
@lgn301:~>/usr/lpp/mmfs/bin/mmlsquota -u --block-size auto fs01 fs02
```

Block Limits							File Limits					
Filesystem	type	blocks	quota	limit	in_doubt	grace	files	quota	limit	in_doubt	grace	Remarks
fs01	USR	208.5G	1.465T	1.562T	0	none	26032	0	0	0	none	NCHC_AIcls.twcc.ai
fs02	USR	92.34G	100G	200G	0	none	1077534	0	0	2223396	none	NCHC_AIcls.twcc.ai

臺灣衫三號使用者硬碟空間配額說明

- 每位使用者的家目錄的空間上限為 100 G。
- 每位使用者的 `/work/[UserName]` 的空間上限為 1.5 T。
- 可使用以下指令來查詢

```
: /usr/lpp/mmfs/bin/mmlsquota -u  
[UserName] --block-size auto fs01  
fs02
```

其中 fs01 為 `/work`，而 fs02 為 `/home`，當 `/work` 空間不足時，會清除長期未被使用的資料。

```
@lgn301:~>df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	189G	4.0K	189G	1%	/dev
tmpfs	189G	99M	189G	1%	/dev/shm
tmpfs	189G	2.7G	186G	2%	/run
tmpfs	189G	0	189G	0%	/sys/fs/cgroup
/dev/mapper/xcatvg-root	442G	23G	420G	6%	/
/dev/sda2	506M	144M	362M	29%	/boot
/dev/sda1	50M	11M	39M	23%	/boot/efi
fs02	1.8P	852T	926T	48%	/home
fs01	7.6P	2.5P	5.1P	33%	/work

Linux 顯示硬碟空間指令 → **df**

df 例 1：以易讀的方式顯示被掛載的硬碟空間

```
df -h
```

認識登入節點—檔案資訊

```
@lgn304:~>ls -l /
total 2888
lrwxrwxrwx. 1 root root 7 Oct 8 2020 bin -> usr/bin
dr-xr-xr-x. 6 root root 4096 Apr 21 2021 boot
drwxr-xr-x. 22 root root 6340 Apr 21 11:35 dev
drwxr-xr-x. 161 root root 12288 Jun 20 14:34 etc
drwxr-xr-x 3 root root 26 Jan 1 2021 gpfs
drwxr-xr-x 9477 root root 524288 Jul 11 21:00 home
lrwxrwxrwx. 1 root root 7 Oct 8 2020 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Oct 8 2020 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 Apr 11 2018 media
dr-xr-xr-x 2 root root 6 Nov 9 2020 misc
drwxr-xr-x. 2 root root 6 Apr 11 2018 mnt
dr-xr-xr-x 2 root root 6 Nov 9 2020 net
drwxr-xr-x. 13 root root 196 Jan 7 2022 opt
dr-xr-xr-x 2900 root root 0 Jul 17 2021 proc
dr-xr-xr-x. 24 root root 4096 Jul 4 17:19 root
drwxr-xr-x 50 root root 1680 Jul 10 03:17 run
lrwxrwxrwx. 1 root root 8 Oct 8 2020/sbin -> usr/sbin
drwxrwxrwt 3 root root 15 May 31 16:27 scratch
drwxr-xr-x. 2 root root 6 Apr 11 2018 srv
drwxr-xr-x 9 root root 262144 Jul 8 23:31 staging
drwxr-xr-x 7 root root 262144 May 16 13:53 staging2
dr-xr-xr-x 2 root root 6 Oct 12 2020 starterkit
dr-xr-xr-x 13 root root 0 Aug 8 2021 sys
drwxrwxrwt. 230 root root 90112 Jul 12 09:06 tmp
drwxr-xr-x. 15 root root 177 Dec 22 2020 usr
drwxr-xr-x. 23 root root 4096 Aug 8 2021 var
drwxr-xr-x 9391 root root 524288 Jul 11 21:05 work
drwxr-xr-x 7 root root 4096 Nov 2 2020 xcatpost

@lgn304:~>
```

```
lrwxrwxrwx. 1 root root 7 Oct 8 2020 bin -> usr/bin
dr-xr-xr-x. 6 root root 4096 Apr 21 2021 boot
drwxr-xr-x 22 root root 6340 Apr 21 11:35 dev
drwxr-xr-x. 161 root root 12288 Jun 20 14:34 etc
```

[權限] [硬連結數量] [擁有者] [群組] [大小] [修改日期] [名稱] [連結資訊]

ls 例 3：顯示例 2 的詳細內容

```
ls -l /
```


Linux 檔案權限

```
lrwxrwxrwx. 1 root root 7 Oct 8 2020 bin -> usr/bin
dr-xr-xr-x. 6 root root 4096 Apr 21 2021 boot
drwxr-xr-x 22 root root 6340 Apr 21 11:35 dev
drwxr-xr-x. 161 root root 12288 Jun 20 14:34 etc
```

[權限] [硬連結數量] [擁有者] [群組] [大小] [修改日期] [名稱] [連結資訊]

權限說明

第一個字元：**類型**

第二到第四個字元：**擁有者的權限**

第五到第七個字元：**群組的權限**

第八到第十個字元：**其他人的權限**

第十一個字元：**額外的權限類型**

```
lrwxrwxrwx.
dr-xr-xr-x.
```

1 2~4 5~7 8~10 11

類型

- l：連結
- d：目錄
- ：檔案

擁有者的權限 > **群組的權限** > **其他人的權限**

- r：可被讀取
- w：可被寫入
- x：檔案可被執行，目錄可被顯示
- ：不具有對應位置上的權限

！注意：實際要執行檔案或顯示目錄要同時有 **r** 與 **x** 的權限

關於擁有者

一定是某位使用者，只有 root 與擁有者可以改變該目錄或是檔案的權限

關於群組

每一位使用者都至少屬於一個群組，一個群組可以有數位使有者。爲了方便資料分享而設計。

額外的權限類型

- .：Security-Enhanced Linux (SELinux) 權限設定
- +：其他的權限設定，例如：Access Control List (ACL)

Linux 隱藏檔

```
@lgn304:~>ls -lh /
total 2.9M
lrwxrwxrwx. 1 root root 7 Oct 8 2020 bin -> usr/bin
dr-xr-xr-x. 6 root root 4.0K Apr 21 2021 boot
drwxr-xr-x 22 root root 6.2K Apr 21 11:35 dev
drwxr-xr-x. 161 root root 12K Jun 20 14:34 etc
drwxr-xr-x 3 root root 26 Jan 1 2021 gpfs
drwxr-xr-x 9477 root root 512K Jul 11 21:00 home
lrwxrwxrwx. 1 root root 7 Oct 8 2020 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Oct 8 2020 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 Apr 11 2018 media
dr-xr-xr-x 2 root root 6 Nov 9 2020 misc
drwxr-xr-x. 2 root root 6 Apr 11 2018 mnt
dr-xr-xr-x 2 root root 6 Nov 9 2022 net
drwxr-xr-x. 13 root root 196 Jan 7 2022 opt
dr-xr-xr-x 2918 root root 0 Jul 17 2021 proc
dr-xr-xr-x. 24 root root 4.0K Jul 4 17:19 root
drwxr-xr-x 50 root root 1.7K Jul 10 03:17 run
lrwxrwxrwx. 1 root root 8 Oct 8 2020/sbin -> usr/sbin
drwxrwxrwt 3 root root 15 May 31 16:27 scratch
drwxr-xr-x. 2 root root 6 Apr 11 2018 srv
drwxr-xr-x 9 root root 256K Jul 8 23:31 staging
drwxr-xr-x 7 root root 256K May 16 13:53 staging2
dr-xr-xr-x 2 root root 6 Oct 12 2020 starterkit
dr-xr-xr-x 13 root root 0 Aug 8 2021 sys
drwxrwxrwt. 231 root root 88K Jul 12 09:08 tmp
drwxr-xr-x. 15 root root 177 Dec 22 2020 usr
drwxr-xr-x. 23 root root 4.0K Aug 8 2021 var
drwxr-xr-x 9391 root root 512K Jul 11 21:05 work
drwxr-xr-x 7 root root 4.0K Nov 2 2020 xcatpost
@lgn304:~>
```

ls 例 4：較例 3 更容易讀的檔案大小資訊

```
ls -lh /
```

```
@lgn304:~>ls -lha /
total 3.0M
dr-xr-xr-x. 26 root root 4.0K May 6 16:05 .
dr-xr-xr-x. 26 root root 4.0K May 6 16:05 ..
-rw-r--r-- 1 root root 0 Oct 8 2020 .autorelabel
lrwxrwxrwx. 1 root root 7 Oct 8 2020 bin -> usr/bin
dr-xr-xr-x. 6 root root 4.0K Apr 21 2021 boot
drwxr-xr-x 22 root root 6.2K Apr 21 11:35 dev
drwxr-xr-x. 161 root root 12K Jun 20 14:34 etc
drwxr-xr-x 3 root root 26 Jan 1 2021 gpfs
drwxr-xr-x 9477 root root 512K Jul 11 21:00 home
lrwxrwxrwx. 1 root root 7 Oct 8 2020 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Oct 8 2020 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 Apr 11 2018 media
dr-xr-xr-x 2 root root 6 Nov 9 2020 misc
drwxr-xr-x. 2 root root 6 Apr 11 2018 mnt
dr-xr-xr-x 2 root root 6 Nov 9 2020 net
```

ls 例 5：顯示例 4 中沒有顯示的隱藏檔與隱藏目錄

```
ls -lha /
```

隱藏檔

以點 `.` 為名稱開頭的檔案

特殊隱藏檔

`.`：代表此目錄

`..`：代表上一層目錄

小提示：可以自己測試一下

```
ls . 與 ls ..
```

認識登入節點— CPU

```
@lgn304:~>ls /proc
1      1385   163257 196    224852 263337 39458 70
10     1389   164     1962   225    263338 39467 70406
100    139    164109 1964   225121 263339 39474 70509
10006  139137 164304 196401 225122 263342 39480 70584
10009  139352 165    196403 225123 263343 39509 706
100116 139353 1652   196630 225124 263344 39513 708
10029  13944  165258 196633 225326 263345 39519 70829
10040  1395   1653   196722 225424 263949 39524 70830
10057  13953  1657   196733 225425 26398 39530 709
10057  13953  1657   196733 225425 26398 39530 709
133    155370 191486 220768 256433 37441 638
133060 155739 191491 220787 256436 3745 63823
13311  155789 1915    220792 256441 37469 639
133370 156    191503 220795 256445 3748 64
133606 156139 191514 220807 256453 37580 640
133607 156140 191516 220813 256461 37581 641
137341 162398 195353 223881 26280 39280 68832
137342 162402 195    223925 263    39340 69
1374   162423 195368 224    263203 39397 69257
1375   162429 195472 224033 263231 39424 69279
1378   162751 195533 22404  263232 39429 69332
1379   162924 195756 22412  26332  39432 6968
138    163    195821 22417  26333  39444 7
@lgn304:~>
```

ls 例 6：顯示指定路徑的目錄資訊

```
ls /proc
```

```
@lgn304:~>ls /proc/cpuinfo
/proc/cpuinfo
@lgn304:~>
```

ls 例 7：顯示指定路徑的檔案資訊

```
ls /proc/cpuinfo
```

```
@lgn304:~>ls /proc/c*
/proc/cgroups /proc/cmdline /proc/consoles /proc/cpuinfo /proc/crypto
@lgn304:~>
```

ls 例 8：顯示 /proc 中以 c 開頭的目錄與檔案

```
ls /proc/c*
```

萬用字元 (Wildcard characters)

*****：存在大於等於 0 個字元

?：存在 1 個字元

認識登入節點— CPU

```
@lgn304:~>less /proc/cpuinfo
```

顯示檔案內容 → less

less 例 1：顯示 /proc/cpuinfo 的內容

```
less /proc/cpuinfo
```

```
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 85
model name    : Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GHz
stepping      : 7
microcode     : 0x500002c
cpu MHz       : 2700.000
cache size    : 39424 KB
physical id   : 0
siblings      : 28
core id       : 0
cpu cores     : 28
apicid        : 0
initial apicid : 0
fpu           : yes
fpu_exception : yes
cpuinfo
```

在 less 的畫面中，可以使用鍵盤方向鍵來上下左右捲動，按 q 可以離開

/proc/cpuinfo 有即時的 CPU 資訊，可以知道 CPU 型號、時脈、與支援的功能 (flags)

```
fpu_exception : yes
cpuid level   : 22
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca c
mov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdp
e1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopolog
y nonstop_tsc aperfmperf eagerfpu pni pclmuldq dtes64 monitor ds_cpl vmx sm
x est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic movbe
popcnt tsc deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefet
h epb cat_l3 cdp_l3 invpcid_single intel_ppin intel_pt ssbd mba ibrs ibpb st
ibp ibrs_enhanced tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust
bmi1 hle avx2 smep bmi2 erms invpcid rtm cqm mpx rdt_a avx512f avx512dq rdse
ed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetb
v1 cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local dtherm ida arat pln pts
hwp epp pku ospke avx512_vnni md_clear spec_ctrl intel_stibp flush_lld arch
capabilities
bogomips      : 5400.00
```

less 例 2：顯示例 1 檔案的行號

```
less -N /proc/cpuinfo
```

```
19 wp            : yes
20 flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr p
20 ge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pb
20 e syscall nx pdp1gb rdtscp lm constant_tsc art arch_perfmon pebs bt
20 s rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmul
20 dq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr p
20 dcm pcid dca sse4_1 sse4_2 x2apic movbe popcnt tsc deadline timer ae
20 s xsave avx f16c rdrand lahf_lm abm 3dnowprefetch epb cat_l3 cdp_l3
20 invpcid_single intel_ppin intel_pt ssbd mba ibrs ibpb stibp ibrs_enh
20 anced tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1
20 hle avx2 smep bmi2 erms invpcid rtm cqm mpx rdt_a avx512f avx512dq
20 rdseed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt
20 xsavec xgetbv1 cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local dth
20 erm ida arat pln pts hwp epp pku ospke avx512_vnni md_clear spec_ctr
20 l intel_stibp flush_lld arch_capabilities
21 bogomips      : 5400.00
```

Bash Shell 指令：變更工作目錄

```
@lgn304:~>cd /
@lgn304:/>ls
bin    gpfs    media   opt     sbin    staging2  usr
boot   home    misc    proc    scratch starterkit var
dev    lib     mnt     root    srv      sys      work
etc    lib64   net     run     staging  tmp      xcatpost
```

變更工作目錄 → `cd`

cd 例 1：變更工作目錄為根目錄

```
cd /
```

```
@lgn304:/proc>cd
@lgn304:~>cd ~
@lgn304:~>
```

cd 例 2：變更工作目錄為家目錄

```
cd ~
```

或

```
cd
```

```
@lgn304:/>cd proc
@lgn304:/proc>
```

cd 例 3：變更工作目錄為 `/proc`

```
cd /
```

```
cd proc
```

或

```
cd /; cd proc
```

或

```
cd / && cd proc
```

或是

```
cd /proc
```


Bash Shell 指令：執行控制

後面指令是否執行的判斷： **&&** 與 **||**

前指令 **&&** 後指令：若前指令執行**正確**，則後指令就接着執行。

前指令 **||** 後指令：若前指令執行**有誤**，則後指令就接着執行。

```
@lgn304:/proc>ls mem* && pwd && uname -a
meminfo
/proc
Linux lgn304.nchc-2020-hpc 3.10.0-1127.el7.x86_64 #1 SMP Tue Mar 31 23:36:51
UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

```
@lgn304:/proc>ls memo* && pwd && uname -a
ls: cannot access memo*: No such file or directory
```

```
@lgn304:/proc>ls memo* || pwd && uname -a
ls: cannot access memo*: No such file or directory
/proc
Linux lgn304.nchc-2020-hpc 3.10.0-1127.el7.x86_64 #1 SMP Tue Mar 31 23:36:51
UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

Bash Shell 操作：執行控制

後面指令是否執行的判斷：**&&** 與 **||**

前指令 **&&** 後指令：若前指令執行**正確**，則後指令就接着執行。

前指令 **||** 後指令：若前指令執行**有誤**，則後指令就接着執行。

```
@lgn304:/proc>ls memo* || pwd || uname -a  
ls: cannot access memo*: No such file or directory  
/proc
```

```
@lgn304:/proc>ls memo* && pwd || uname -a  
ls: cannot access memo*: No such file or directory  
Linux lgn304.nchc-2020-hpc 3.10.0-1127.el7.x86_64 #1 SMP Tue Mar 31 23:36:51  
UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

認識登入節點— OS

顯示系統資訊 → `uname -a`

顯示系統資訊 → `cat /etc/*-release`

```
@batman:~> cat /etc/*-release
NAME="openSUSE Leap"
VERSION="15.3"
ID="opensuse-leap"
ID_LIKE="suse opensuse"
VERSION_ID="15.3"
PRETTY_NAME="openSUSE Leap 15.3"
ANSI_COLOR="0;32"
CPE_NAME="cpe:/o:opensuse:leap:15.3"
BUG_REPORT_URL="https://bugs.opensuse.org"
HOME_URL="https://www.opensuse.org/"
```


認識登入節點— CPU & Memory

@lgn304: ~->top

顯示系統資源使用狀況 → top

top 例 1：不加參數，使用預設值

top

```
top - 14:04:23 up 360 days, 5:46, 71 users, load average: 2.88, 2.85, 2.70
Tasks: 2993 total, 1 running, 2956 sleeping, 12 stopped, 24 zombie
%Cpu(s): 1.6 us, 1.0 sy, 0.0 ni, 97.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0
KiB Mem : 39465363+total, 27438755+free, 10929528+used, 10970812 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 27795033+avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
74924		20	0	131716	9860	1516	S	14.4	0.0	13:07.83
83017		20	0	795928	42028	4736	S	11.4	0.0	8923:23
1346		20	0	398608	7576	1120	S	9.2	0.0	75762:36
26059		20	0	1266088	3036	576	S	4.9	0.0	10461:55
250947		20	0	1466364	1.1g	1156	S	4.6	0.3	5732:05
141778		20	0	169416	102168	2328	S	3.9	0.0	9922:29
254344		20	0	332444	74832	3552	S	3.9	0.0	0:08.91
254331		20	0	332444	74832	3552	S	3.6	0.0	0:08.89
254333		20	0	332444	74832	3552	S	3.6	0.0	0:09.27
141817		0	-20	50.5g	23.2g	4.9g	S	3.3	6.2	137761:07
216859		20	0	747940	69636	4	S	3.3	0.0	205:10.24

第一列：目前時間，開機了多長時間，使用者數，1分鐘內，五分鐘內，十五分鐘內的負載

第二列，程序數量：全部，在執行的，在休息等待的，暫停的，當掉的

第三列，CPU 使用比例：執行，系統操作，最佳化運作，休息，等待資料讀寫(I/O)，硬體中斷，軟體中斷，虛擬機等待

第四列，記憶體使用量：總量，沒被佔用，程式使用中，資料快取

第五列，Swap（虛擬記憶體）：總量，沒被佔用，程式使用中，真正可用的記憶體量

認識登入節點— CPU & Memory

@lgn304: ~>top

顯示系統資源使用狀況 → top

top 例 1：不加參數，使用預設值

top

```
top - 14:04:23 up 360 days, 5:46, 71 users, load average: 2.88, 2.85, 2.70
Tasks: 2993 total, 1 running, 2956 sleeping, 12 stopped, 24 zombie
%Cpu(s): 1.6 us, 1.0 sy, 0.0 ni, 97.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0
KiB Mem : 39465363+total, 27438755+free, 10929528+used, 10970812 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 27795033+avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
74924		20	0	131716	9860	1516	S	14.4	0.0	13:07.83
83017		20	0	795928	42028	4736	S	11.4	0.0	8923:23
1346		20	0	398608	7576	1120	S	9.2	0.0	75762:36
26059		20	0	1266088	3036	576	S	4.9	0.0	10461:55
250947		20	0	1466364	1.1g	1156	S	4.6	0.3	5732:05
141778		20	0	169416	102168	2328	S	3.9	0.0	9922:29
254344		20	0	332444	74832	3552	S	3.9	0.0	0:08.91
254331		20	0	332444	74832	3552	S	3.6	0.0	0:08.89
254333		20	0	332444	74832	3552	S	3.6	0.0	0:09.27
141817		0	-20	50.5g	23.2g	4.9g	S	3.3	6.2	137761:07
216859		20	0	747940	69636	4	S	3.3	0.0	205:10.24

PID：程序被分配到的ID

USER：執行該程序的主機帳號

PR：動態優先級，越低越搶資源

NI：靜態優先級，越低越搶資源

VIRT：虛擬記憶體使用，並非真正的使用量

RES：真正的主記憶體使用量

SHR：程序共享的主記憶體使用量

S：狀態，R (執行)，S (休息)，Z (當掉)

%CPU：CPU 使用百分比，一個核心為 100

%MEM：記憶體使用百分比

TIME+：累計的 CPU 佔用時間
乘上使用的核心數

認識登入節點— CPU & Memory

@lgn304: ~->top

顯示系統資源使用狀況 → top

top 例 1：不加參數，使用預設值

top

```
top - 14:04:23 up 360 days, 5:46, 71 users, load average: 2.88, 2.85, 2.70
Tasks: 2993 total, 1 running, 2956 sleeping, 12 stopped, 24 zombie
%Cpu(s): 1.6 us, 1.0 sy, 0.0 ni, 97.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0
KiB Mem : 39465363+total, 27438755+free, 10929528+used, 10970812 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 27795033+avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
74924		20	0	131716	9860	1516	S	14.4	0.0	13:07.83
83017		20	0	795928	42028	4736	S	11.4	0.0	8923:23
1346		20	0	398608	7576	1120	S	9.2	0.0	75762:36
26059		20	0	1266088	3036	576	S	4.9	0.0	10461:55
250947		20	0	1466364	1.1g	1156	S	4.6	0.3	5732:05
141778		20	0	169416	102168	2328	S	3.9	0.0	9922:29
254344		20	0	332444	74832	3552	S	3.9	0.0	0:08.91
254331		20	0	332444	74832	3552	S	3.6	0.0	0:08.89
254333		20	0	332444	74832	3552	S	3.6	0.0	0:09.27
141817		0	-20	50.5g	23.2g	4.9g	S	3.3	6.2	137761:07
216859		20	0	747940	69636	4	S	3.3	0.0	205:10.24

此畫面按如下按鍵，則會...

d：更改畫面更新時間，預設 3 秒

k：移除一個程序，使之不再繼續佔用任何資源

u：輸入主機帳號，使 top 只顯示該帳號的程序

q：離開

e：?

E：?

l：?

m：?

h：?

認識登入節點— CPU & Memory

```
sanskrit007@lgn304:~>top -o %MEM
```

顯示系統資源使用狀況 → **top**

top 例 2：指定排序欄位為 %MEM

```
top -o %MEM
```

```
top - 15:05:00 up 360 days, 6:47, 68 users, load average: 6.97, 6.90, 5.56
Tasks: 3015 total, 4 running, 2974 sleeping, 13 stopped, 24 zombie
%Cpu(s): 4.8 us, 2.5 sy, 0.0 ni, 91.0 id, 1.6 wa, 0.0 hi, 0.1 si, 0.0
KiB Mem : 39465363+total, 36008672 free, 29271372+used, 65931232 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 94580840 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
137966	nanosam+	20	0	174.3g	174.2g	1200	R	99.7	46.3	25:37.51
141817	root	0	-20	50.5g	23.2g	4.9g	S	6.9	6.2	137767:54
256461	polkitd	20	0	7743116	6.8g	968	S	0.0	1.8	184:18.40
236194	yesild6+	20	0	75.5g	4.9g	476296	S	0.3	1.3	130:04.70
47593	jeff424+	20	0	18.7g	4.1g	1800	S	0.7	1.1	1448:35
222107	nanosam+	20	0	4074056	3.8g	0	S	0.0	1.0	0:25.83
120974	u1272905	20	0	13.4g	3.0g	996	S	0.3	0.8	1461:46
1436	nsd	20	0	5772548	2.9g	6776	S	0.0	0.8	660:14.03
143454	biology	20	0	3280180	2.7g	776	S	0.0	0.7	466:38.15
133060	biology1	20	0	239.2g	2.6g	1724	S	0.3	0.7	296:34.51
217880	buch1019	20	0	14.5g	2.6g	8328	S	0.3	0.7	280:26.55

認識登入節點— CPU & Memory

```
sanskrit007@lgn304:~>top -b -n 1
```

顯示系統資源使用狀況 → **top**

top 例 3：不使用互動界面，直接輸出一次

```
top -b -n 1
```

```
284494 u6136713 20 0 42748 76 0 S 0.0 0.0 0:00.00
284495 u6136713 20 0 117456 920 4 S 0.0 0.0 0:00.16
284748 u6520191 20 0 113400 304 4 S 0.0 0.0 0:00.03
285067 u6520191 20 0 60352 448 0 S 0.0 0.0 0:00.00
285068 u6520191 20 0 92688 872 4 S 0.0 0.0 0:00.18
285149 u6520191 20 0 131720 860 28 S 0.0 0.0 0:03.11
285173 u6520191 20 0 370188 848 0 S 0.0 0.0 0:00.04
285178 u6520191 20 0 387100 624 0 S 0.0 0.0 0:00.00
285341 u3762779 20 0 4368 100 0 S 0.0 0.0 0:00.00
285520 root 20 0 226380 1208 0 S 0.0 0.0 0:00.12
285553 nhri060+ 20 0 0 0 0 Z 0.0 0.0 0:00.00
285720 calvind+ 20 0 116068 516 4 S 0.0 0.0 0:00.10
286206 peimiao+ 20 0 72468 772 0 S 0.0 0.0 0:00.00
286505 u3414958 20 0 618404 9712 28 S 0.0 0.0 0:01.76
286514 u3414958 20 0 42748 80 0 S 0.0 0.0 0:00.00
286515 u3414958 20 0 117276 748 4 S 0.0 0.0 0:00.25
286692 root 20 0 0 0 0 S 0.0 0.0 0:20.83
```

```
@lgn304:~>
```

Bash Shell 操作：資料篩選與資料流重導向

```
sanskrit007@lgn304:~>top -b -n 1|grep R
```

顯示系統資源使用狀況 → **top**

top 例 4：直接輸出一次，並將輸出的資料轉到 **grep** 去篩選，篩選出有字元 **R** 的列

```
top -b -n 1|grep R
```

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+
9012 ls807te+ 20 0 1423284 1.3g 1500 R 100.0 0.4 0:08.03
14997 biology 20 0 313032 158480 9752 R 100.0 0.0 0:01.07
271853 aston66+ 20 0 59.7g 19.7g 1104 R 100.0 5.2 2:32.63
6522 u3121714 20 0 25.2g 34392 996 R 100.0 0.0 0:18.29
26037 root 20 0 231344 11252 1392 R 100.0 0.0 71:22.41
137966 nanosam+ 20 0 219.8g 219.6g 1200 R 100.0 58.4 32:26.86
15207 ls807te+ 20 0 166012 14396 4376 R 45.0 0.0 0:00.09
15190 sanskri+ 20 0 166504 5168 1568 R 25.0 0.0 0:00.06
131470 nanosam+ 20 0 73772 2860 1992 R 10.0 0.0 0:54.73
131469 nanosam+ 20 0 202548 5156 1004 R 0.0 0.0 0:50.67
sanskrit007@lgn304:~>
```

資料流重導向 (I/O Redirection)

- >：將輸出存到檔案，取代檔案內資料
- >>：將輸出存到檔案，從檔案最後繼續寫入
- |：將前一個程式的輸出轉成後面程式的輸入

例：將 **top** 的輸出結果存到 **~/top_data**

```
top -b -n 1 > ~/top_data
```

整列篩選 → **grep**

grep 例 1：截取 **top** 指令中自己帳號的所有程序

```
top -b -n 1 | grep `whoami`
```

！注意：有些時候上面這個指令會永遠沒有資料被截取出來

認識登入節點— NVIDIA GPU

顯示 GPU 資訊系統資源使用狀況 → `nvidia-smi`

@lgn304:~>nvidia-smi									
Mon Jul 25 11:39:30 2022									

NVIDIA-SMI		450.80.02		Driver Version: 450.80.02			CUDA Version: 11.0		

GPU	Name	Persistence-M		Bus-Id	Disp.A	Volatile	Uncorr. ECC		
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage		GPU-Util	Compute M.	MIG M.	
=====									
0	Tesla	V100-PCIE...	On	00000000:1B:00.0	Off			0	
N/A	31C	P0	34W / 250W	762MiB / 32510MiB			0%	Default	N/A

1	Tesla	V100-PCIE...	On	00000000:1E:00.0	Off			0	
N/A	32C	P0	36W / 250W	762MiB / 32510MiB			0%	Default	N/A

2	Tesla	V100-PCIE...	On	00000000:DC:00.0	Off			0	
N/A	32C	P0	38W / 250W	762MiB / 32510MiB			0%	Default	N/A

3	Tesla	V100-PCIE...	On	00000000:DD:00.0	Off			0	
N/A	31C	P0	35W / 250W	762MiB / 32510MiB			0%	Default	N/A

Processes:									
GPU	GI	CI	PID	Type	Process name	GPU Memory			
	ID	ID				Usage			
=====									
0	N/A	N/A	167454	C	...envs/yolov4_tf/bin/python	453MiB			
0	N/A	N/A	176127	C	...da/envs/yolov7/bin/python	305MiB			
1	N/A	N/A	167454	C	...envs/yolov4_tf/bin/python	453MiB			
1	N/A	N/A	176127	C	...da/envs/yolov7/bin/python	305MiB			
2	N/A	N/A	167454	C	...envs/yolov4_tf/bin/python	453MiB			
2	N/A	N/A	176127	C	...da/envs/yolov7/bin/python	305MiB			
3	N/A	N/A	167454	C	...envs/yolov4_tf/bin/python	453MiB			
3	N/A	N/A	176127	C	...da/envs/yolov7/bin/python	305MiB			

Bash Shell 指令：目錄與檔案操作

建立目錄 → `mkdir [name]`

建立目錄 → `mkdir -p [path]/[name]`

刪除目錄 → `rm -r [name]`

刪除檔案 → `rm [name]`

複製目錄 → `cp -r [name] [path]/`

複製目錄 → `cp -r [name1] [name2]`

複製檔案 → `cp [name] [path]/`

複製檔案 → `cp [name1] [name2]`

目錄改名 → `mv [name1] [name2]`

移動目錄 → `mv [name] [path]/`

檔案改名 → `mv [name1] [name2]`

移動檔案 → `mv [name] [path]/`

建立檔案或編輯 → `vi [name]`

建立檔案或編輯 → `nano [name]`

看檔案內容 → `less [name]`

印出檔案內容到螢幕 → `cat [name]`

覆蓋檔案內容 →

```
cat << EOF > [name]
```

檔案的內容會被我蓋掉

```
EOF
```

添加資訊到檔案內容最後 →

```
cat << EOF >> [name]
```

此行以上的是檔案原本的內容

這 `cat` 的功能挺方便

```
EOF
```


Bash Shell 使用者設定檔：~/.bashrc

每次啟動 bash shell，無論是藉由登入、開啓 terminal 或是執行 `bash`，都會讀取 `~/.bashrc` 來載入設定。請將如下的設定或是你自己的設定，寫入 `~/.bashrc` 之中。

```
export PATH="/home/`whoami`/bin:$PATH"
```

```
alias o='less'
alias grep='grep --color=auto'
alias l='ls -aF'
```

```
export PS1="\u@\h:\w>"
```

```
LANG=en_US.UTF-8
LANG=zh_TW.UTF-8
LC_CTYPE=$LANG
LC_NUMERIC=$LANG
LC_TIME=$LANG
LC_COLLATE=$LANG
LC_MONETARY=$LANG
LC_MESSAGES=$LANG
LC_PAPER=$LANG
LC_NAME=$LANG
LC_ADDRESS=$LANG
LC_TELEPHONE=$LANG
LC_MEASUREMENT=$LANG
LC_IDENTIFICATION=$LANG
LC_ALL=$LANG
```

```
export ORI_LS_COLORS=$LS_COLORS
```

```
export
```

```
LS_COLORS="no=00:fi=00:di=01;34:ln=00;36:pi=40;33:
so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=41;33
;01:ex=00;32:*.cmd=00;32:*.exe=01;32:*.com=01;32:*.
bat=01;32:*.btm=01;32:*.dll=01;32:*.tar=00;31:*.t
bz=00;31:*.tgz=00;31:*.rpm=00;31:*.deb=00;31:*.arj
=00;31:*.taz=00;31:*.lzh=00;31:*.lzma=00;31:*.zip=
00;31:*.zoo=00;31:*.z=00;31:*.Z=00;31:*.gz=00;31:*.
bz2=00;31:*.tb2=00;31:*.tz2=00;31:*.tbz2=00;31:*.
xz=00;31:*.avi=01;35:*.bmp=01;35:*.dl=01;35:*.fli=
01;35:*.gif=01;35:*.gl=01;35:*.jpg=01;35:*.jpeg=01
;35:*.mkv=01;35:*.mng=01;35:*.mov=01;35:*.mp4=01;3
5:*.mpg=01;35:*.pcx=01;35:*.pbm=01;35:*.pgm=01;35:
*.png=01;35:*.ppm=01;35:*.svg=01;35:*.tga=01;35:*.
tif=01;35:*.webm=01;35:*.webp=01;35:*.wmv=01;35:*.
xbm=01;35:*.xcf=01;35:*.xpm=01;35:*.aiff=00;32:*.a
pe=00;32:*.au=00;32:*.flac=00;32:*.m4a=00;32:*.mid
=00;32:*.mp3=00;32:*.mpc=00;32:*.ogg=00;32:*.voc=0
0;32:*.wav=00;32:*.wma=00;32:*.wv=00;32:"
```

撰寫 Bash Script

在 `bash` 環境能夠執行的指令，都能寫成一個文字檔，並藉由執行該檔案來執行檔案的內容。

1. 建立目錄 `~/bin`
2. 將如下資訊寫入 `~/bin/ohmy1`

```
echo "Hello my friend~"
echo "Hello \nmy friend~"
echo
echo -e "Hello my friend~\nI am `whoami`. How have
you been?"
echo
date "+%Y/%m/%d %X"
echo
echo "現在時間：`date "+%Y/%m/%d %X"`"
sleep 10
echo "現在時間：`date "+%Y/%m/%d %X"`"
```

執行程式的要件

1. 目前帳號有讀取該檔案與執行該檔案的權限
2. Shell 找得到該檔案
3. 程式的內容是可以被看懂的

1. 修改檔案或目錄權限 → `chmod 500 [name]`

第二到第四個字元：擁有者的權限

第五到第七個字元：群組的權限

第八到第十個字元：其他人的權限

```
lrwxrwxrwx.
dr-xr-xr-x.
```

1 2~4 5~7 8~10 11

類型

l：連結
d：目錄
-：檔案

421 421 421

```
lrwxrwxrwx.
dr-xr-xr-x.
```

401 401 401

↓ ↓ ↓
5 5 5

`chmod 555`

修改擁有者 → `chown [user] [file]`

修改群組 → `chgrp [grp] [name]`

執行程式的要件

1. 目前帳號有讀取該檔案與執行該檔案的權限
2. Shell 找得到該檔案
3. 程式的內容是可以被看懂的

2.1. 確認與修改 PATH 變數

確認：`echo $PATH`

修改：`PATH="$HOME/bin:$PATH"`

修改：`export PATH="$HOME/bin:$PATH"`

2.2. 確認與修改 LD_LIBRARY_PATH 變數

確認：`echo $LD_LIBRARY_PATH`

修改：`LD_LIBRARY_PATH="$HOME/lib:$LD_LIBRARY_PATH"`

修改：`export LD_LIBRARY_PATH="$HOME/lib:$LD_LIBRARY_PATH"`

執行程式的要件

1. 目前帳號有讀取該檔案與執行該檔案的權限
2. Shell 找得到該檔案
3. 程式的內容是可以被看懂的

3.1. Binary file

輸入該檔案名稱直接執行，如果內容與機器格式以及作業系統特性相符就能執行。

3.2. Text file

方法一：使用與內容相符的程式來執行。例如：sh, bash, python, perl, julia ... 等。

```
bash [file]
```

```
python [file]
```

這種方法的檔案可以不用具有可執行的權限，只要可讀取即可。

方法二：直接執行該檔案，前提是檔案第一行必須以正確格式指明要用來執行該檔案的程式。例如：
： sh, bash, python, perl, julia ... 等。

執行 Bash Script

在 `bash` 環境能夠執行的指令，都能寫成一個文字檔，並藉由執行該檔案來執行檔案的內容。

1. 建立目錄 `~/bin`
2. 將如下資訊寫入 `~/bin/ohmy1`

方法一

```
echo "Hello my friend~"
echo "Hello \nmy friend~"
echo
echo -e "Hello my friend~\nI am `whoami`. How have
you been?"
echo
date "+%Y/%m/%d %X"
echo
echo "現在時間：`date "+%Y/%m/%d %X"`"
sleep 10
echo "現在時間：`date "+%Y/%m/%d %X"`"
```

方法一的執行

```
bash ~/bin/ohmy1
```

方法二

```
#!/bin/bash
echo "Hello my friend~"
echo "Hello \nmy friend~"
echo
echo -e "Hello my friend~\nI am `whoami`. How have
you been?"
echo
date "+%Y/%m/%d %X"
echo
echo "現在時間：`date "+%Y/%m/%d %X"`"
sleep 10
echo "現在時間：`date "+%Y/%m/%d %X"`"
```

方法二的執行

```
ohmy1 或 ~/bin/ohmy1
```

執行 Bash Script

```
echo "This command is $command"
```

```
if [[ $# == 0 ]]
```

```
then
```

```
  pathIn=`pwd`
```

```
  echo "使用工作目錄 $pathIn"
```

```
elif [[ $# == 1 ]]
```

```
then
```

```
  echo "使用目錄$pathIn"
```

```
else
```

```
  echo "Too many things..."
```

```
  exit
```

```
fi
```

```
all=`ls -p $pathIn` || exit
```

```
allNum=`ls -p $pathIn | wc -l`
```

```
dirNum=`ls -p $pathIn | grep / | wc -l`
```

```
fileNum=`ls -p $pathIn | grep -v / | wc -l`
```

```
echo "總共有 $allNum 個檔案與目錄"
```

```
echo "目錄有 $dirNum 個"
```

```
echo "檔案有 $fileNum 個"
```

```
echo "時間爲:`date +%Y/%m/%d %X`"
```

```
if [[ $dirNum > 0 ]]
```

```
then
```

```
  echo
```

```
  echo "在 $pathIn 中的目錄爲:"
```

```
  for dir in `ls -p $pathIn | grep /\`
```

```
  do
```

```
    echo $dir
```

```
  done
```

```
fi
```

```
if [[ $fileNum > 0 ]]
```

```
then
```

```
  echo
```

```
  echo "在 $pathIn 中的檔案爲:"
```

```
  for dir in `ls -p $pathIn | grep -v /\`
```

```
  do
```

```
    echo $dir
```

```
  done
```

```
fi
```

其他 Linux 指令

取得執行檔與說明檔的完整路徑 → `whereis [name]`

顯示程式說明 → `man [name]`

刪除程序 → `kill PID`

顯示此次所使用的指令歷史 → `history`

休息幾秒 → `sleep [second]`

顯示此次所使用的指令歷史 → `history`

解壓縮檔案 → `tar -xvzf filename.tar.gz`

```
tar -xjvf filename.tar.bz2
```

```
tar -xJvf filename.tar.xz
```

壓縮檔案 → `tar -czvf filename.tar.gz filename`

SLURM 操作與各式環境設定

臺灣杉三號 SLURM 使用說明頁面

台灣杉三號使用說明

<https://man.twcc.ai/@TWCC-III-manual/H1bEXeGcu>

台灣杉三號生醫節點使用說明

https://man.twcc.ai/@Ldk_QYrOR2yo3m8Cb1549A/rkegDKslF

SLURM Summary

<https://slurm.schedmd.com/pdfs/summary.pdf>