教育部「5G行動寬頻人才培育跨校教學聯盟計畫」

5G行動網路協定與核網技術聯盟中心

「5G行動寬頻協同網路」課程模組

實驗一
開源碼 srsLTE
平台建置與基本量測

副教授：吳俊興

助教：林原進、吳振宇

國立高雄大學 資訊工程學系

Outline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- 基本 Linux 指令
  - 檔案相關指令
  - 網路相關指令
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
- 執行程式暨測試
- 總結

實驗目的

- 建置srsLTE的srsUE、srsENB+srsEPC，讓學生學會建立srsLTE行動通訊網路開源碼實驗平台。
- 透過srsUE以USRP連接srsENB+srsEPC進行觀察與量測，讓學生熟悉網路的偵錯及量測工具。

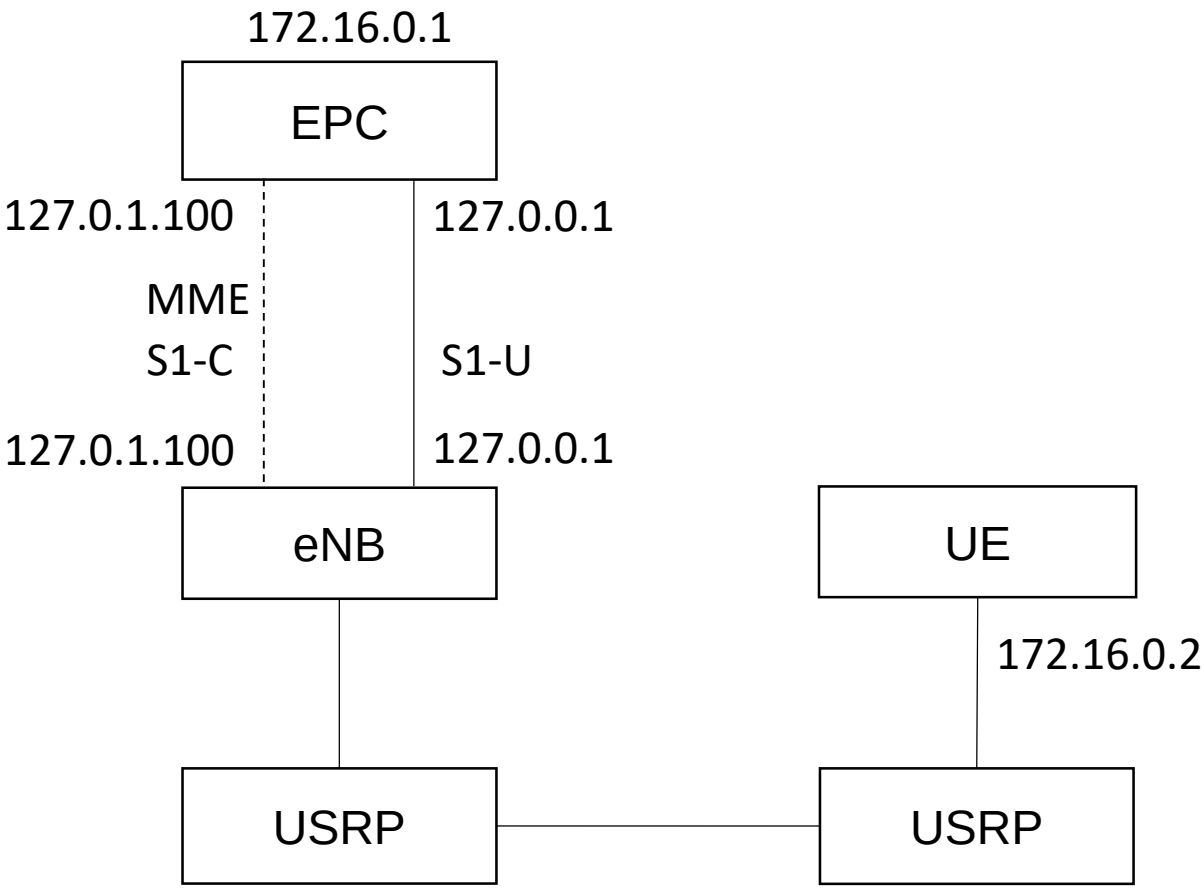
實驗內容

- 利用指令了解 Liunx 運作
  - 更改目錄
  - 移動檔案
  - 網路環境設置
- 在兩台主機上分別安裝 srsLTE 環境
  - 安裝所需套件
  - 執行程式
  - 量測封包

Outline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- 基本 Linux 指令
  - 檔案相關指令
  - 網路相關指令
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
- 執行程式暨測試
- 總結

srsLTE Small Cell 實驗架構



軟硬體環境 – 硬體

名稱	規格	數量	目的
EPC+eNB	電腦型號 ： ASUS VivoMini UN65H	1	啟動 MME,S-GW,P-GW
	USRP B210	1	啟動 srsLTE eNB
UE	電腦型號 ： ASUS NB M580V	1	模擬 UE
	USRP B210	1	啟動 srsLTE UE

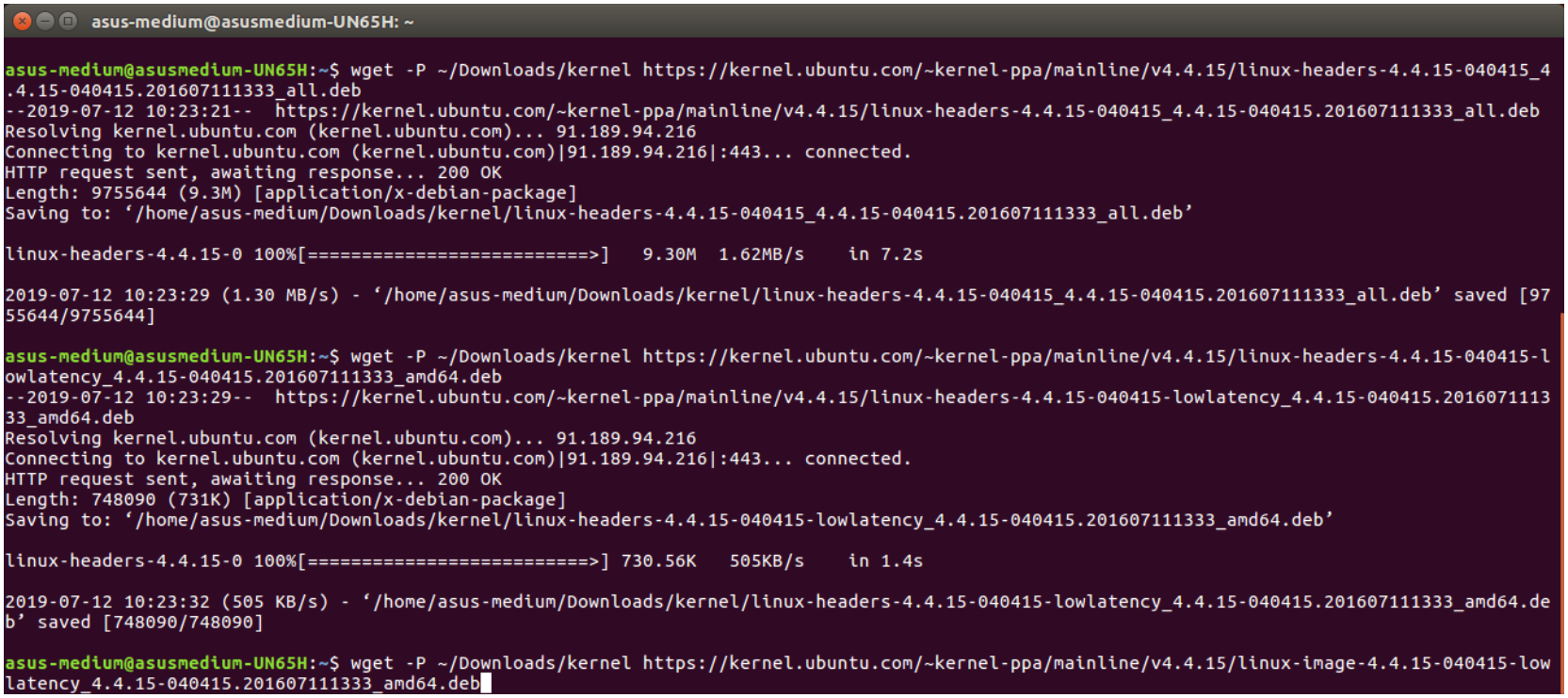
軟硬體環境 – 軟體

名稱	軟體	版本
EPC+eNB	OS : Ubuntu	Ubuntu 16.04
		Kernel : 4.15.0-041500-lowlatancy
	srsLTE	srsLTE 19.03 5343b33f8ab2edf7319b6abb07bbc3970541517a
UE	OS : Ubuntu	Ubuntu 16.04
		Kernel : 4.15.0-041500-lowlatancy
	srsLTE	srsLTE 19.03 5343b33f8ab2edf7319b6abb07bbc3970541517a

下載及安裝Kernel

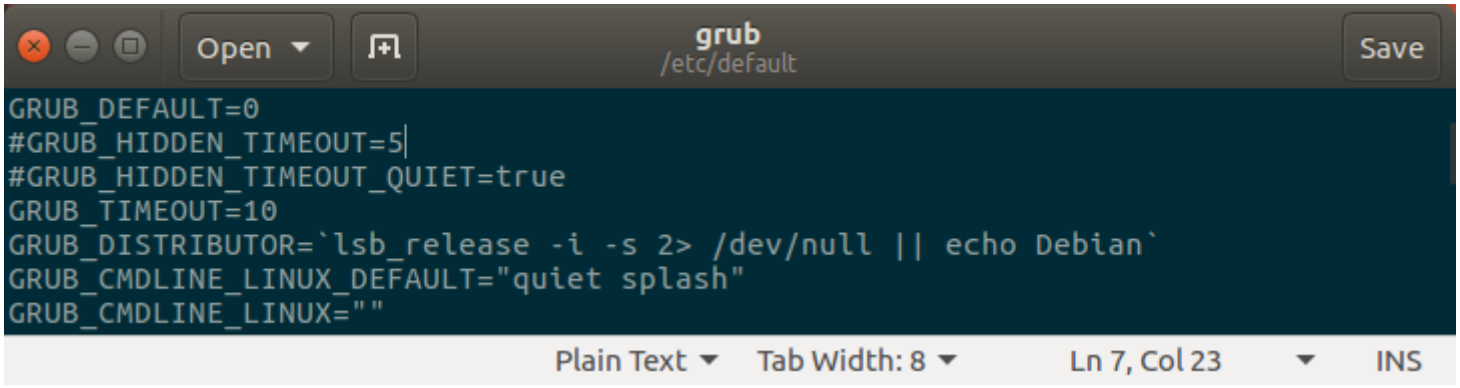
- 開啟一個終端機，並且依序輸入
- wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-headers-4.4.15-040415_4.4.15-040415.201607111333_all.deb
- wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-headers-4.4.15-040415-lowlatency_4.4.15-040415.201607111333_amd64.deb
- wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-image-4.4.15-040415-lowlatency_4.4.15-040415.201607111333_amd64.deb
- sudo dpkg -i ~/Downloads/kernel/*.deb

安裝過程


asus-medium@asusmedium-UN65H: ~
asus-medium@asusmedium-UN65H:~$ wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-headers-4.4.15-040415_4.4.15-040415.201607111333_all.deb
--2019-07-12 10:23:21-- https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-headers-4.4.15-040415_4.4.15-040415.201607111333_all.deb
Resolving kernel.ubuntu.com (kernel.ubuntu.com)... 91.189.94.216
Connecting to kernel.ubuntu.com (kernel.ubuntu.com)|91.189.94.216|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9755644 (9.3M) [application/x-debian-package]
Saving to: '/home/asus-medium/Downloads/kernel/linux-headers-4.4.15-040415_4.4.15-040415.201607111333_all.deb'
linux-headers-4.4.15-0 100%[=========>] 9.30M 1.62MB/s in 7.2s
2019-07-12 10:23:29 (1.30 MB/s) - '/home/asus-medium/Downloads/kernel/linux-headers-4.4.15-040415_4.4.15-040415.201607111333_all.deb' saved [9755644/9755644]
asus-medium@asusmedium-UN65H:~$ wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-headers-4.4.15-040415-lowlatency_4.4.15-040415.201607111333_amd64.deb
--2019-07-12 10:23:29-- https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-headers-4.4.15-040415-lowlatency_4.4.15-040415.201607111333_amd64.deb
Resolving kernel.ubuntu.com (kernel.ubuntu.com)... 91.189.94.216
Connecting to kernel.ubuntu.com (kernel.ubuntu.com)|91.189.94.216|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 748090 (731K) [application/x-debian-package]
Saving to: '/home/asus-medium/Downloads/kernel/linux-headers-4.4.15-040415-lowlatency_4.4.15-040415.201607111333_amd64.deb'
linux-headers-4.4.15-0 100%[=========>] 730.56K 505KB/s in 1.4s
2019-07-12 10:23:32 (505 KB/s) - '/home/asus-medium/Downloads/kernel/linux-headers-4.4.15-040415-lowlatency_4.4.15-040415.201607111333_amd64.deb' saved [748090/748090]
asus-medium@asusmedium-UN65H:~$ wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-image-4.4.15-040415-lowlatency_4.4.15-040415.201607111333_amd64.deb

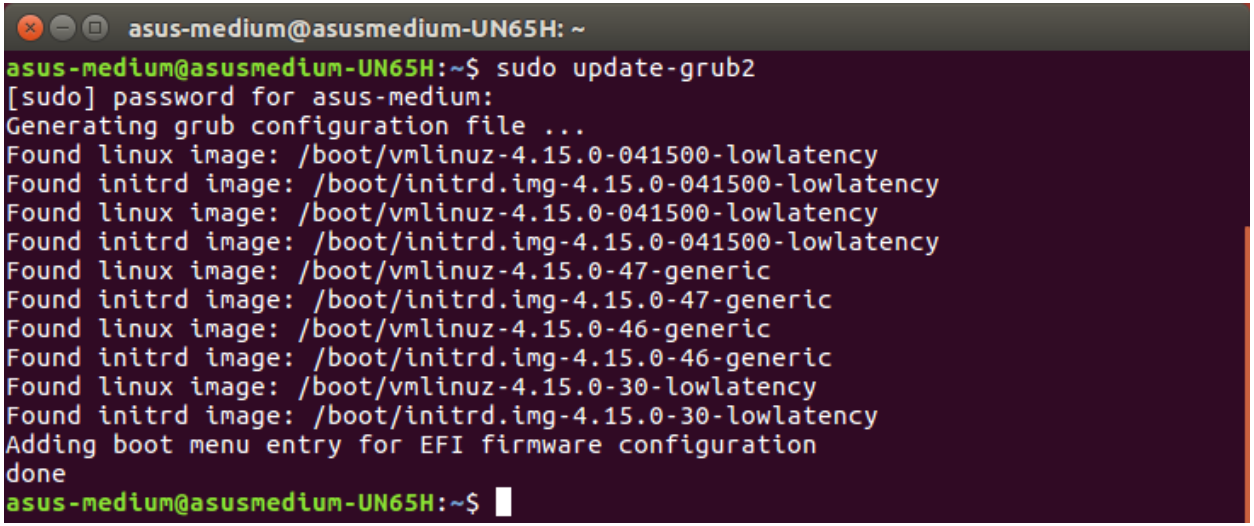
修改開機選單和設定

- 開啟終端機輸入以下指令
- sudo gedit /etc/default/grub
- 找到下列文字
  GRUB_HIDDEN_TIMEOUT=0
  GRUB_HIDDEN_TIMEOUT_QUIET=true
- 改成
  #GRUB_HIDDEN_TIMEOUT=0
  #GRUB_HIDDEN_TIMEOUT_QUIET=true



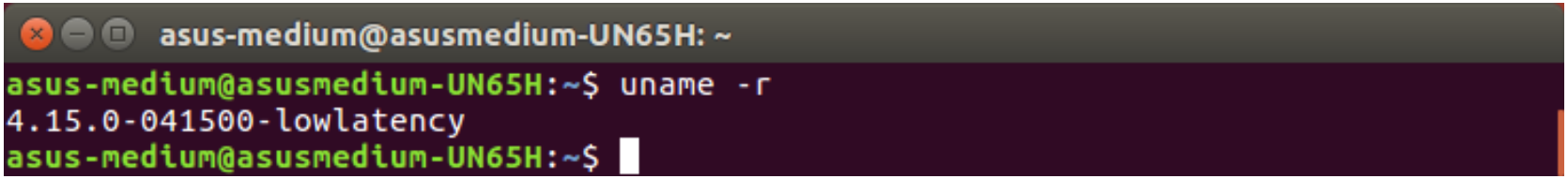
更新grub設定

- 在終端機輸入以下指令
- sudo update-grub2
- 接著輸入以下指令，重新啟動電腦
- sudo reboot
- 然後在開機選單選擇剛才安裝的lowlatency



檢查Kernel版本

- 重新開機後在終端機輸入指令，確認版本
- unamr -r



Outline

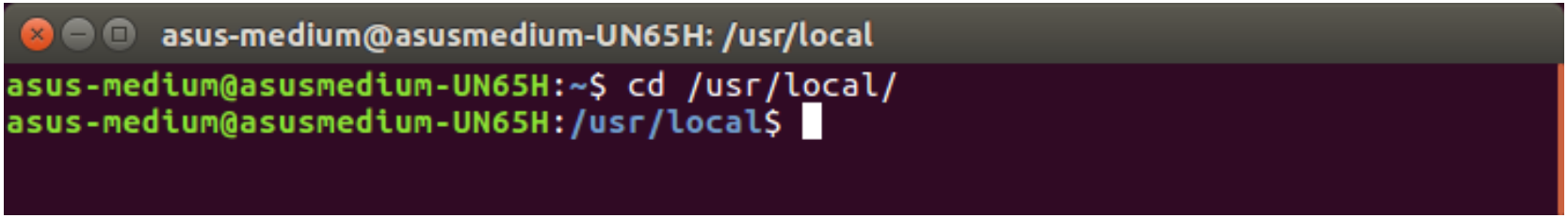
- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- 基本 Linux 指令
  - 檔案相關指令
  - 網路相關指令
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
- 執行程式暨測試
- 總結

指令概述

- 指令格式分三大部分 : [Command] [Options] [Arguments]
  - ping 127.0.0.1 -c 10
- 選項前面必須加”-”作為前導，多個選項可以合併
  - ls -a -l -t 等同 ls -alt
- Unix系統 指令區分大小寫

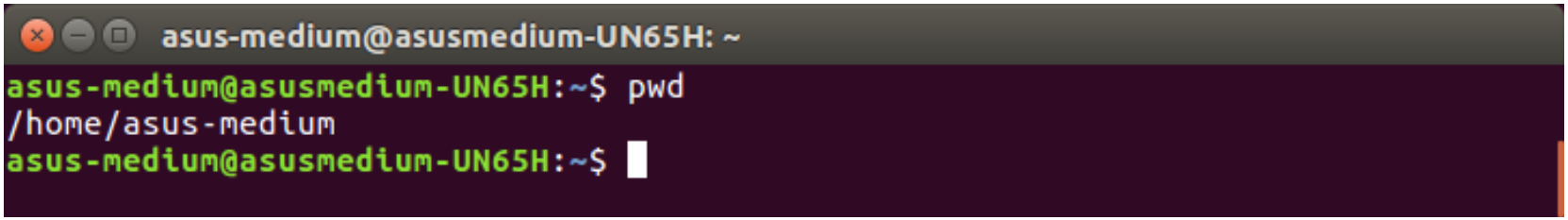
cd

- 更改工作路徑指令
- 相對路徑 vs. 絕對路徑
- cd ..　　#回到上一層
- cd ~　　#在ubuntu 預設是 /home/user_name/
- cd /usr/bin #移動到指定路徑



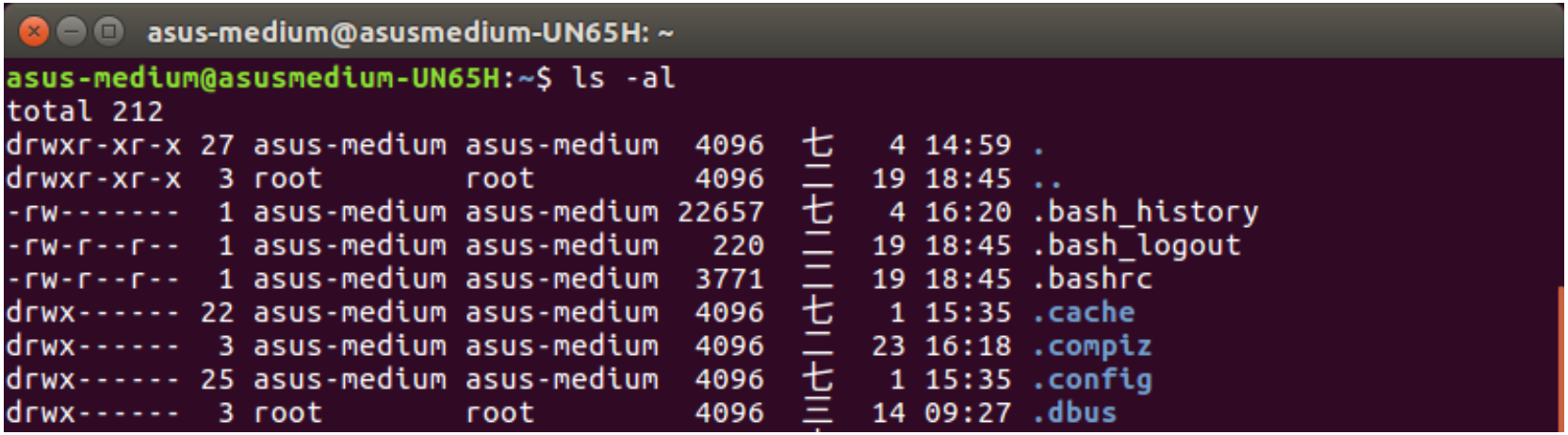
pwd

- 顯示目前的工作路徑



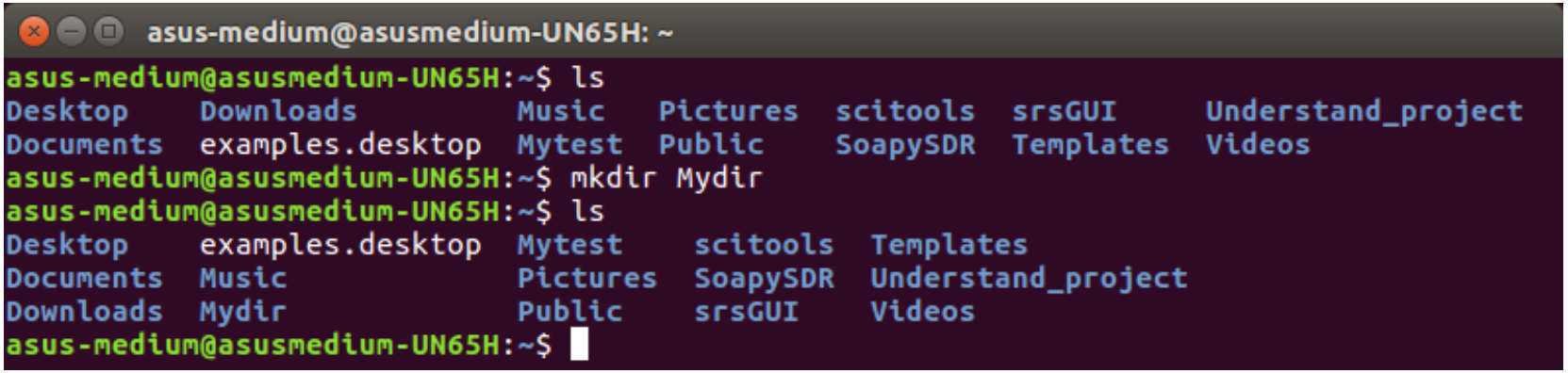
ls

- 顯示檔案名稱與屬性的指令
- ls -a　　#顯示全部檔案(包括隱藏檔案)
- ls -l #顯示更詳細的資訊
- ls --help　# ls指令的詳細用法



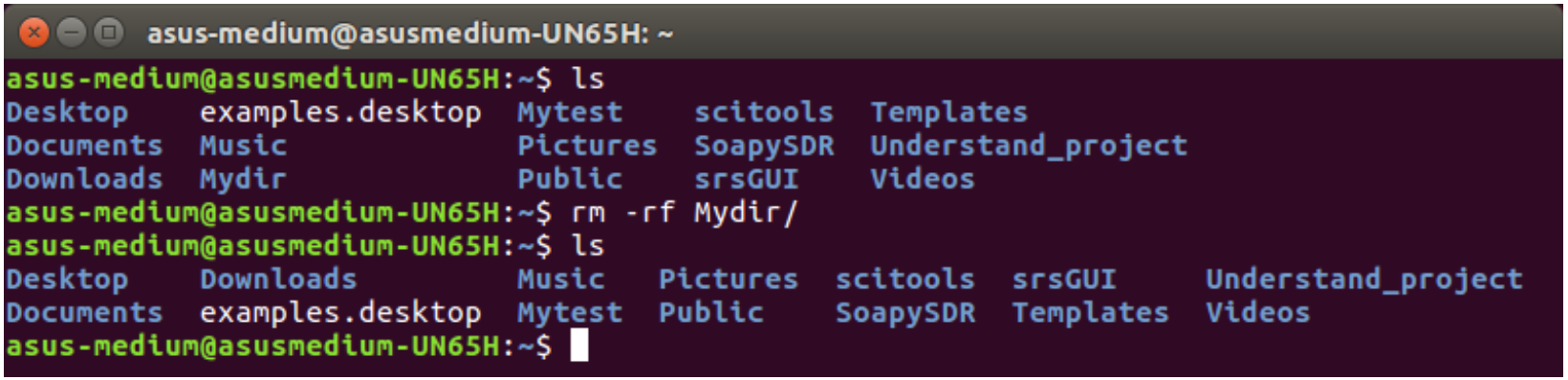
mkdir

- 建立一個目錄/資料夾
- mkdir [folder_name]
- mkdir --help　　#mkdir指令的詳細用法



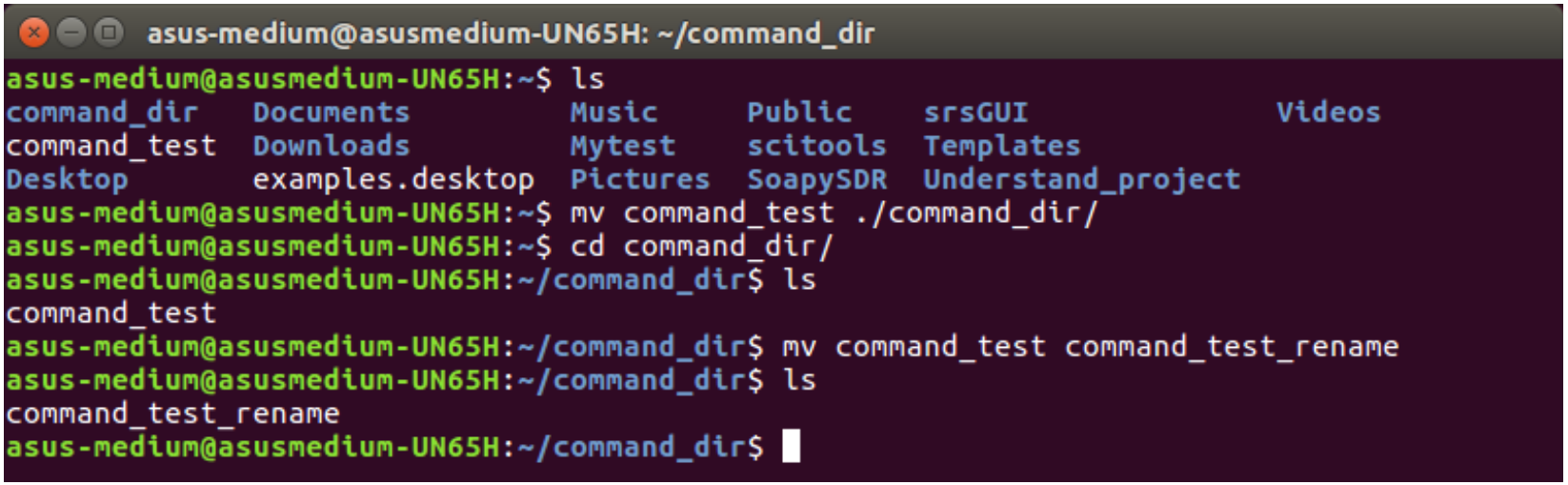
rm

- 移除的指令
- rm -r　　#連同目錄裡面包含的檔案一併刪除(recursive)
- rm -i　　#在每次刪除檔案前，都會確認一次
- rm -f　　#強制執行，並且不會確認
- rm --help #rm指令的詳細用法



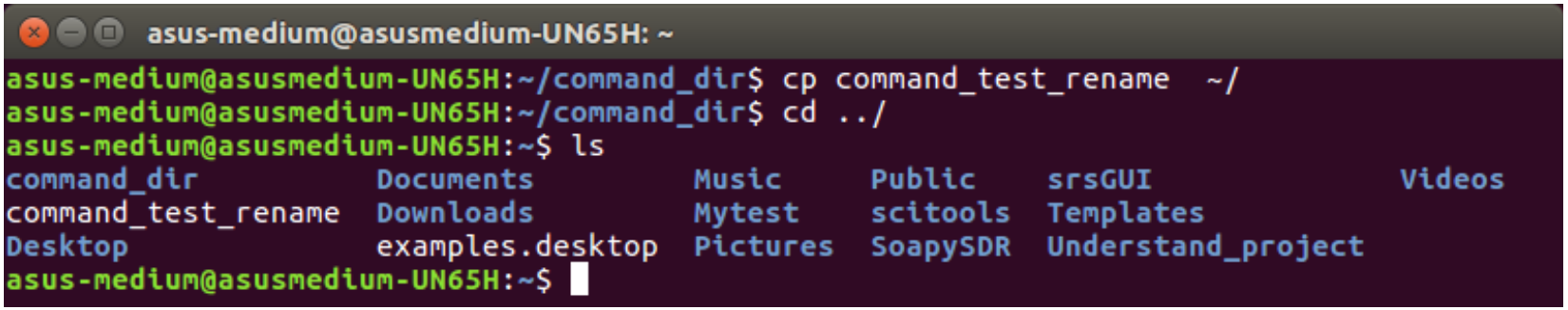
mv

- 移動檔案或是目錄的指令
- mv [file_name] [dir_path] #將檔案移動至指定目錄
- mv [file_name] [file_rename] #重新命名檔案
- mv --help #mv指令的詳細用法



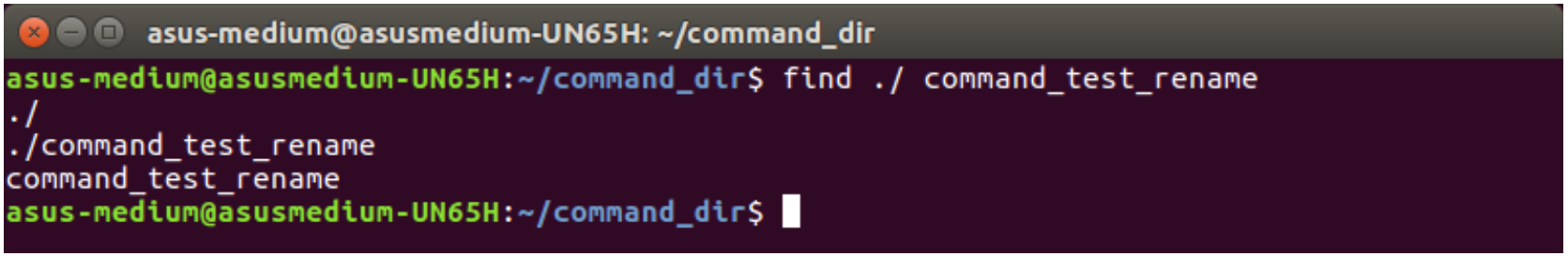
cp

- 複製檔案的指令
- cp [file_name] [path] #複製檔案至指定目錄
- cp --help #cp指令的詳細用法



find

- 尋找檔案的指令
- find [path] [file_name]
  #在指定目錄下尋找檔名為file_name的檔案
- find --help #find指令的詳細用法



grep

- 搜尋某些特定字元的指令，通常搭配其他指令使用
- grep [OPTION]... PATTERN [FILE]...
- grep --help　　#grep指令的詳細用法

Linux 檔案權限

- 三種身分: owner/group/others
- 三種權限: read/write/execute
  #若是不具有該權限則會用 - 表示
- 權限分數: r = 4分,w = 2分,x = 1分,- = 0分

新增使用者

- sudo useradd test1
- sudo passwd test1

chmod

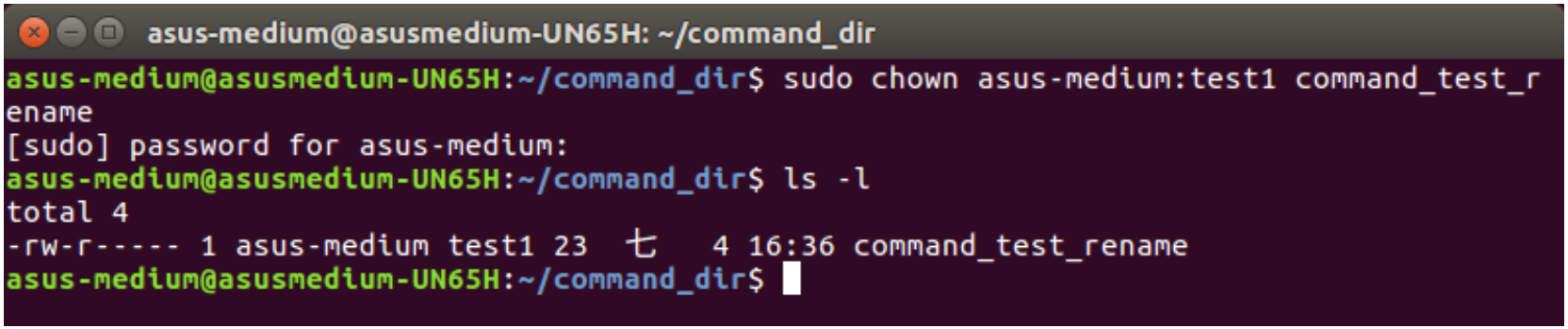
- 更改檔案權限的指令，可分兩種方式設定
- 數字 chmod 740 [file_name]
- 文字

chmod	u	+(加上)	r	檔案/目錄
	g	-(減去)	w	
	o	=(設定)	x	
	a			

```
asus-medium@asusmedium-UN65H: ~/command_dir
asus-medium@asusmedium-UN65H:~/command_dir$ ls -l
total 4
-rw-rw-r-- 1 asus-medium asus-medium 23  七   4 16:36 command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$ chmod 777 command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$ ls -l
total 4
-rwxrwxrwx 1 asus-medium asus-medium 23  七   4 16:36 command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$ chmod u=rw,g=r,o=- command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$ ls -l
total 4
-rw-r----- 1 asus-medium asus-medium 23  七   4 16:36 command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$
```

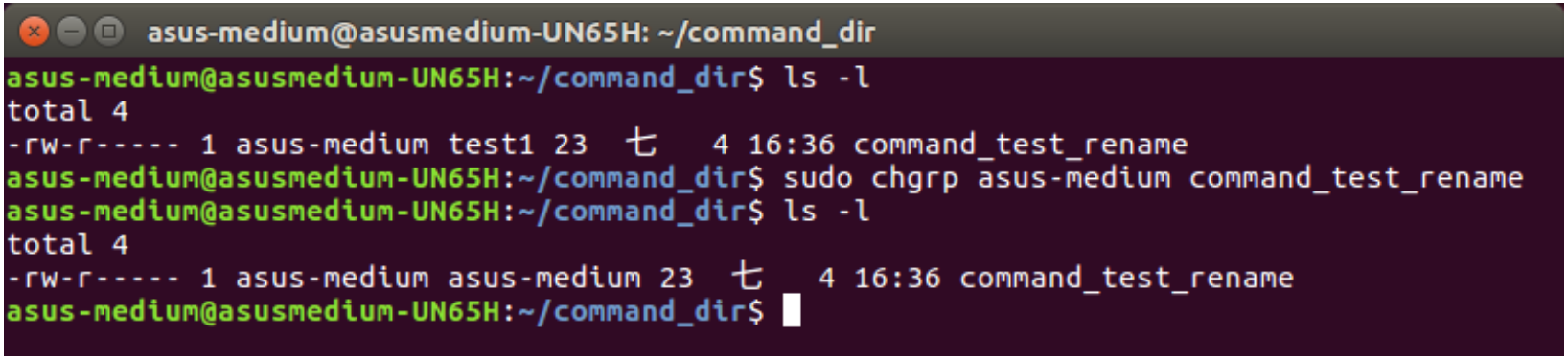
chown

- 更改檔案的擁有人及擁有群組的指令
- chown owner:group [file]



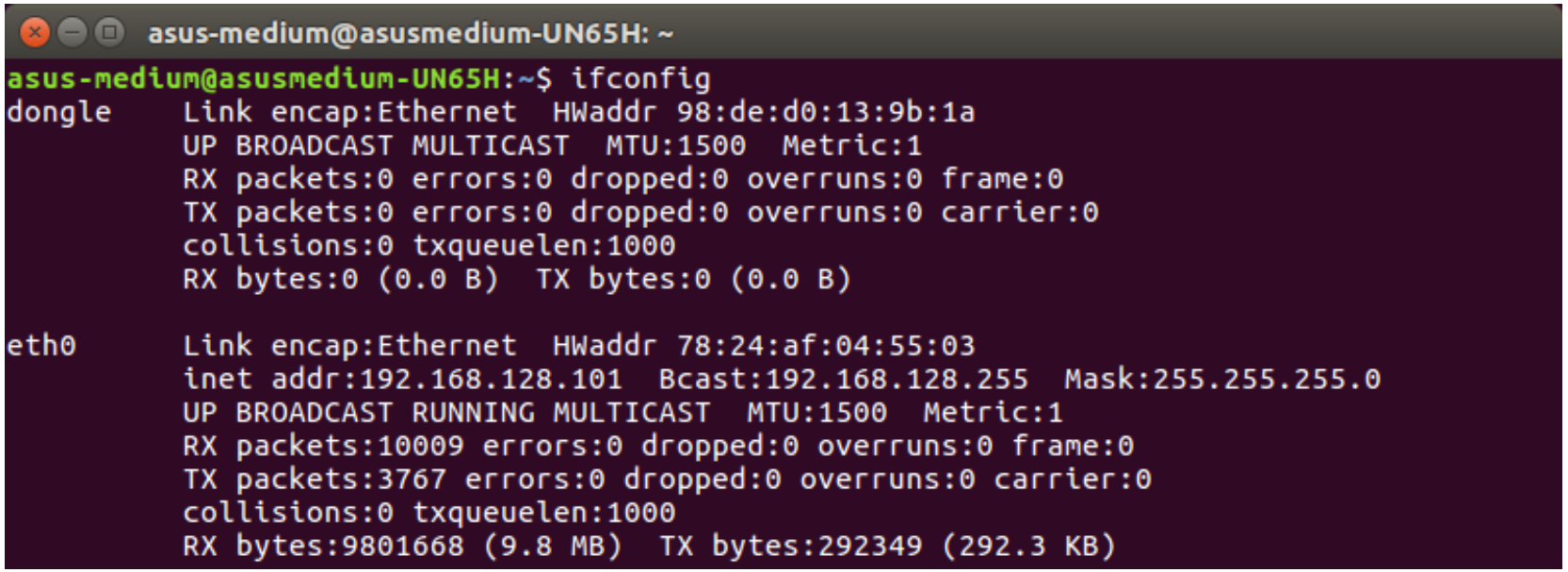
chgrp

- 改變檔案的擁有群組
- chgrp group [file]

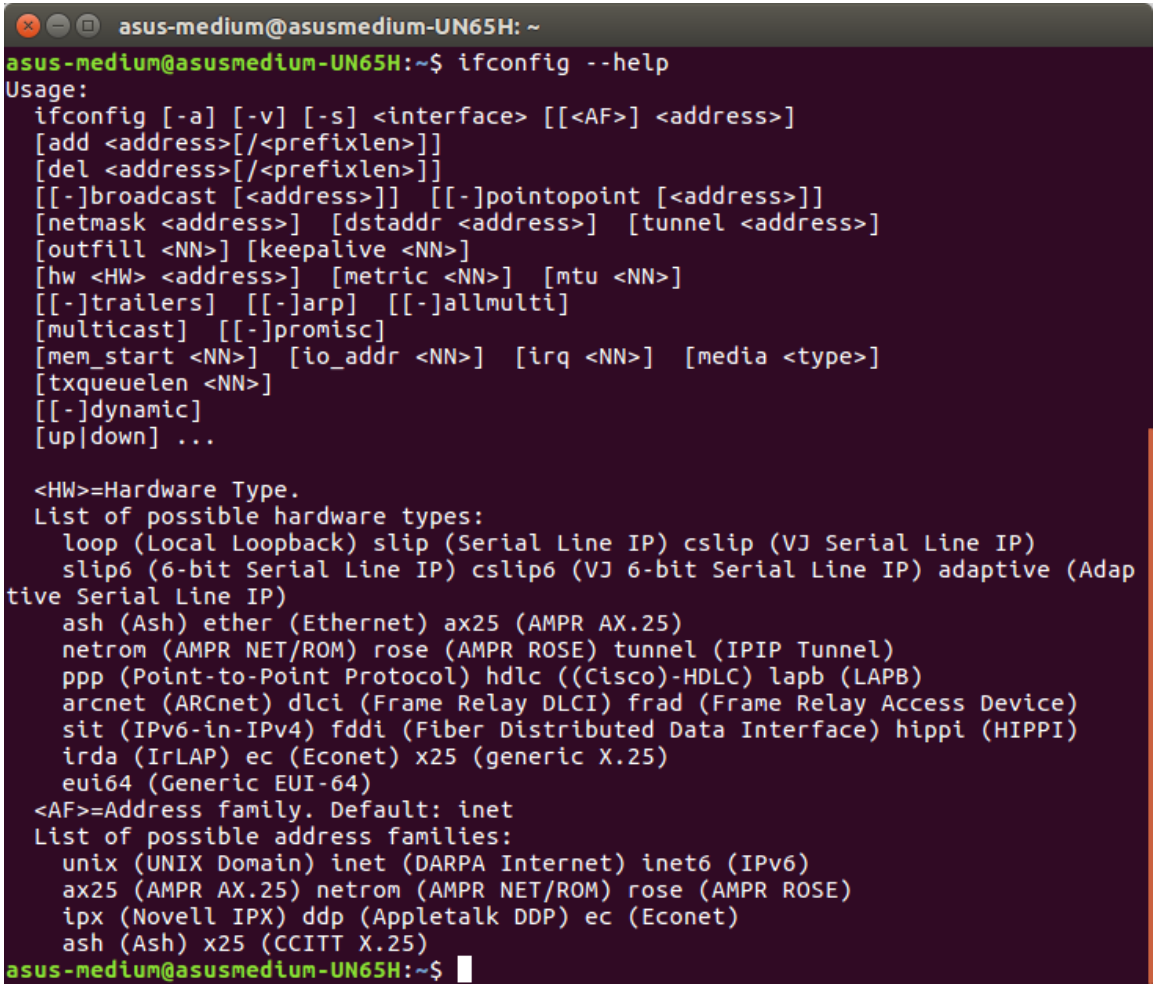


ifconfig

- 顯示網路介面卡狀況的指令
- ifconfig　#顯示使用中的網卡參數
- ifconfig -a#顯示全部的網卡，包含關閉的
- ifconfig --help #ifconfig指令的詳細用法

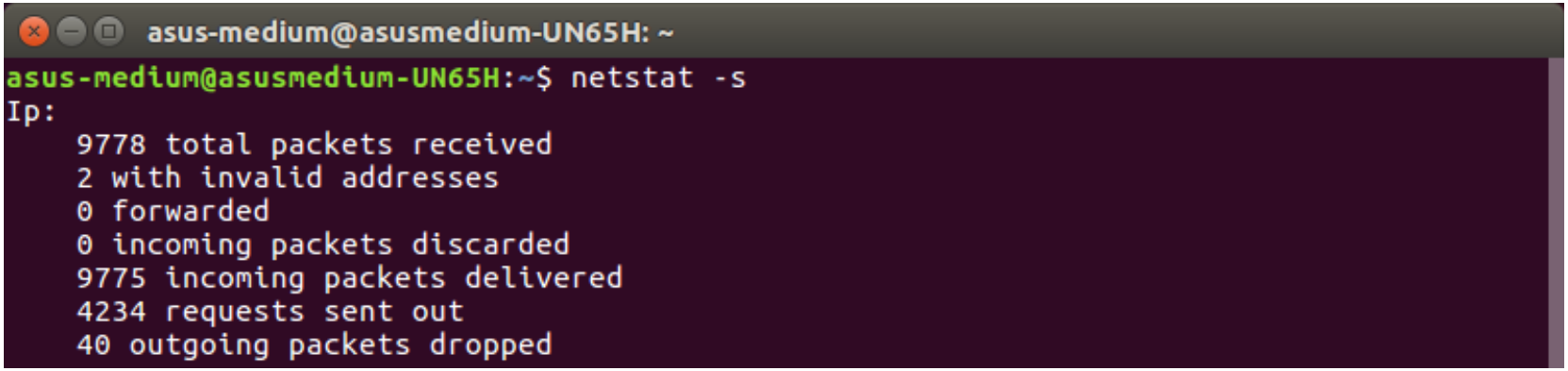


ifconfig


asus-medium@asusmedium-UN65H: ~
asus-medium@asusmedium-UN65H:~$ ifconfig --help
Usage:
ifconfig [-a] [-v] [-s] <interface> [[<AF>] <address>]
[add <address>[/<prefixlen>]]
[del <address>[/<prefixlen>]]
[[-]broadcast [<address>]] [[-]pointopoint [<address>]]
[netmask <address>] [dstaddr <address>] [tunnel <address>]
[outfill <NN>] [keepalive <NN>]
[hw <HW> <address>] [metric <NN>] [mtu <NN>]
[[-]trailers] [[-]arp] [[-]allmulti]
[multicast] [[-]promisc]
[mem_start <NN>] [io_addr <NN>] [irq <NN>] [media <type>]
[txqueuelen <NN>]
[[-]dynamic]
[up|down] ...
<HW>=Hardware Type.
List of possible hardware types:
loop (Local Loopback) slip (Serial Line IP) cslip (VJ Serial Line IP)
slip6 (6-bit Serial Line IP) cslip6 (VJ 6-bit Serial Line IP) adaptive (Adaptive Serial Line IP)
ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
netrom (AMPR NET/ROM) rose (AMPR ROSE) tunnel (IPIP Tunnel)
ppp (Point-to-Point Protocol) hdlc ((Cisco)-HDLC) lapb (LAPB)
arcnet (ARCnet) dlci (Frame Relay DLCI) frad (Frame Relay Access Device)
sit (IPv6-in-IPv4) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
irda (IrLAP) ec (Econet) x25 (generic X.25)
eui64 (Generic EUI-64)
<AF>=Address family. Default: inet
List of possible address families:
unix (UNIX Domain) inet (DARPA Internet) inet6 (IPv6)
ax25 (AMPR AX.25) netrom (AMPR NET/ROM) rose (AMPR ROSE)
ipx (Novell IPX) ddp (Appletalk DDP) ec (Econet)
ash (Ash) x25 (CCITT X.25)
asus-medium@asusmedium-UN65H:~$

netstat

- 顯示Linux網路系統的詳細資訊
- netstat -a #顯示所有開啟的Socket
- netstat -p #顯示程式名稱
- netstat -s #顯示每個協定的統計結果
- netstat --help #netstat指令的更詳細用法

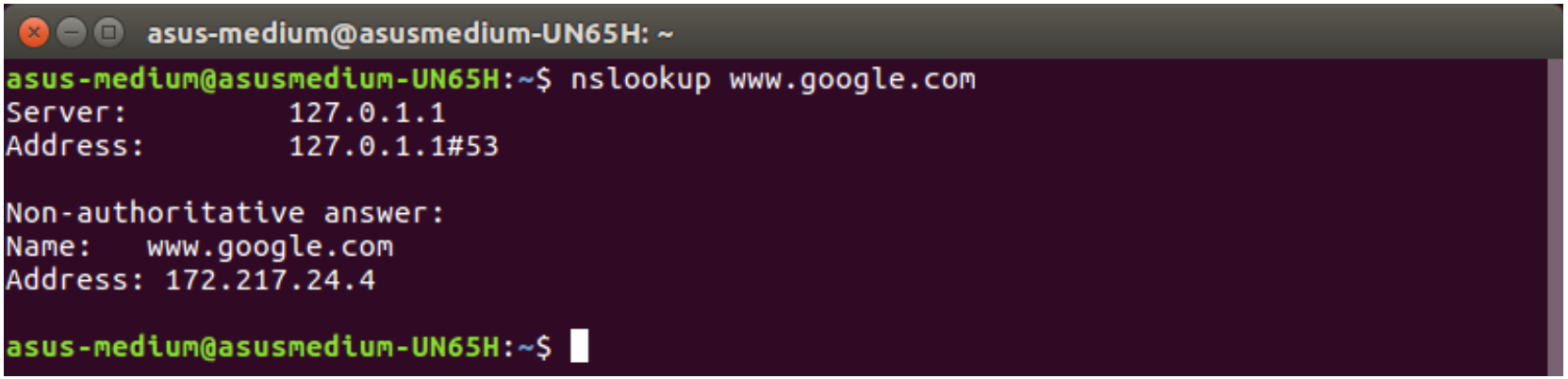


netstat

```
ue@ue-X580VD:~$ netstat --help
usage: netstat [-vWeenNcCF] [<Af>] -r         netstat {-V|--version|-h|--help}
       netstat [-vWnNcaeol] [<Socket> ...]
       netstat { [-vWeenNac] -i | [-cWnNe] -M | -s }

        -r, --route              display routing table
        -i, --interfaces         display interface table
        -g, --groups             display multicast group memberships
        -s, --statistics         display networking statistics (like SNMP)
        -M, --masquerade         display masqueraded connections

        -v, --verbose            be verbose
        -W, --wide               don't truncate IP addresses
        -n, --numeric            don't resolve names
        --numeric-hosts          don't resolve host names
        --numeric-ports          don't resolve port names
        --numeric-users          don't resolve user names
        -N, --symbolic           resolve hardware names
        -e, --extend             display other/more information
        -p, --programs           display PID/Program name for sockets
        -c, --continuous         continuous listing

        -l, --listening          display listening server sockets
        -a, --all, --listening   display all sockets (default: connected)
        -o, --timers             display timers
        -F, --fib                display Forwarding Information Base (default)
        -C, --cache              display routing cache instead of FIB

  <Socket>={-t|--tcp} {-u|--udp} {-w|--raw} {-x|--unix} --ax25 --ipx --netrom
  <AF>=Use '-6|-4' or '-A <af>' or '--<af>'; default: inet
  List of possible address families (which support routing):
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
ue@ue-X580VD:~$
```

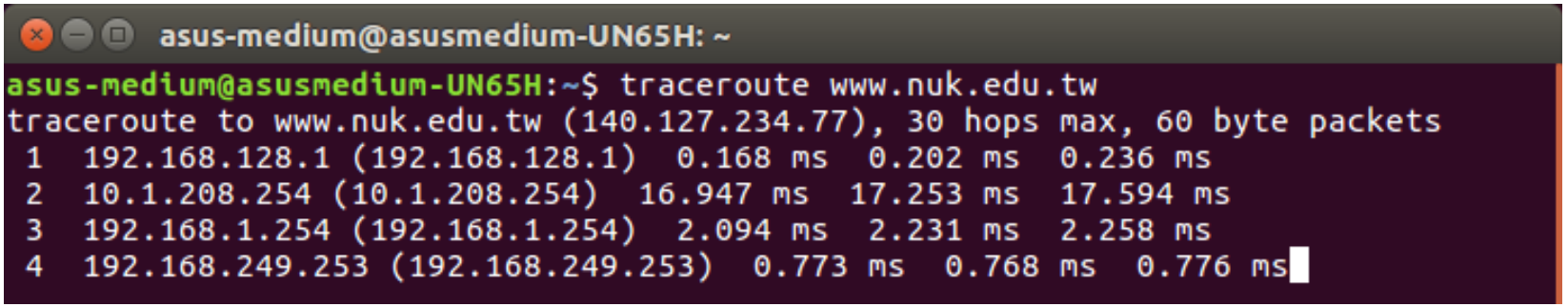
nslookup

- 查詢DNS
- nslookup 網址

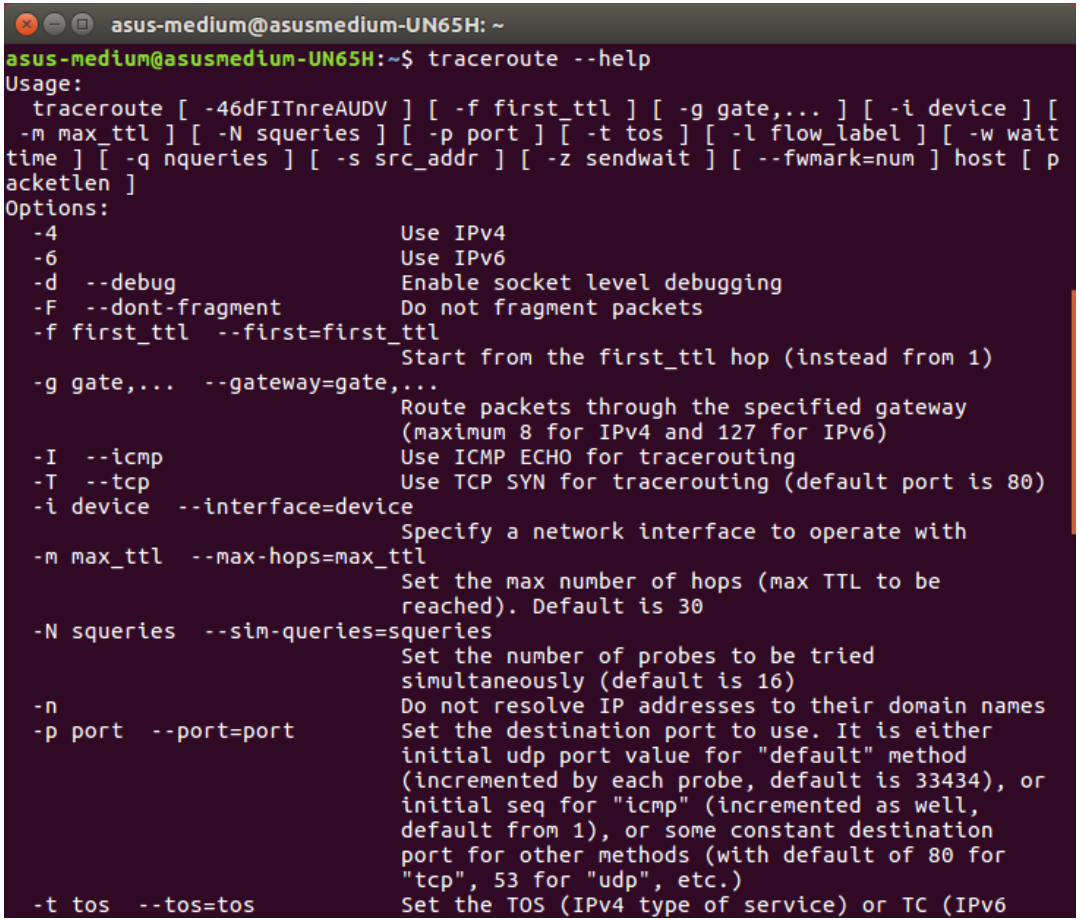


traceroute

- 追蹤封包流向
- traceroute dst
- traceroute --help　　#traceroute指令的詳細用法

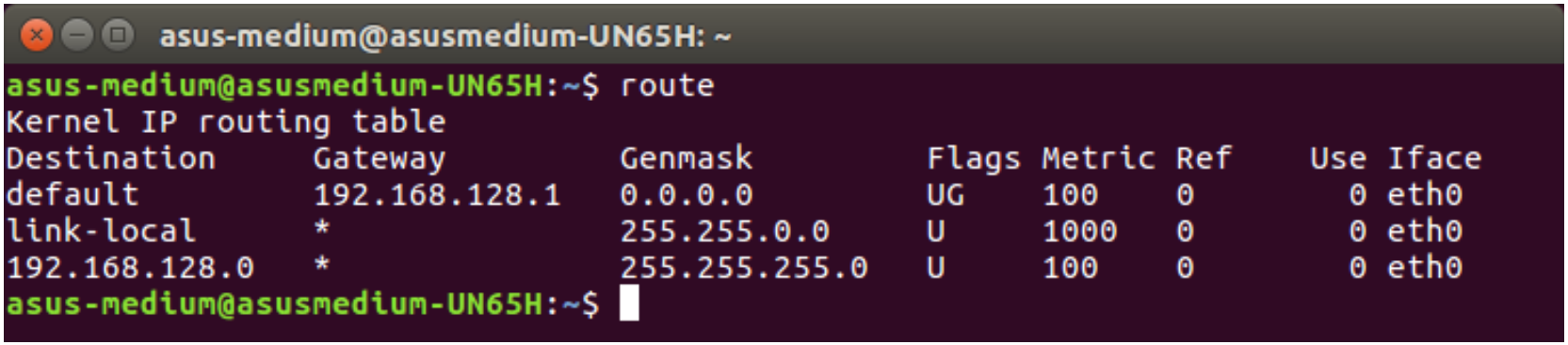


traceroute


asus-medium@asusmedium-UN65H: ~
asus-medium@asusmedium-UN65H:~$ traceroute --help
Usage:
  traceroute [-46dFITnreAUDV] [-f first_ttl] [-g gate,...] [-i device] [-m max_ttl] [-N squeries] [-p port] [-t tos] [-l flow_label] [-w waittime] [-q nqueries] [-s src_addr] [-z sendwait] [--fwmark=num] host [packetlen]
Options:
  -4   Use IPv4
  -6   Use IPv6
  -d --debug   Enable socket level debugging
  -F --dont-fragment   Do not fragment packets
  -f first_ttl --first=first_ttl
                              Start from the first_ttl hop (instead from 1)
  -g gate,... --gateway=gate,...
                              Route packets through the specified gateway
                              (maximum 8 for IPv4 and 127 for IPv6)
  -I --icmp   Use ICMP ECHO for tracerouting
  -T --tcp   Use TCP SYN for tracerouting (default port is 80)
  -i device --interface=device
                              Specify a network interface to operate with
  -m max_ttl --max-hops=max_ttl
                              Set the max number of hops (max TTL to be
                              reached). Default is 30
  -N squeries --sim-queries=squeries
                              Set the number of probes to be tried
                              simultaneously (default is 16)
  -n   Do not resolve IP addresses to their domain names
  -p port --port=port   Set the destination port to use. It is either
                              initial udp port value for "default" method
                              (incremented by each probe, default is 33434), or
                              initial seq for "icmp" (incremented as well,
                              default from 1), or some constant destination
                              port for other methods (with default of 80 for
                              "tcp", 53 for "udp", etc.)
  -t tos --tos=tos   Set the TOS (IPv4 type of service) or TC (IPv6

route

- 顯示目前的Routing table
- route –n　#不會自動將ip位址轉成文字
- route --help #route指令的詳細用法

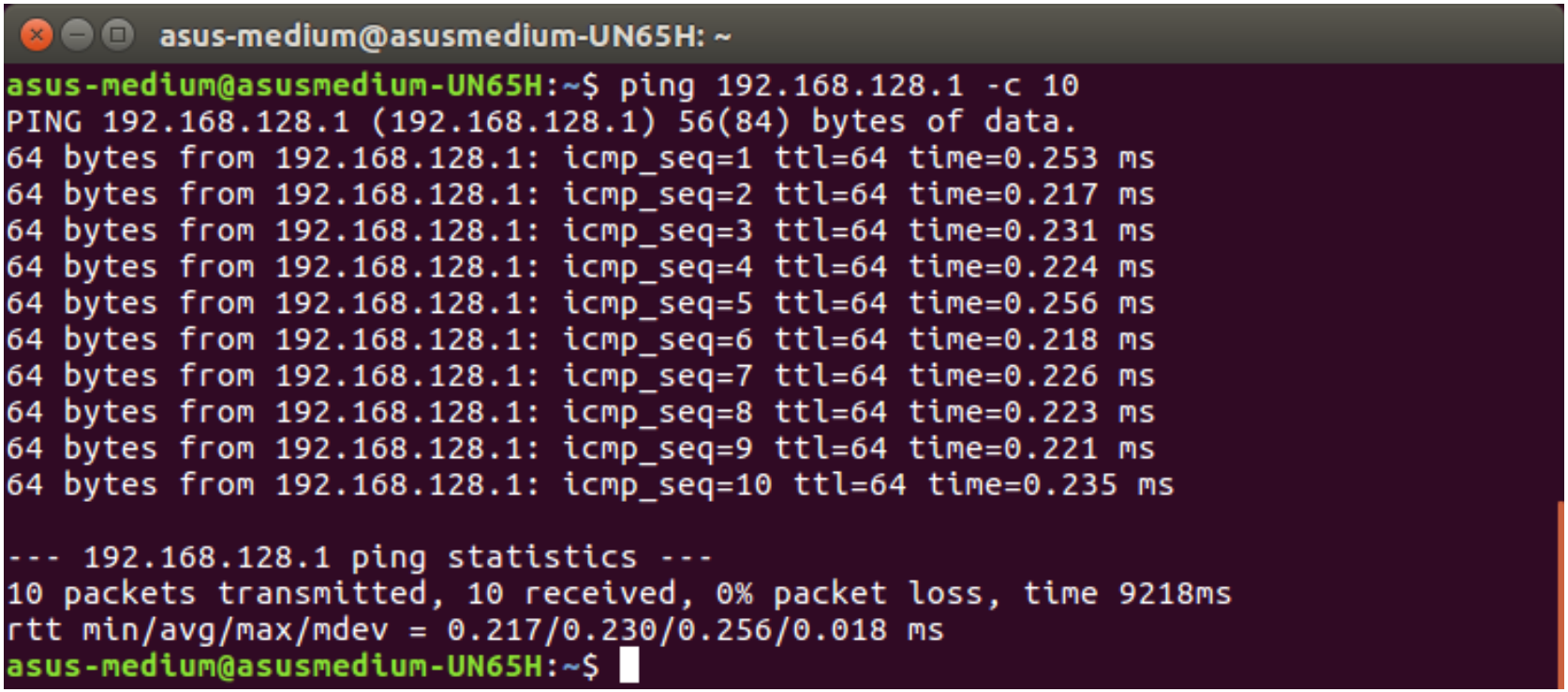


route

```
ue@ue-X580VD:~$ route --help
Usage: route [-nNvee] [-FC] [<AF>]           List kernel routing tables
       route [-v] [-FC] {add|del|flush} ...  Modify routing table for AF.

       route {-h|--help} [<AF>]              Detailed usage syntax for specified AF.
       route {-V|--version}                  Display version/author and exit.

        -v, --verbose            be verbose
        -n, --numeric            don't resolve names
        -e, --extend             display other/more information
        -F, --fib                display Forwarding Information Base (default)
        -C, --cache              display routing cache instead of FIB

  <AF>=Use '-A <af>' or '--<af>'; default: inet
  List of possible address families (which support routing):
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
ue@ue-X580VD:~$
```

ping

- 利用ICMP協定，發出ECHO_REQUEST到目的地
- ping dst –c 10 #發出10個封包

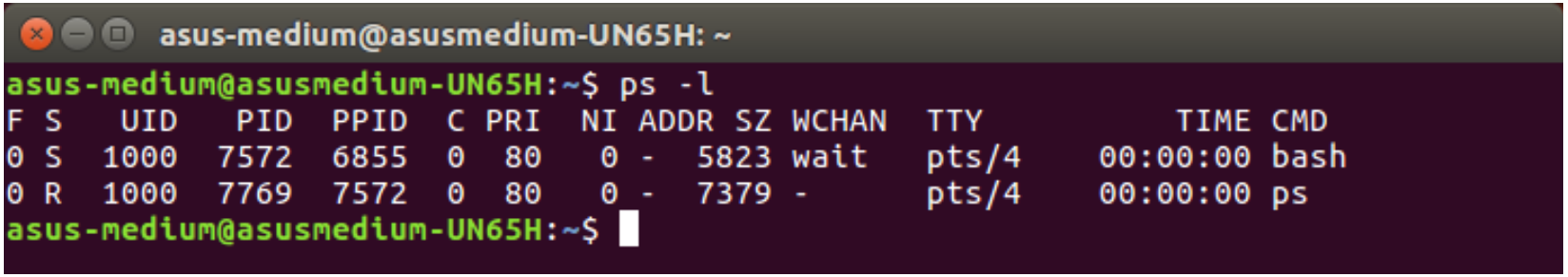


ping

```
ue@ue-X580VD:~$ ping --help
ping: invalid option -- '-'
Usage: ping [-aAbBdDfhLnOqrRUvV] [-c count] [-i interval] [-I interface]
            [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
            [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
            [-w deadline] [-W timeout] [hop1 ...] destination
ue@ue-X580VD:~$
```

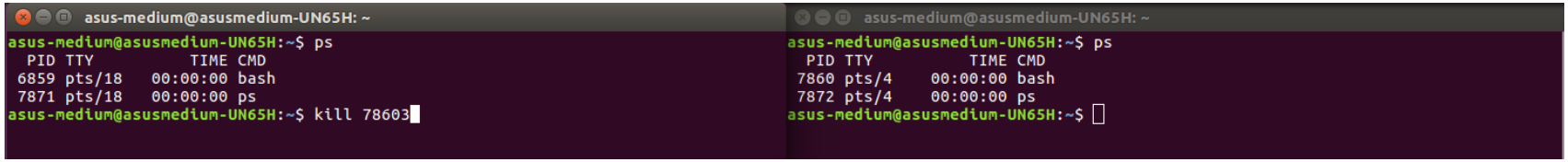
ps

- 查看執行中的程式指令
- ps -l　　#查看目前bash的程序
- ps aux　　#查看所有運行中的程序



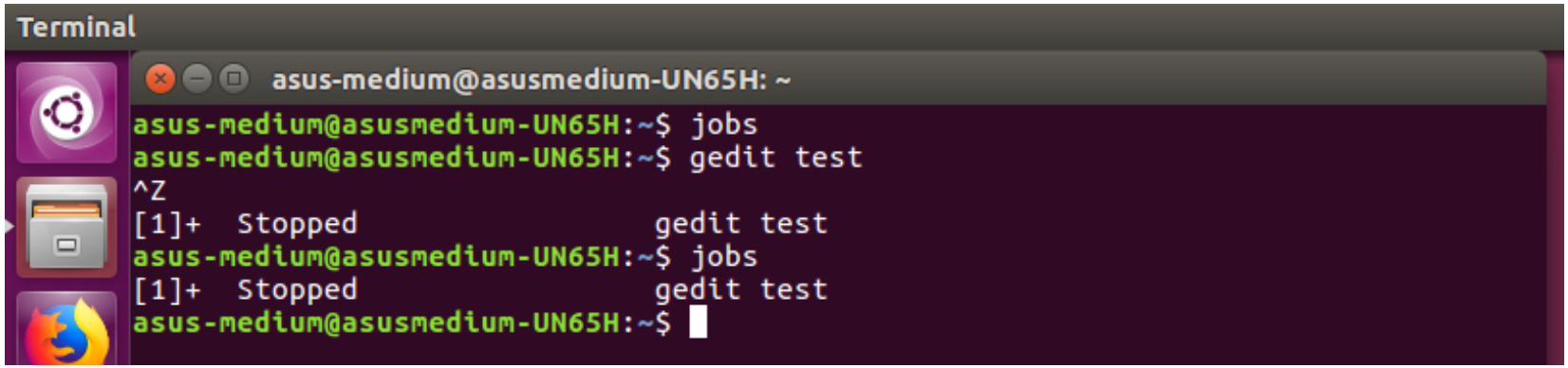
kill

- 結束執行中的程式的指令
- kill [PID]



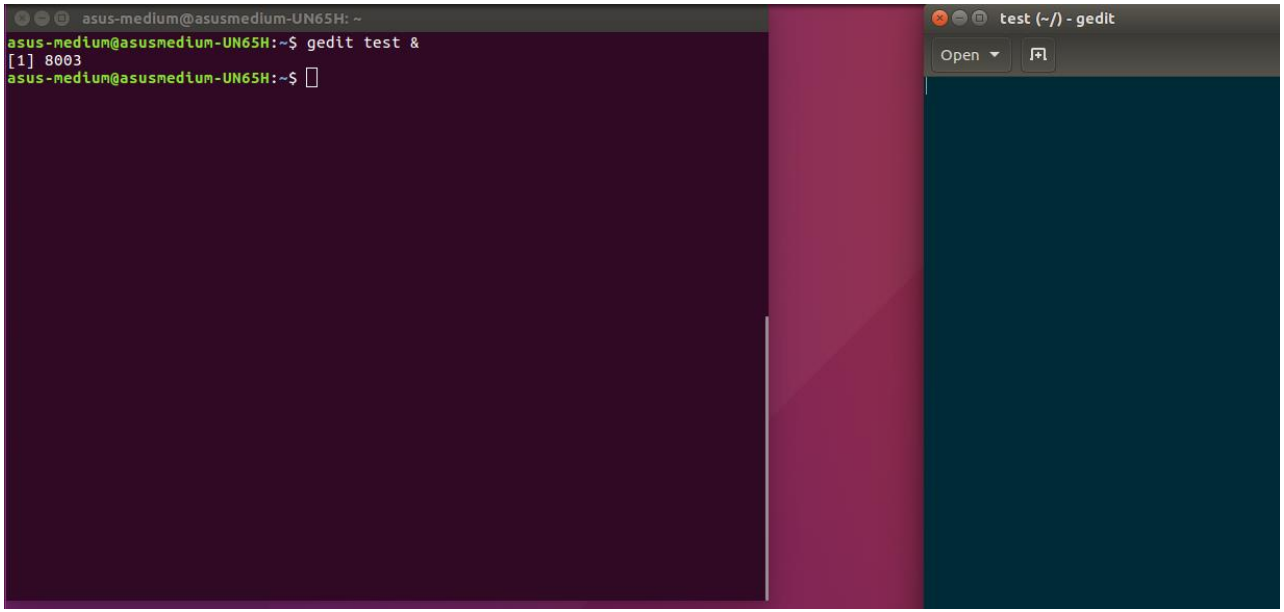
jobs

- 查看目前背景有哪些工作
- jobs　　#顯示工作狀態
- jobs －l #顯示工作狀態外，同時顯示PID



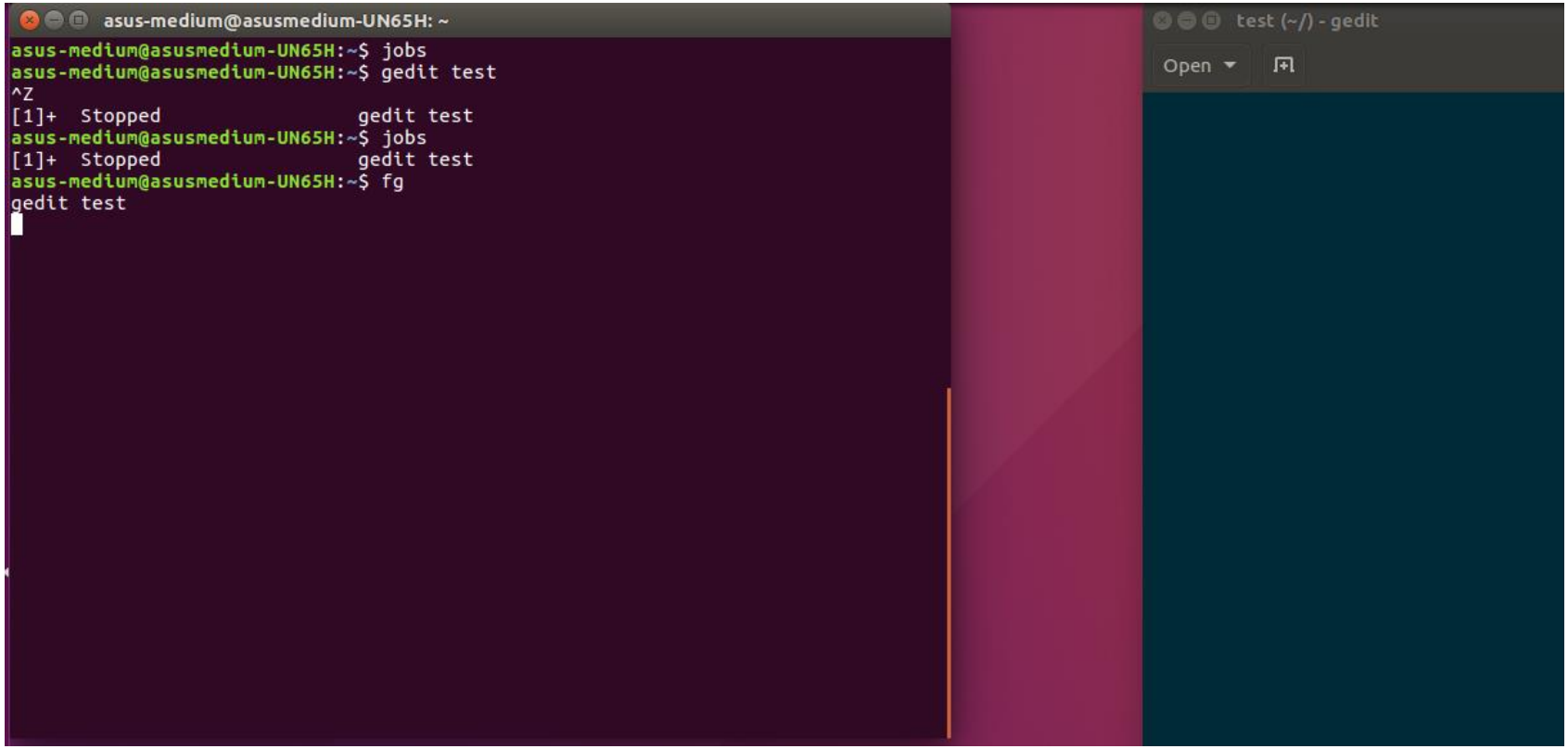
&,[Ctrl]+[Z]

- 在指令後面加上”&” ，代表將工作放在背景執行
  #工作狀態為”Running”
- 輸入完指令後，按下[Ctrl]+[Z]，代表把工作放置背景暫停
  #工作狀態為”Stopped”



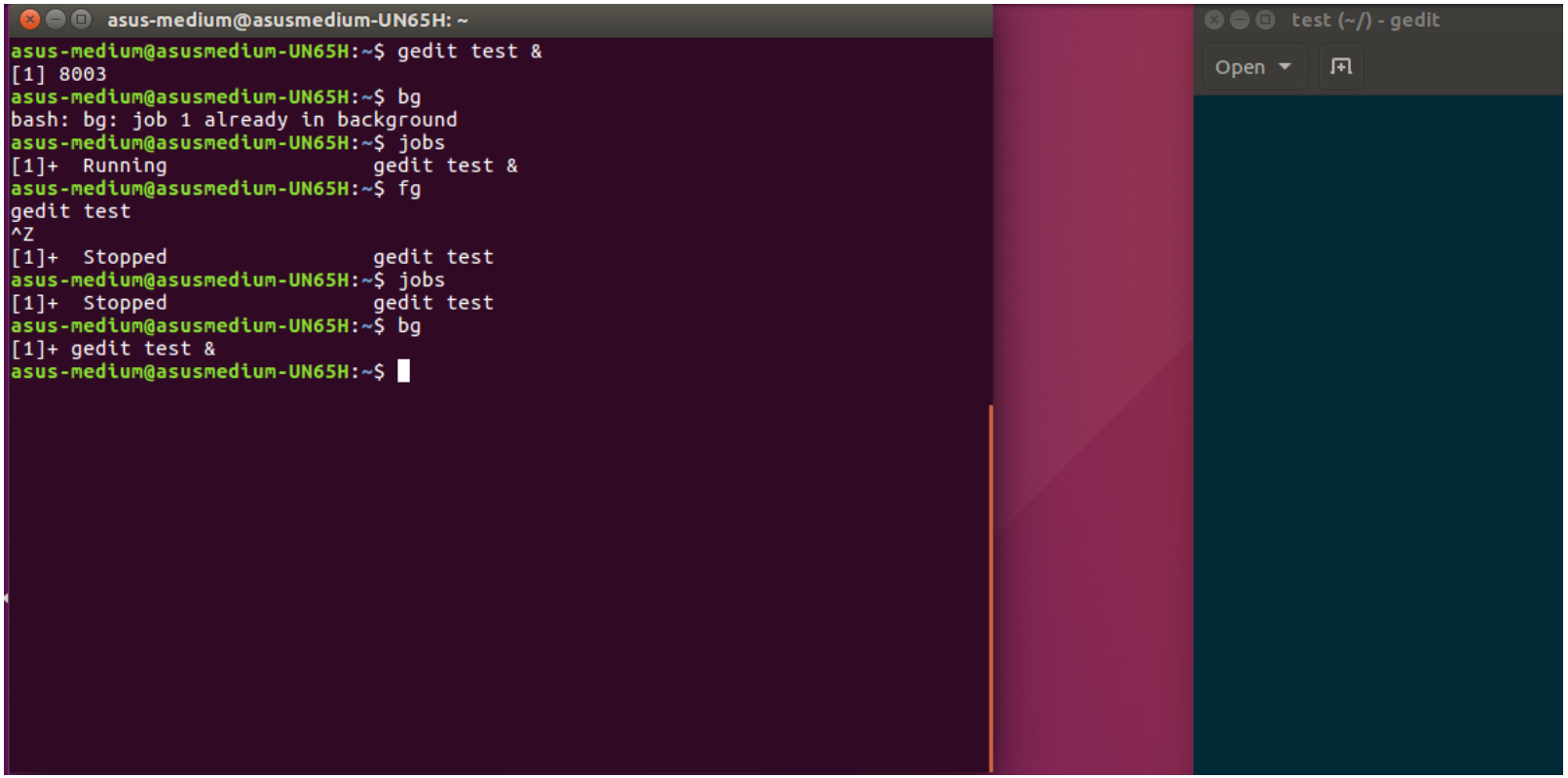
fg

- 將背景工作拿至前景處理



bg

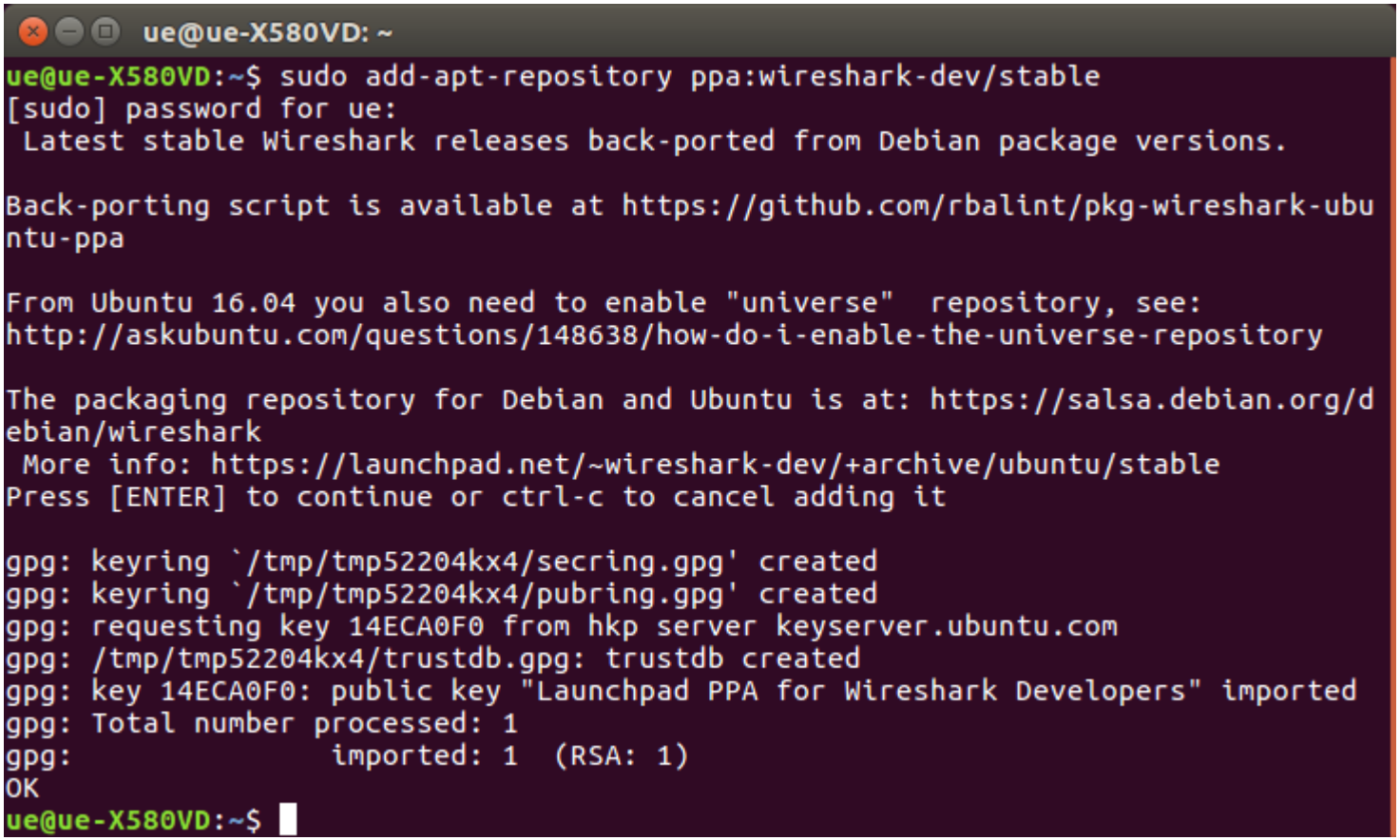
- 將目前執行的工作放置背景執行，效果等同指令後面加上”&”



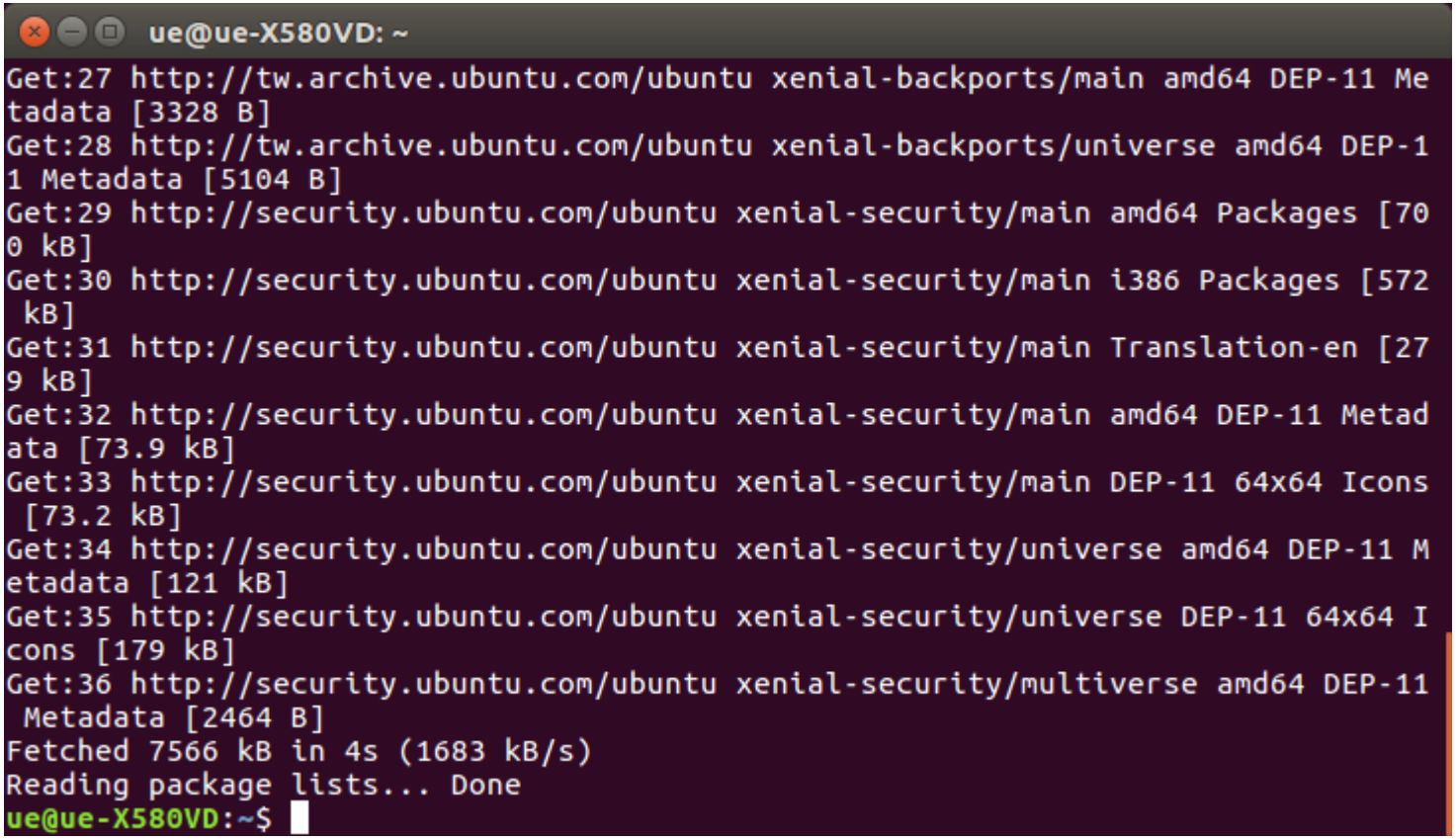
安裝Wireshark

- sudo add-apt-repository ppa:wireshark-dev/stable
- sudo apt-get update
- sudo apt-get install wireshark
- #若是依舊無法抓取網卡，請執行以下步驟
- $sudo adduser $USER wireshark
- #或是
- sudo groupadd wireshark
- sudo gpasswd -a $USER wireshark #re-login
- sudo chgrp wireshark /usr/bin/dumpcap
- sudo chmod o-rx /usr/sbin/dumpcap

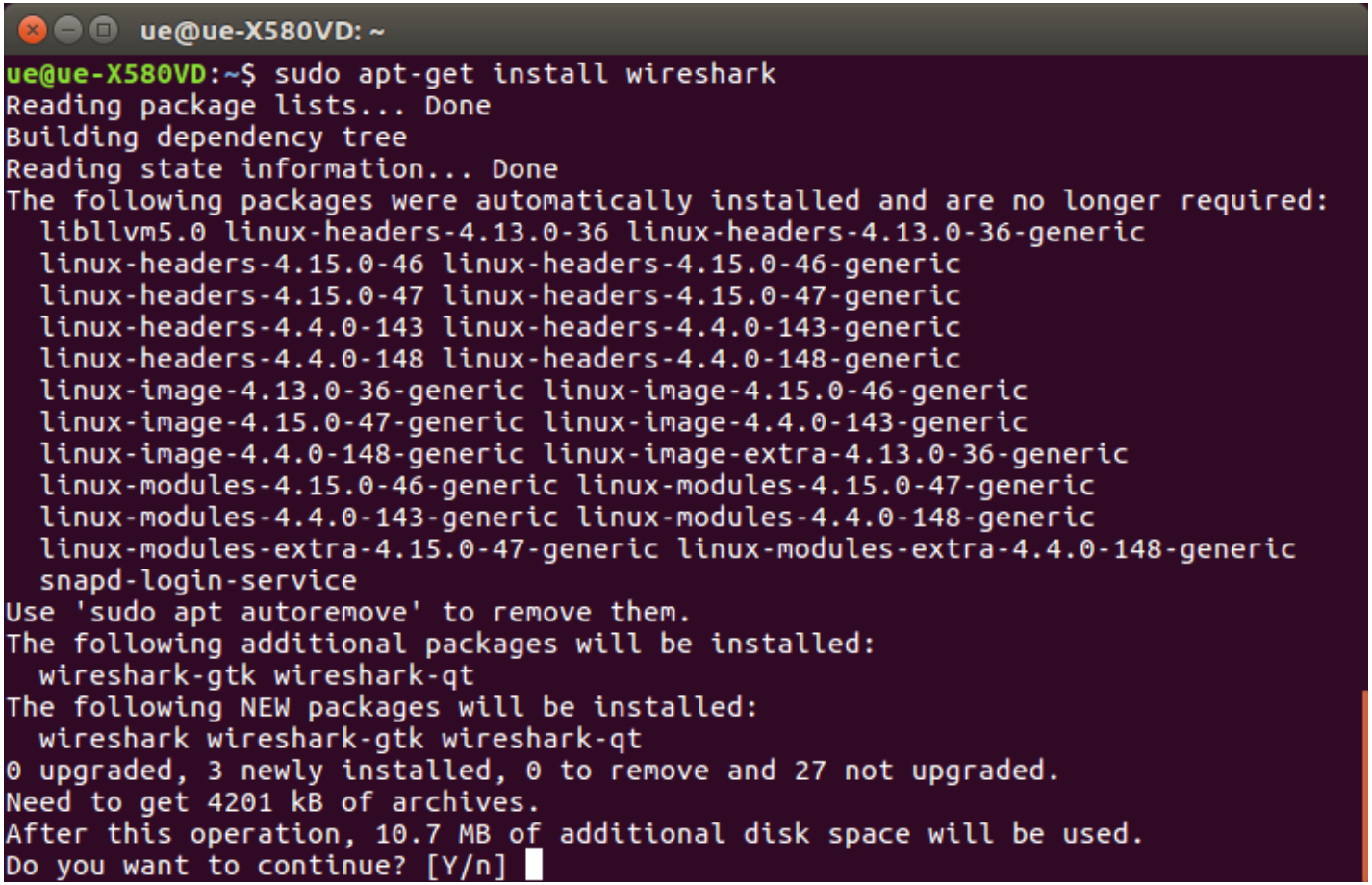
- sudo add-apt-repository ppa:wireshark-dev/stable



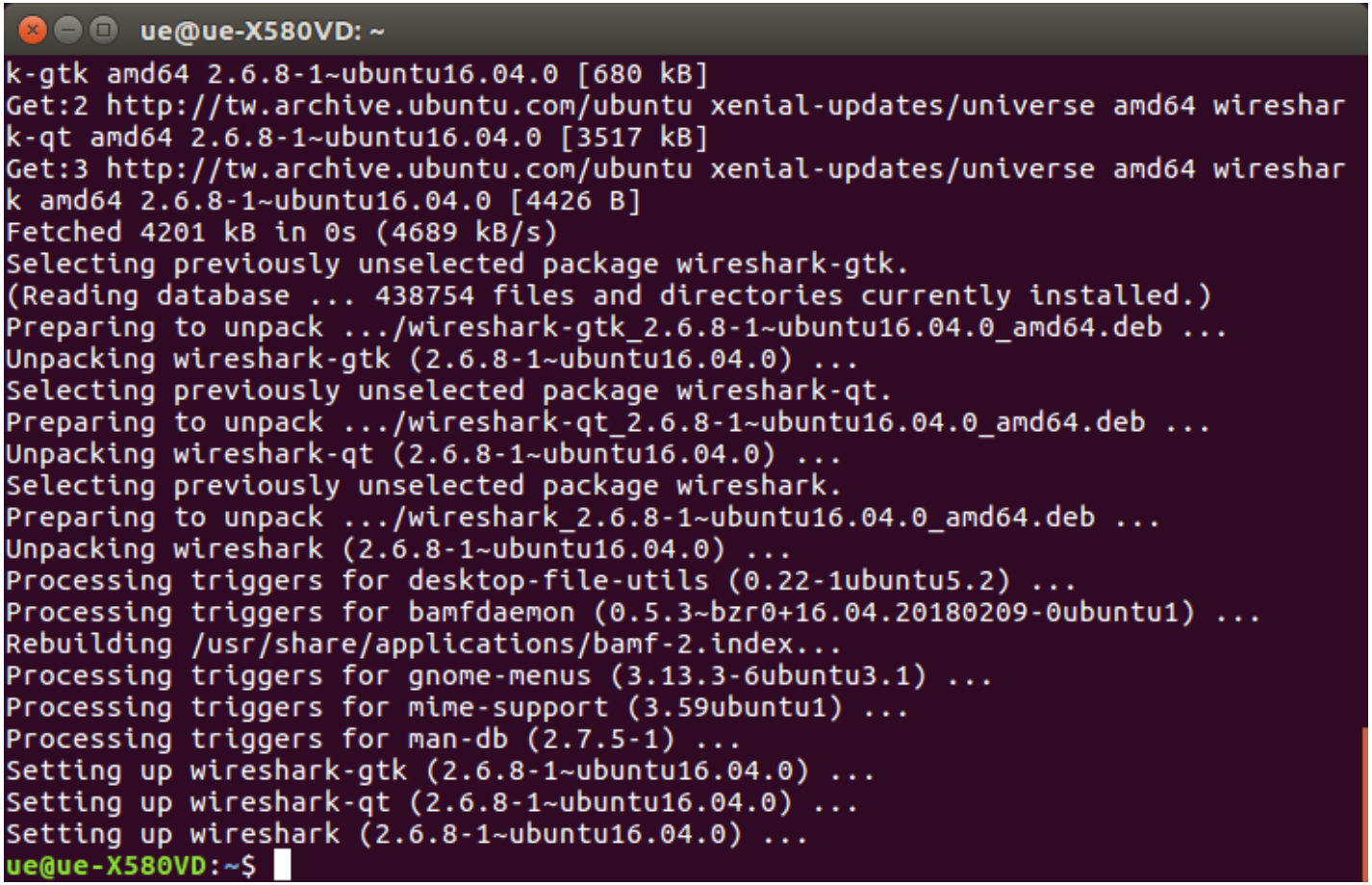
- sudo apt-get update



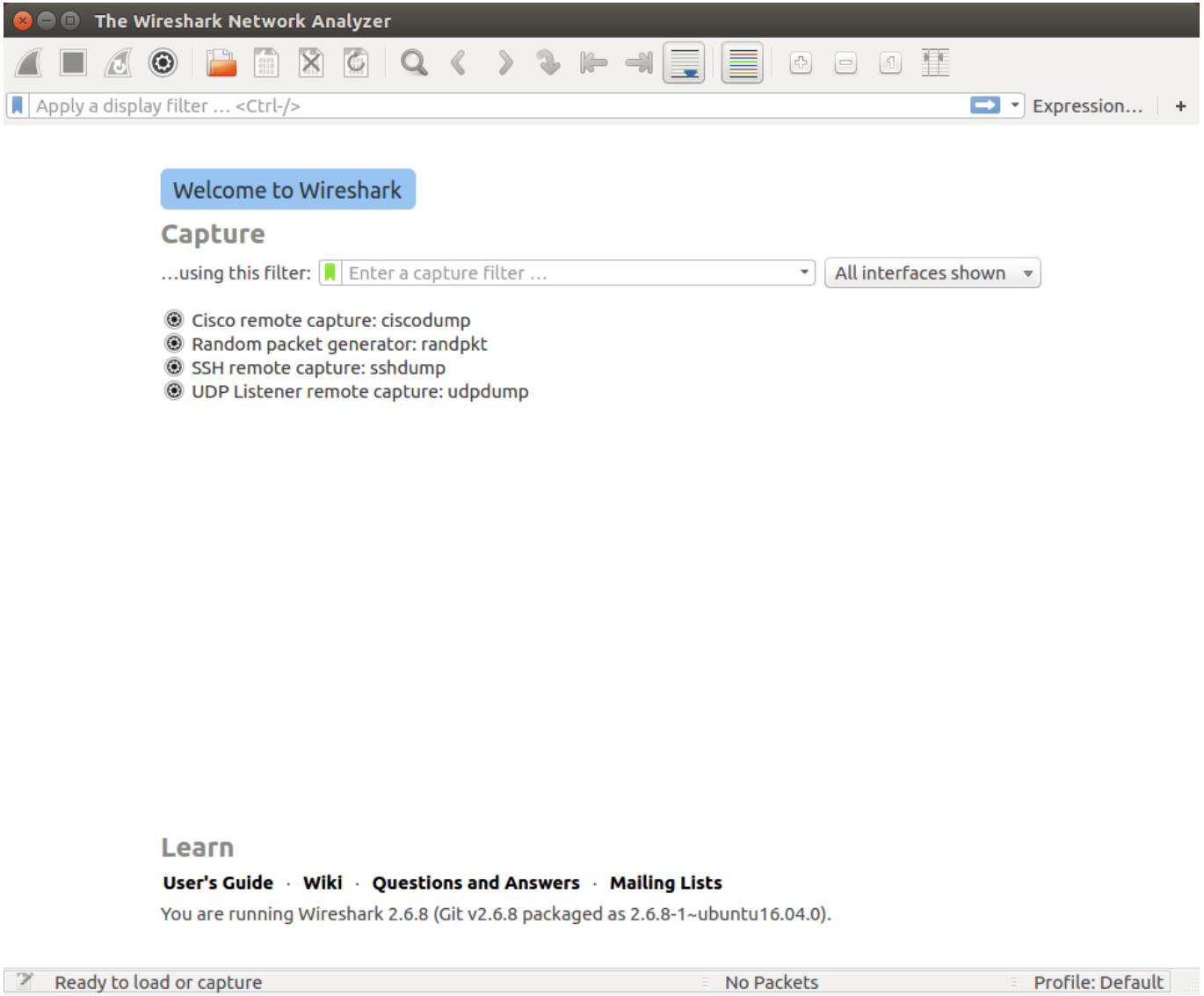
- sudo apt-get install wireshark



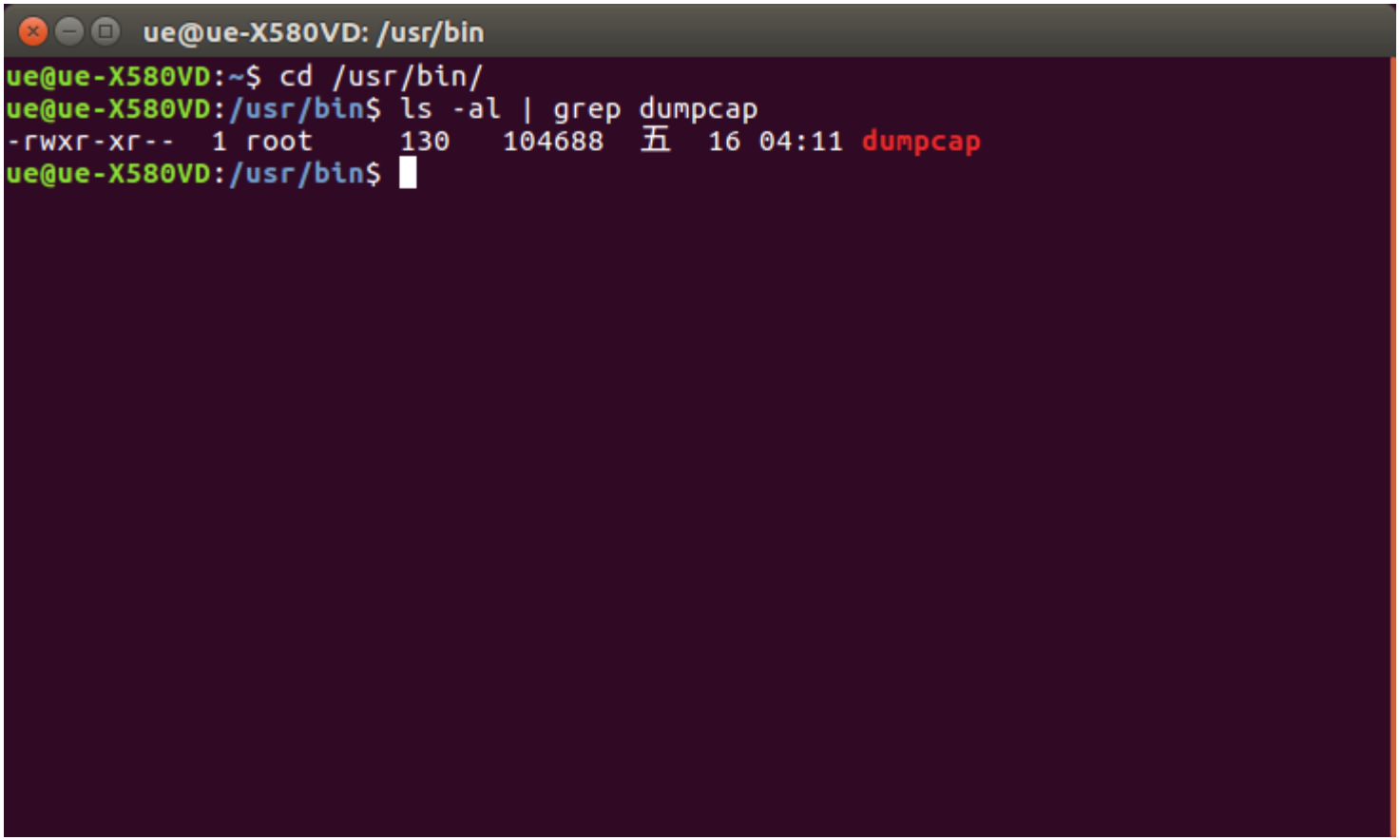
- 輸入“y”後按下enter，接著繼續



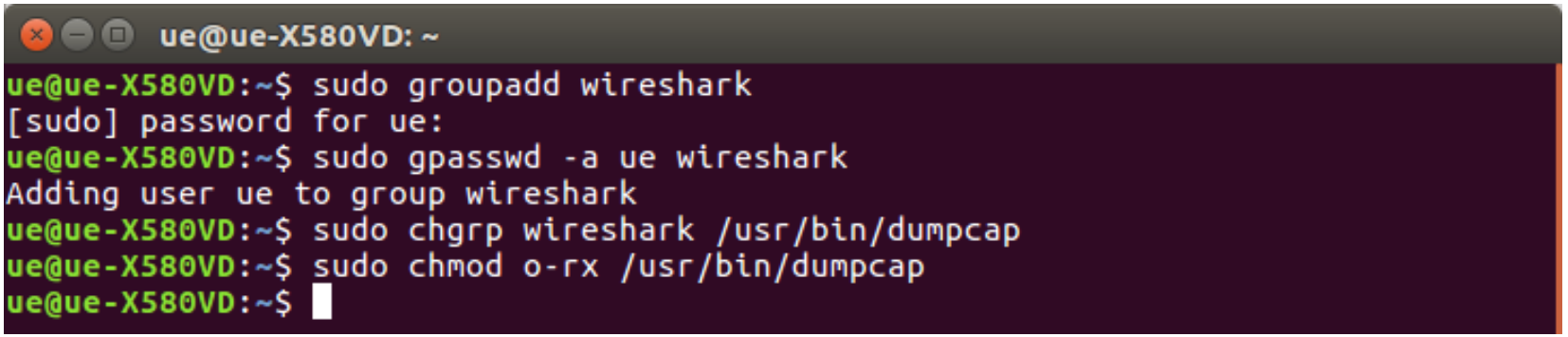
- 安裝完成後，重新登入，並開啟wireshark，
  若是無法正確顯示網路介面卡資訊，請按照以下步驟解決



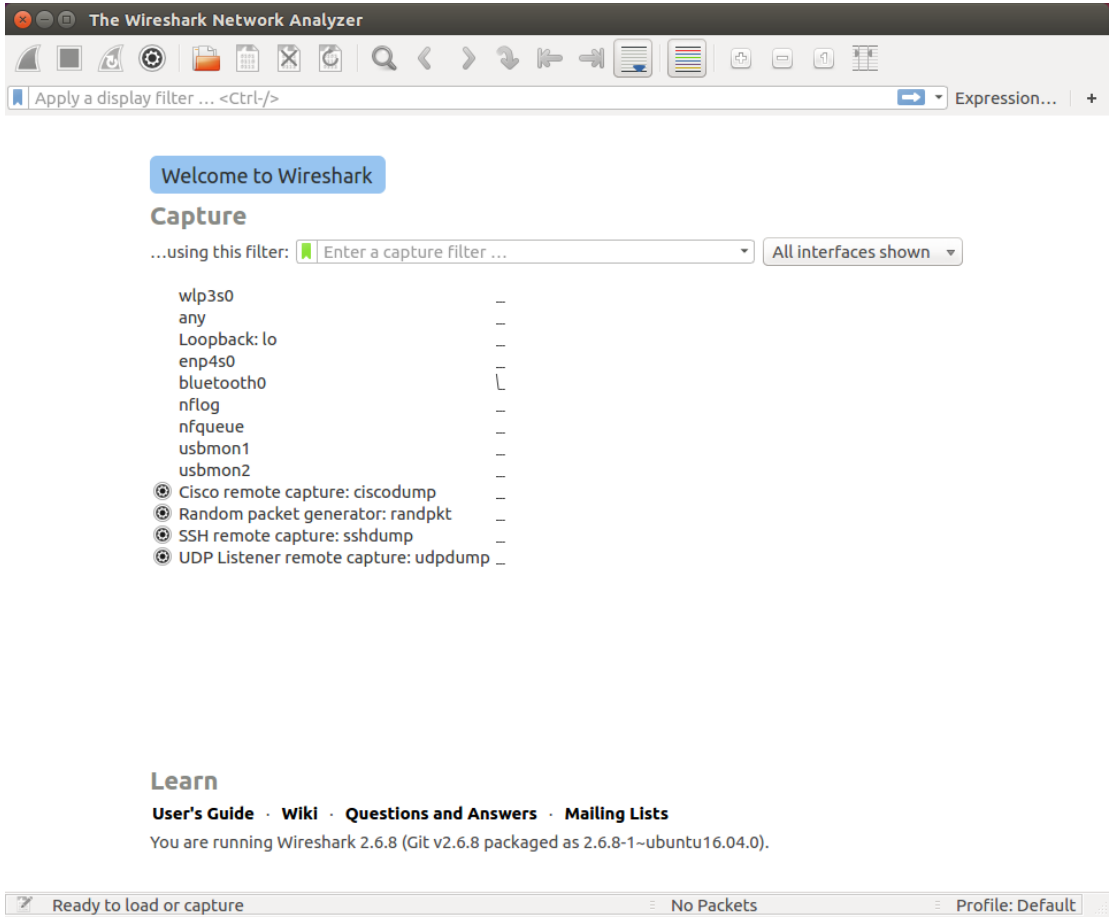
- cd 到 /usr/bin
- 查看 dumpcap 檔案的屬性



- sudo groupadd wireshark
- sudo gpasswd –a $USER wireshark #re-login
- sudo chgrp wireshark /usr/bin/dumpcap
- sudo chmod o-rx /usr/sbin/dumpcap
- #接著重新登入即可



- 若是開啟wireshark後，
  能夠顯示目前使用的網路介面卡，代表修改成功



Outline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- 基本 Linux 指令
  - 檔案相關指令
  - 網路相關指令
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
- 執行程式暨測試
- 總結

Install Packages

- sudo apt-get install cmake libfftw3-dev libboost-all-dev libconfig++-dev libsctp-dev

Install mbed TLS

- wget https://tls.mbed.org/download/start/mbedtls-2.16.0-apache.tgz
- tar zxvf mbedtls-2.16.0-apache.tgz
- sudo mv /path/to/mbedtls-2.16.0 /usr/local
- cd /usr/local/mbedtls-2.16.0
- cmake .
- make
- make test
- cmake -DENABLE_TESTING=Off .
- cmake -DUSE_SHARED_MBEDTLS_LIBRARY=On .
- sudo make install library

ref : https://tls.mbed.org/kb/compiling-and-building/how-do-i-build-compile-mbedtls

Install UHD

- sudo add-apt-repository ppa:ettusresearch/uhd
- sudo apt-get update
- sudo apt-get install libuhd-dev libuhd003 uhd-host
- sudo ./usr/lib/uhd/utils/uhd_images_downloader.py

ref: http://files.ettus.com/manual/page_install.html

Install SoapySDR

- sudo apt-get install cmake g++ libpython-dev python-numpy swig
- git clone https://github.com/pothosware/SoapySDR.git
- cd SoapySDR
- git pull origin master
- mkdir build
- cd build
- cmake ..
- make -j4
- sudo make install
- sudo ldconfig #needed on debian systems
- SoapySDRUtil --info

ref : https://github.com/pothosware/SoapySDR/wiki/BuildGuide#ubuntu

Install bladeRF

- sudo add-apt-repository ppa:bladerf/bladerf
- sudo apt-get update
- sudo apt-get install bladerf
- sudo apt-get install bladerf-firmware-fx3

ref : https://github.com/Nuand/bladeRF/wiki/Getting-Started%3A-Linux

Download and Build srsGUI

- sudo apt-get install libboost-system-dev libboost-test-dev libboost-thread-dev libqwt-dev libqt4-dev
- git clone https://github.com/srsLTE/srsGUI.git
- cd srsgui
- mkdir build
- cd build
- cmake ../
- make
- make test

ref : https://github.com/srslte/srsgui

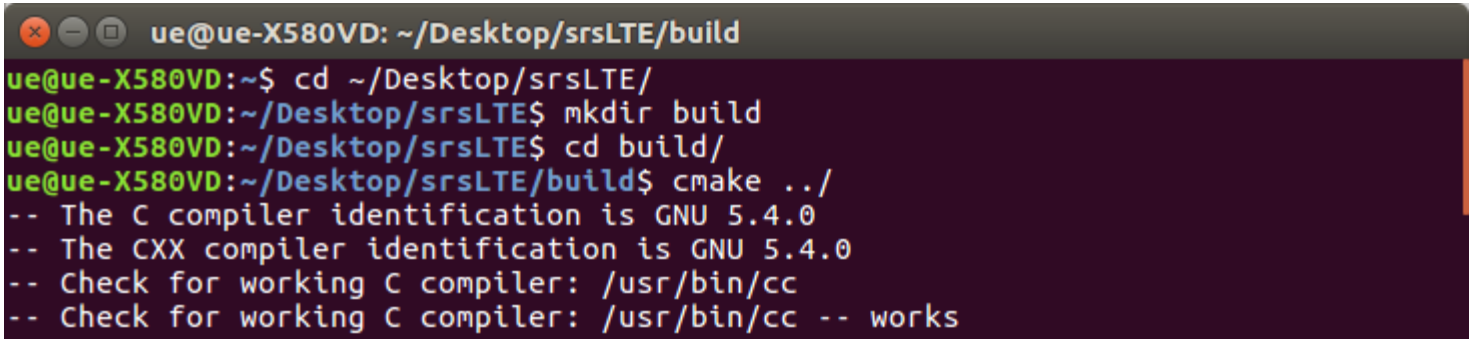
Download and Build srsLTE

- git clone https://github.com/srsLTE/srsLTE.git
- cd srsLTE
- mkdir build
- cd build
- cmake ../
- make
- make test
- sudo make install
- sudo srslte_install_configs.sh [service | user]

ref : https://github.com/srsLTE/srsLTE

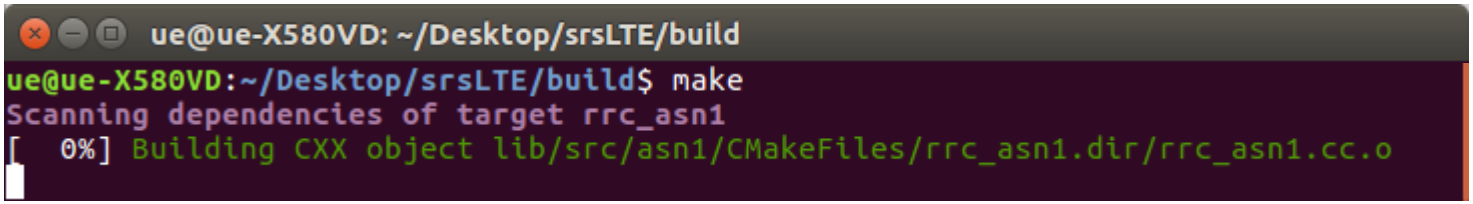
編譯 srsLTE - 1

- cd /path/to/srsLTE
- mkdir build
- cd build
- cmake ../



編譯 srsLTE - 2

- make



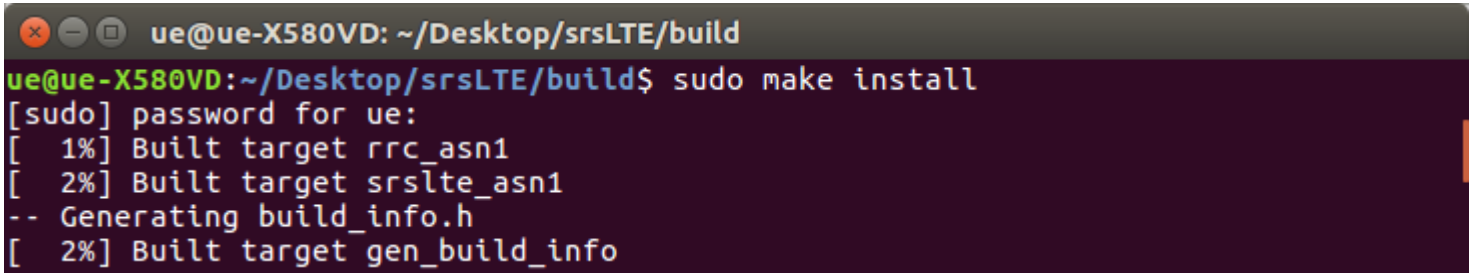
編譯 srsLTE - 3

- make test

```
asus-medium@asusmedium-UN65H: ~/Desktop/enb/build
[ 94%] Built target cell_search
Scanning dependencies of target usrp_capture_sync
[ 95%] Building C object lib/examples/CMakeFiles/usrp_capture_sync.dir/usrp_capture_sync.c.o
[ 95%] Linking C executable usrp_capture
[ 95%] Linking C executable usrp_capture_sync
[ 95%] Built target usrp_capture
Scanning dependencies of target srsue
[ 95%] Building CXX object srsue/src/CMakeFiles/srsue.dir/main.cc.o
[ 95%] Built target usrp_capture_sync
Scanning dependencies of target mac_test
[ 95%] Building CXX object srsue/test/mac/CMakeFiles/mac_test.dir/mac_test.cc.o
[ 96%] Linking CXX executable mac_test
[ 97%] Linking CXX executable srsmbms
[ 97%] Built target mac_test
Scanning dependencies of target srsenb
[ 97%] Built target srsmbms
[ 97%] Building CXX object srsue/src/CMakeFiles/srsue.dir/ue_base.cc.o
[ 97%] Building CXX object srsenb/src/CMakeFiles/srsenb.dir/main.cc.o
[ 97%] Linking CXX executable srsepc
[ 97%] Built target srsepc
Scanning dependencies of target ip_test_enb
[ 97%] Building CXX object srsenb/test/upper/CMakeFiles/ip_test_enb.dir/ip_test.cc.o
[ 98%] Building CXX object srsue/src/CMakeFiles/srsue.dir/ue.cc.o
[ 99%] Linking CXX executable ip_test_enb
[ 99%] Built target ip_test_enb
Scanning dependencies of target benchmark_radio
[ 99%] Building CXX object lib/src/radio/test/CMakeFiles/benchmark_radio.dir/benchmark_radio.cc.o
[ 99%] Linking CXX executable benchmark_radio
[ 99%] Built target benchmark_radio
[ 99%] Building CXX object srsenb/src/CMakeFiles/srsenb.dir/enb.cc.o
[ 99%] Building CXX object srsue/src/CMakeFiles/srsue.dir/metrics_stdout.cc.o
[ 99%] Building CXX object srsue/src/CMakeFiles/srsue.dir/metrics_csv.cc.o
[100%] Building CXX object srsenb/src/CMakeFiles/srsenb.dir/parser.cc.o
[100%] Linking CXX executable srsue
[100%] Building CXX object srsenb/src/CMakeFiles/srsenb.dir/enb_cfg_parser.cc.o
[100%] Built target srsue
[100%] Building CXX object srsenb/src/CMakeFiles/srsenb.dir/metrics_stdout.cc.o
[100%] Linking CXX executable srsenb
[100%] Built target srsenb
asus-medium@asusmedium-UN65H:~/Desktop/enb/build$ make test
```

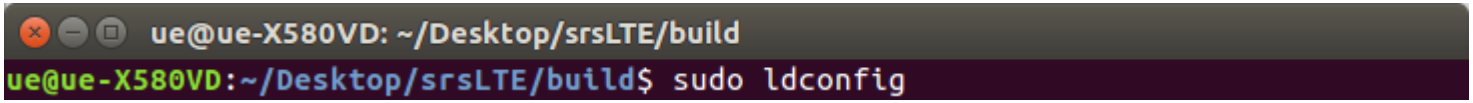
編譯 srsLTE - 4

- sudo make install



編譯 srsLTE - 5

- sudo ldconfig



編譯 srsLTE - 6

- sudo srslte_install_configs.sh [user | service]
- #user install all config files to $HOME/.config/srslte
  #service install all config files to /etc/srslte

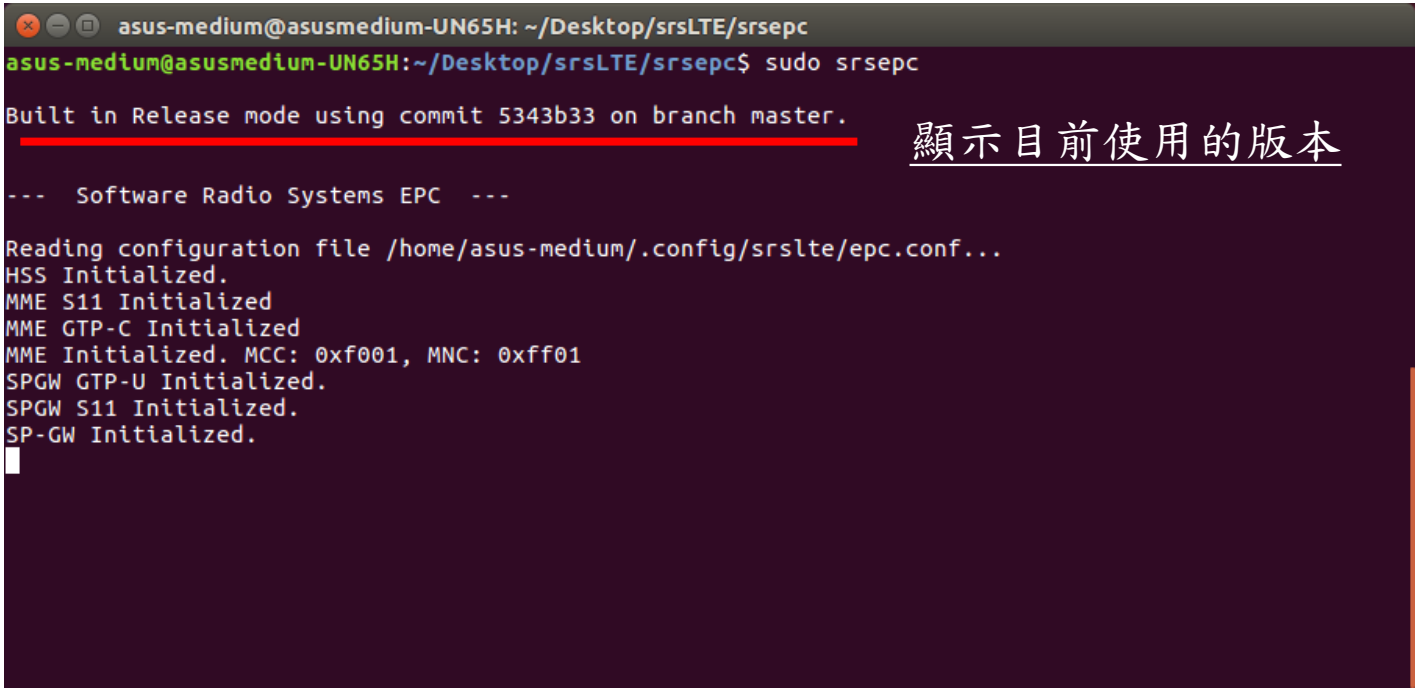
```
ue@ue-X580VD: ~/Desktop/srsLTE/build
ue@ue-X580VD:~/Desktop/srsLTE/build$ sudo srslte_install_configs.sh user
Installing srsLTE configuration files:
 - Creating srsLTE config folder /home/ue/.config/srslte
 - Installing ue.conf.example in /home/ue/.config/srslte/ue.conf
 - Installing enb.conf.example in /home/ue/.config/srslte/enb.conf
 - Installing sib.conf.example in /home/ue/.config/srslte/sib.conf
 - Installing rr.conf.example in /home/ue/.config/srslte/rr.conf
 - Installing drb.conf.example in /home/ue/.config/srslte/drb.conf
 - Installing epc.conf.example in /home/ue/.config/srslte/epc.conf
 - Installing mbms.conf.example in /home/ue/.config/srslte/mbms.conf
 - Installing user_db.csv.example in /home/ue/.config/srslte/user_db.csv
Done.
ue@ue-X580VD:~/Desktop/srsLTE/build$
```

Outline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- 基本 Linux 指令
  - 檔案相關指令
  - 網路相關指令
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
- 執行程式暨測試
- 總結

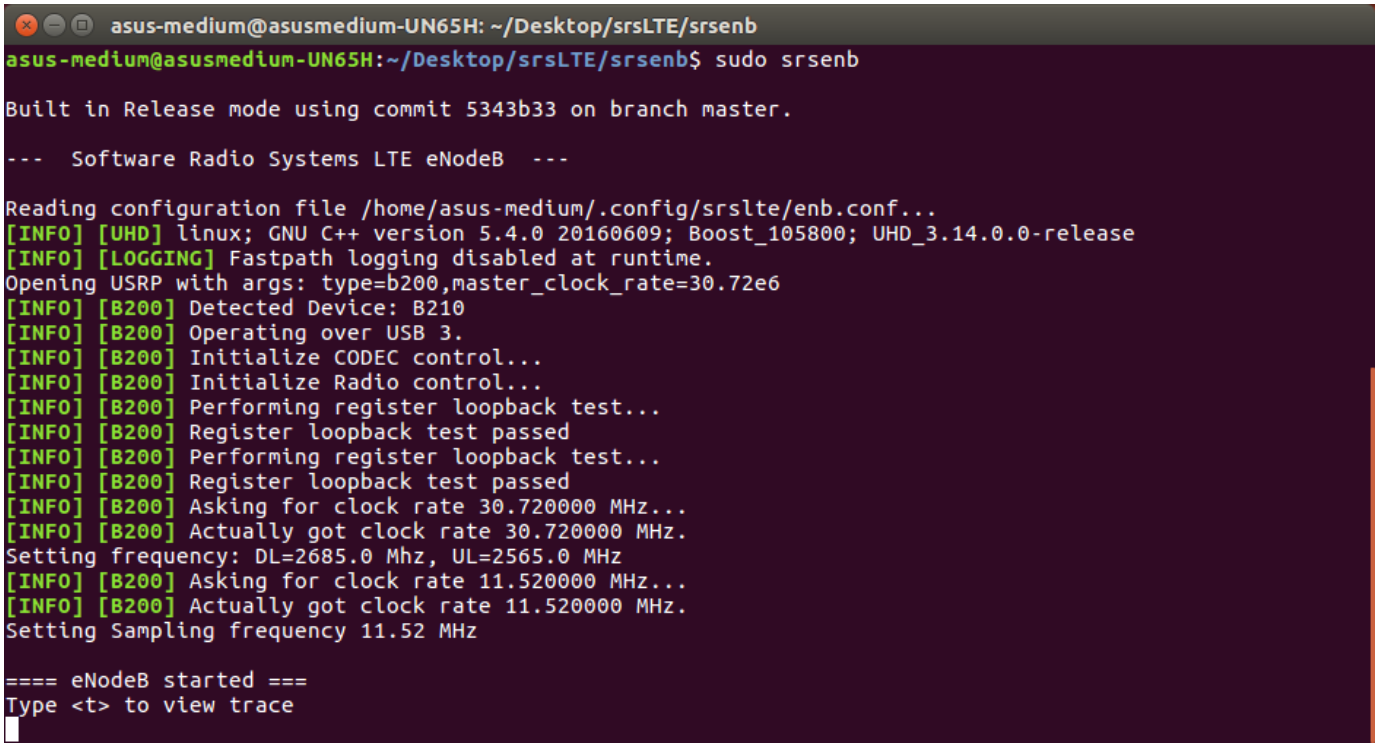
執行程式–EPC

- 順序 ： epc>enb>ue
- cd /path/to/srsLTE/srsepc
- sudo srsepc　　#terminal_1



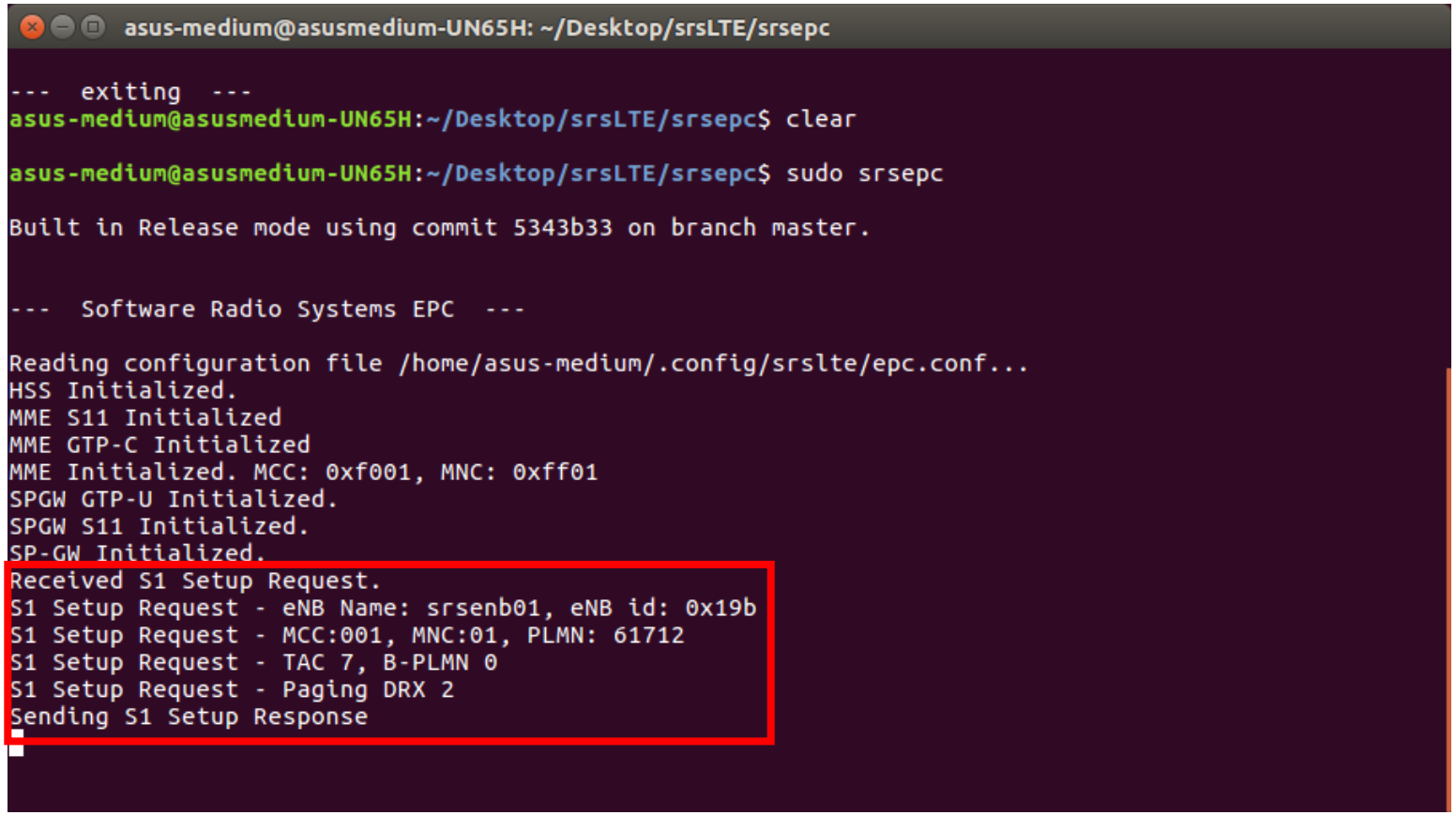
執行程式–eNB

- cd /path/to/srsLTE/srsenb
- sudo srsenb　　#terminal_2



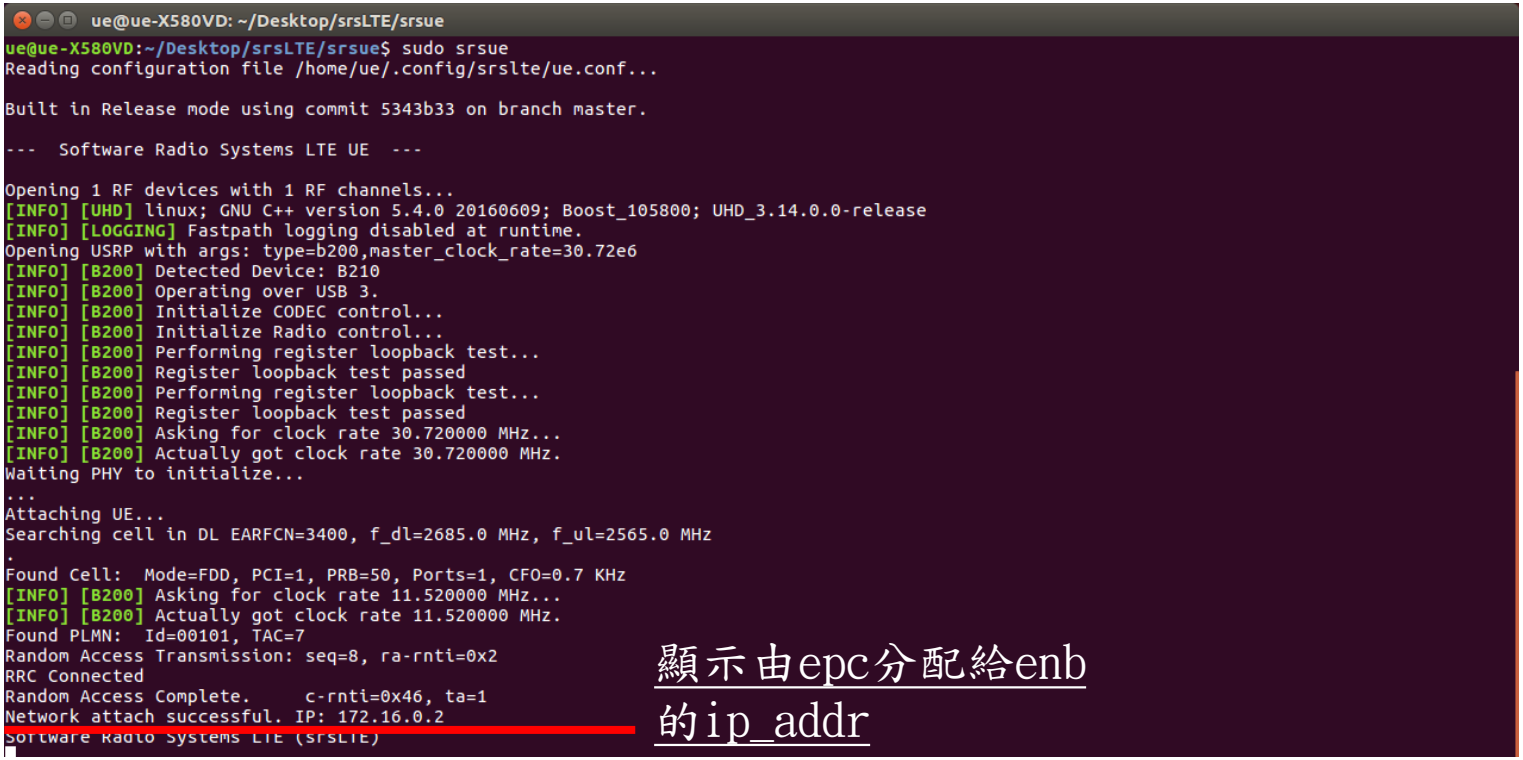
執行程式

- 在啟動eNB後，EPC會接著顯示S1連線的相關資訊



執行程式-UE

- cd /path/to/srsLTE/srsue
- sudo srsue



執行程式

- 當UE成功連線後，EPC及eNB皆會顯示相關的連線資訊

```
asus-medium@asusmedium-UN65H: ~/Desktop/srsLTE/srsepc
Sending S1 Setup Response
Initial UE message: LIBLTE_MME_MSG_TYPE_ATTACH_REQUEST
Received Initial UE message -- Attach Request
Attach request -- GUTI Style Attach request
Attach request -- M-TMSI: 0x2f9f5a6b
Attach request -- eNB-UE S1AP Id: 1
Attach request -- Attach type: 1
Attach Request -- UE Network Capabilities EEA: 11100000
Attach Request -- UE Network Capabilities EIA: 01100000
Attach Request -- MS Network Capabilities Present: false
PDN Connectivity Request -- EPS Bearer Identity requested: 0
PDN Connectivity Request -- Procedure Transaction Id: 1
PDN Connectivity Request -- ESM Information Transfer requested: false
UL NAS: Received Identity Response
ID Response -- IMSI: 001010123456789
Downlink NAS: Sent Authentication Request
UL NAS: Received Authentication Response
Authentication Response -- IMSI 001010123456789
UE Authentication Accepted.
Generating KeNB with UL NAS COUNT: 0
Downlink NAS: Sending NAS Security Mode Command.
UL NAS: Received Security Mode Complete
Security Mode Command Complete -- IMSI: 001010123456789
Getting subscription information -- QCI 7
Sending Create Session Request.
Creating Session Response -- IMSI: 1010123456789
Creating Session Response -- MME control TEID: 1
Received GTP-C PDU. Message type: GTPC_MSG_TYPE_CREATE_SESSION_REQUEST
SPGW: Allocated Ctrl TEID 1
SPGW: Allocated User TEID 1
SPGW: Allocate UE IP 172.16.0.2
Received Create Session Response
Create Session Response -- SPGW control TEID 1
Create Session Response -- SPGW S1-U Address: 127.0.1.100
SPGW Allocated IP 172.16.0.2 to IMSI 001010123456789
Adding attach accept to Initial Context Setup Request
Initial Context Setup Request -- eNB UE S1AP Id 1, MME UE S1AP Id 1
Initial Context Setup Request -- E-RAB id 5
Initial Context Setup Request -- S1-U TEID 0x1. IP 127.0.1.100
Initial Context Setup Request -- S1-U TEID 0x1. IP 127.0.1.100
Initial Context Setup Request -- QCI 7
Received Initial Context Setup Response
E-RAB Context Setup. E-RAB id 5
E-RAB Context -- eNB TEID 0x460003; eNB GTP-U Address 127.0.1.1
UL NAS: Received Attach Complete
Unpacked Attached Complete Message. IMSI 1010123456789
Unpacked Activate Default EPS Bearer message. EPS Bearer id 5
Received GTP-C PDU. Message type: GTPC_MSG_TYPE_MODIFY_BEARER_REQUEST
Sending EMM Information
```

```
==== eNodeB started ===
Type <t> to view trace
RACH:  tti=8341, preamble=6, offset=1, temp_crnti=0x46
User 0x46 connected
```

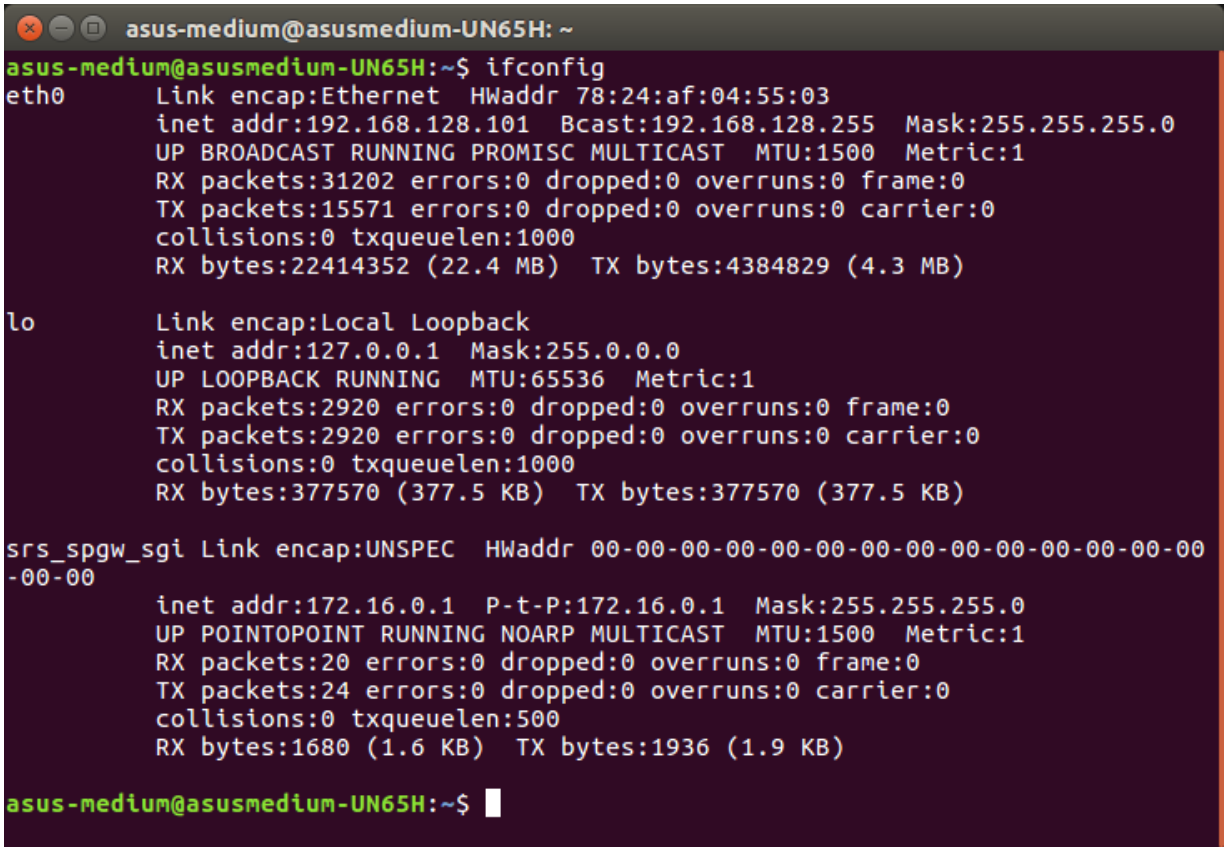
檢查環境

- 利用netstat指令，確認srsLTE程式是否有開啟
- sudo netstat –alpWn | grep srs

```
asus-medium@asusmedium-UN65H:~$ sudo netstat -alepWn | grep srs
udp        0      0 127.0.1.1:2152          0.0.0.0:*                           0          1698722     5337/srsenb

udp        0      0 127.0.1.100:2152        0.0.0.0:*                           0          1700872     5267/srsepc

unix  2      [ ]         DGRAM                    1700867   5267/srsepc         @mme_s11@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
unix  2      [ ]         DGRAM                    1700873   5267/srsepc         @spgw_s11@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
asus-medium@asusmedium-UN65H:~$
```

檢查環境

- ifconfig　　# epc's terminal command
- epc 開啟後會產生新的網卡介面：srs_spgw_sgi



- epc 根據epc.conf設定裡的sgi_if_name 產生新的網卡介面：srs_spgw_sgi

epc.conf (~/.config/srslte) - gedit

Open Save

```
#####################################################################
# SP-GW configuration
#
# gtpu_bind_addr:   GTP-U bind address.
# sgi_if_addr:      SGi TUN interface IP address.
# sgi_if_name:      SGi TUN interface name.
# max_paging_queue: Maximum packets in paging queue (per UE).
#
#####################################################################

[spgw]
gtpu_bind_addr   = 127.0.1.100
sgi_if_addr      = 172.16.0.1
sgi_if_name      = srs_spgw_sgi
max_paging_queue = 100

#####################################################################
# PCAP configuration
#
# Packets are captured to file in the compact format decoded by
# the Wireshark s1ap dissector and with DLT 150.
# To use the dissector, edit the preferences for DLT_USER to
# add an entry with DLT=150, Payload Protocol=s1ap.
#
# enable:   Enable or disable the PCAP.
# filename: File name where to save the PCAP.
#
#####################################################################
[pcap]
enable   = false
filename = /tmp/epc.pcap
```

Matlab Tab Width: 8 Ln 1, Col 1 INS

檢查環境

- ifconfig　　# ue's terminal command
- 成功建立連線後，UE會產生新的網卡介面：tun_srsue

```
ue@ue-X580VD: ~
ue@ue-X580VD:~$ ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:605 errors:0 dropped:0 overruns:0 frame:0
          TX packets:605 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:49266 (49.2 KB)  TX bytes:49266 (49.2 KB)

tun_srsue Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:172.16.0.2  P-t-P:172.16.0.2  Mask:255.255.255.0
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:840 (840.0 B)  TX bytes:840 (840.0 B)
```

- ue 根據ue.conf設定裡的 ip_devname
  產生新的網卡介面：tun_srsue

ue.conf (~/.config/srslte) - gedit

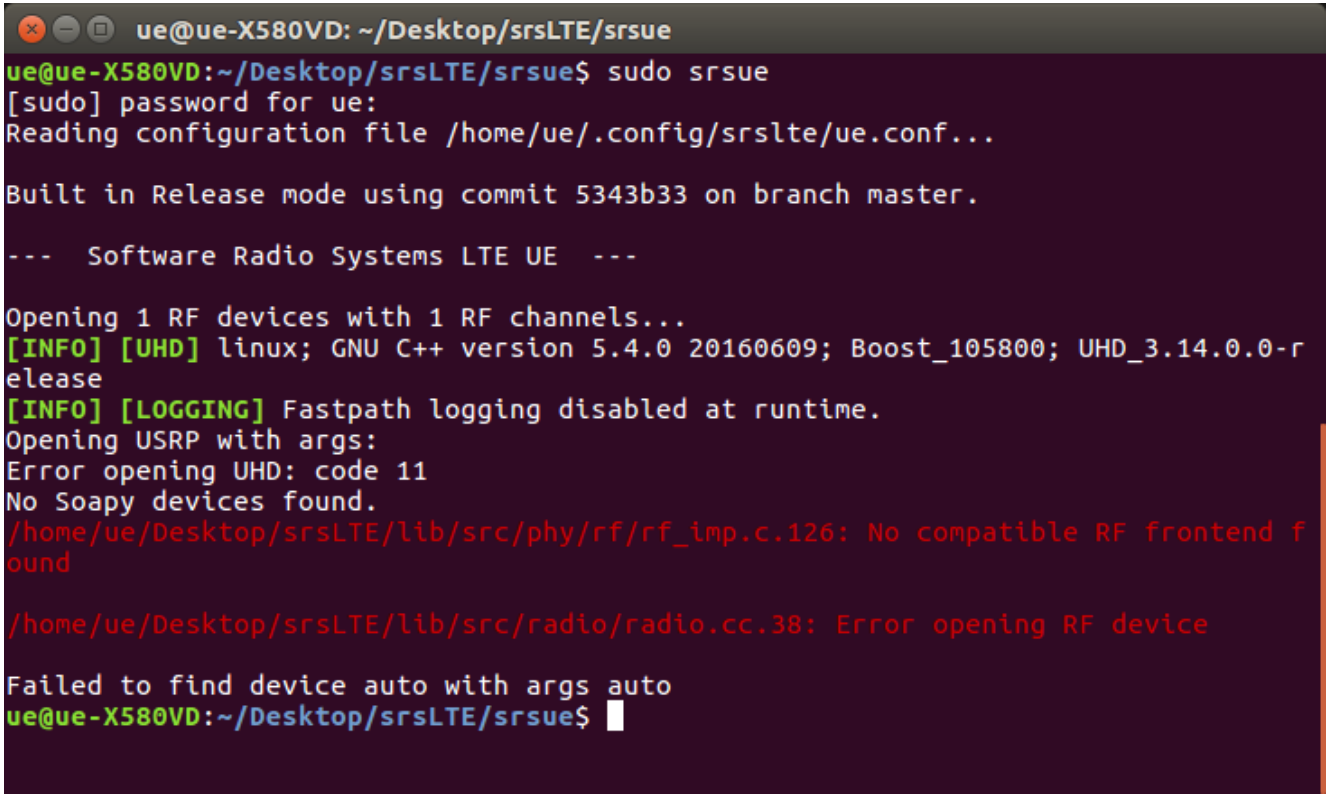
Open Save

```
#
# pdsch_8bit_decoder:    Use 8-bit for LLR representation and turbo decoder trellis computation
(Experimental)
#
#####################################################################
[expert]
#ip_netmask           = 255.255.255.0
#ip_devname           = tun_srsue
#mbms_service         = -1
#rssi_sensor_enabled = false
#rx_gain_offset       = 62
#prach_gain           = 30
#cqi_max              = 15
#cqi_fixed            = 10
#snr_ema_coeff        = 0.1
#snr_estim_alg        = refs
#pdsch_max_its        = 8    # These are half iterations
#nof_phy_threads      = 3
#equalizer_mode       = mmse
#sfo_ema              = 0.1
#sfo_correct_period   = 10
#sss_algorithm        = full
#estimator_fil_auto   = false
#estimator_fil_stddev  = 1.0
#estimator_fil_order   = 4
#snr_to_cqi_offset    = 0.0
#interpolate_subframe_enabled = false
#sic_pss_enabled      = true
#pregenerate_signals = false
#metrics_csv_enable   = false
#metrics_period_secs = 1
```

Matlab Tab Width: 8 Ln 223, Col 70 INS

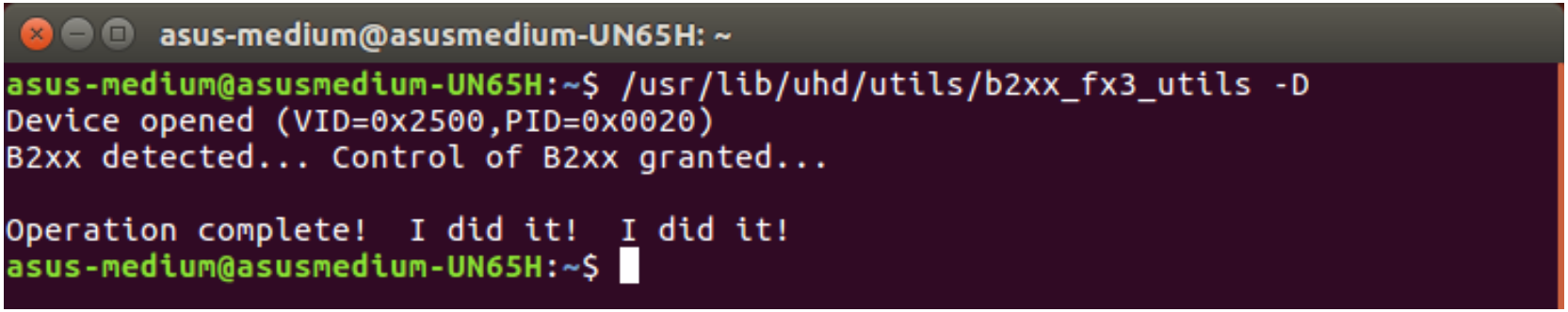
啟動錯誤

- ue或是enb啟動時，需要連接usrp，
  有時候可能因為接觸不良的關係，導致系統無法偵測到usrp，
- 請重新拔插usb接頭。※請確保電腦的usb孔支援3.0



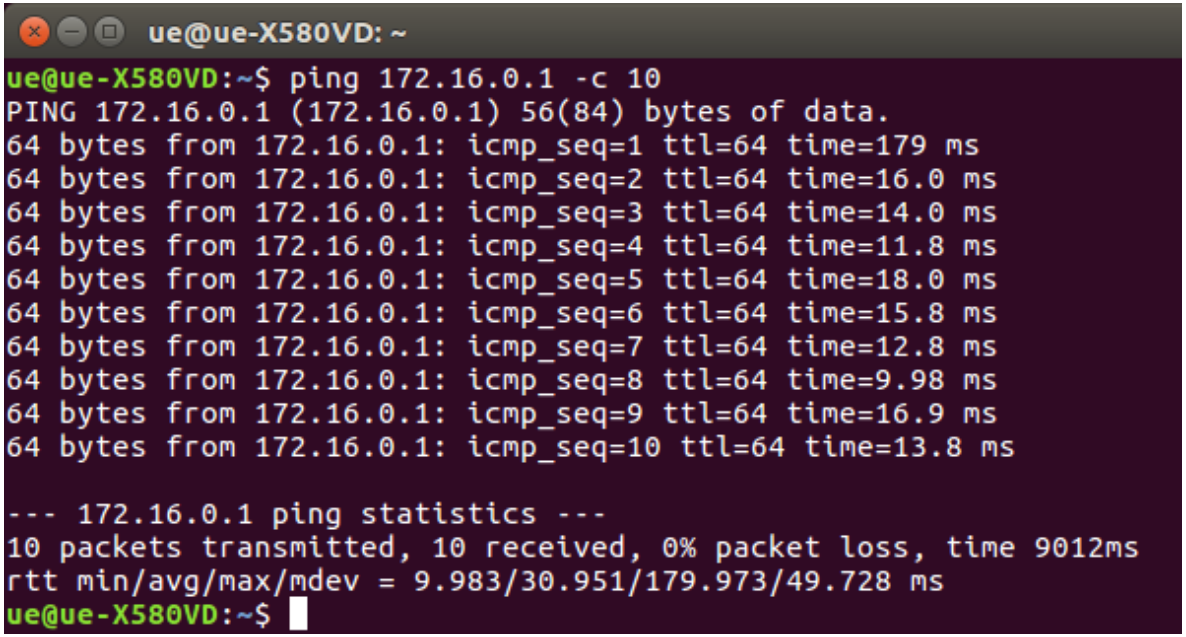
偵測 USRP

- 利用以下指令，讓電腦偵測是否可以讀取usrp
- uhd_find_devices #brief info
- uhd_usrp_probe #detail info
- 使用下列指令，清除電腦之前讀取的相關檔案
- /usr/lib/uhd/utils/b2xx_fx3_utils -D



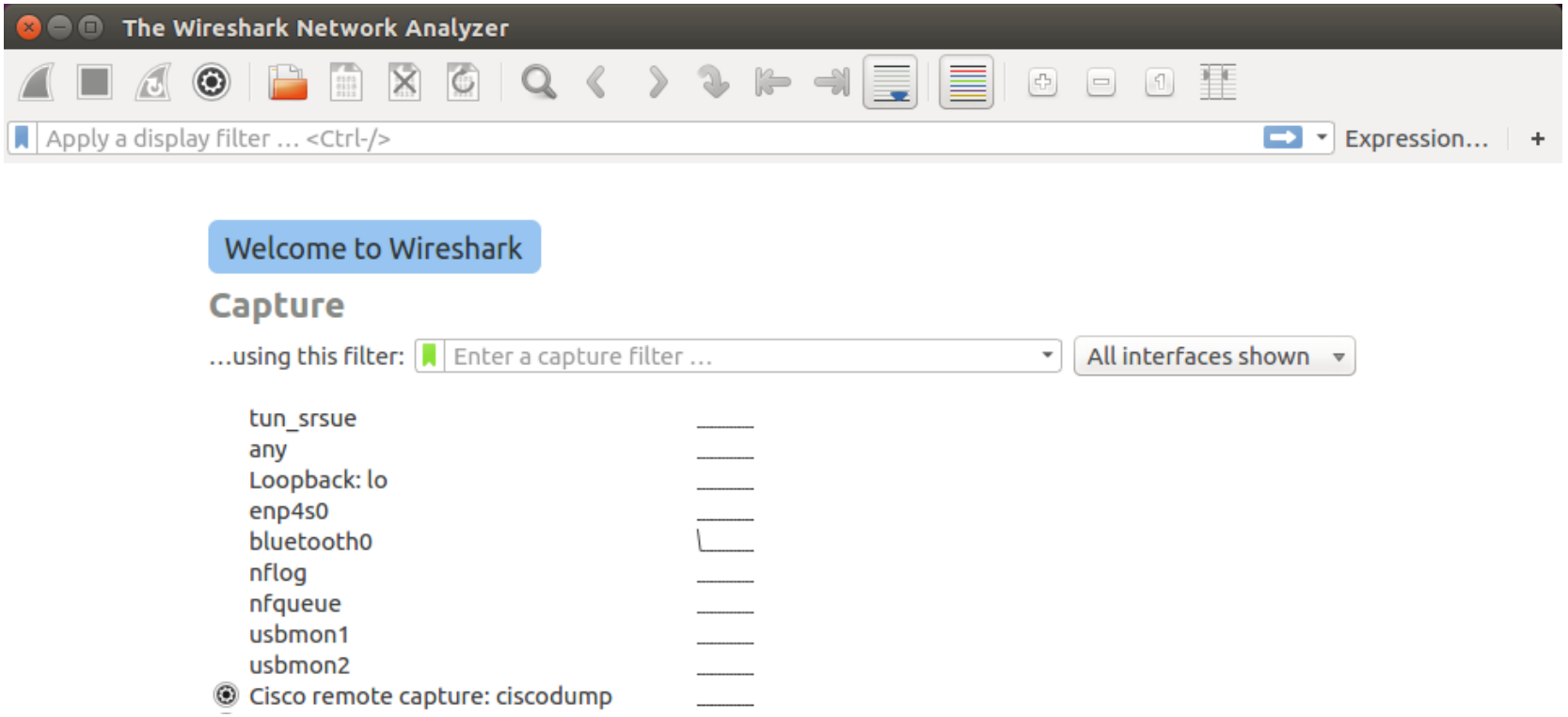
互通測試

- 根據epc設定，預設的ip為 172.16.0.1
- ue 在terminal 輸入 ping 172.16.0.1 –c 10
- 若是有ICMP回覆，則代表建置成功



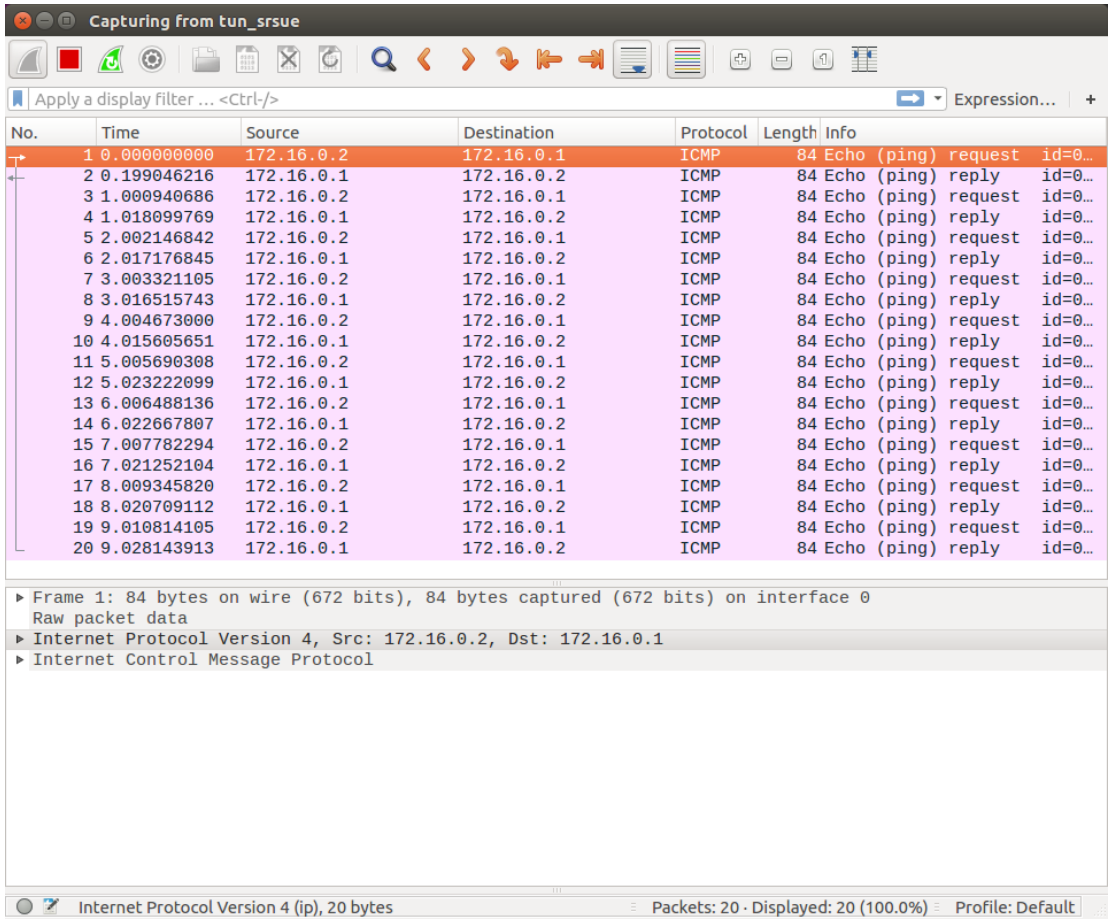
Wireshark介面選擇

- 一開始會出現目前電腦現有的網路卡介面，請挑選欲觀察的網路卡介面



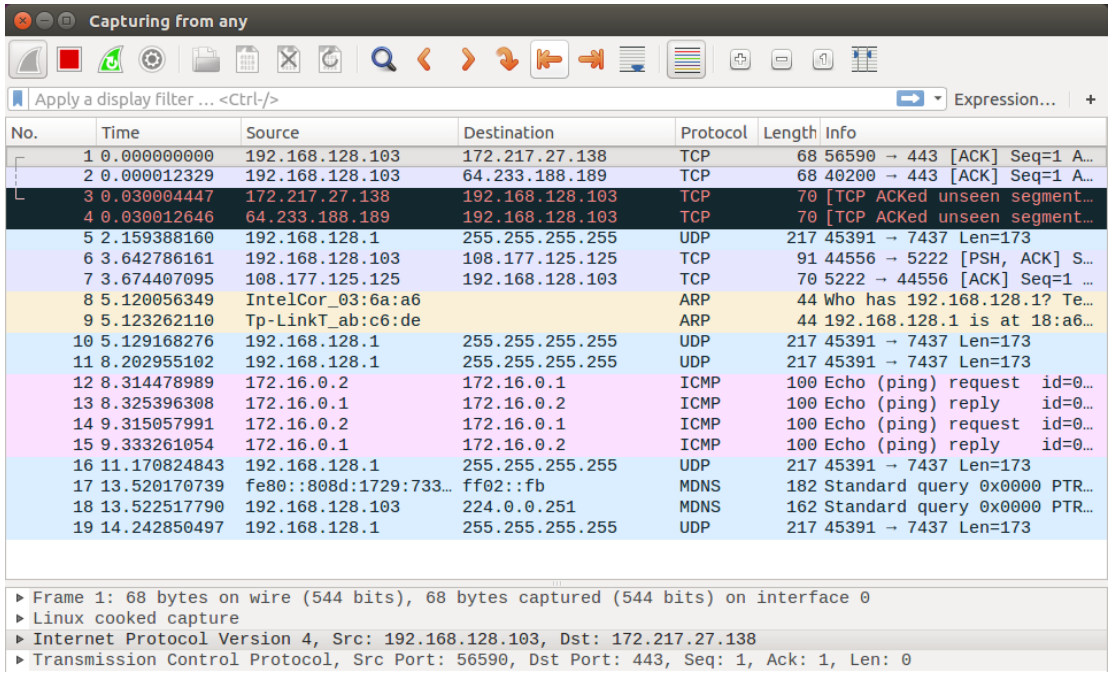
Wireshark-srs介面

- 選擇tun_srsue介面後，再利用ping指令觀察到ue跟enb的封包收送



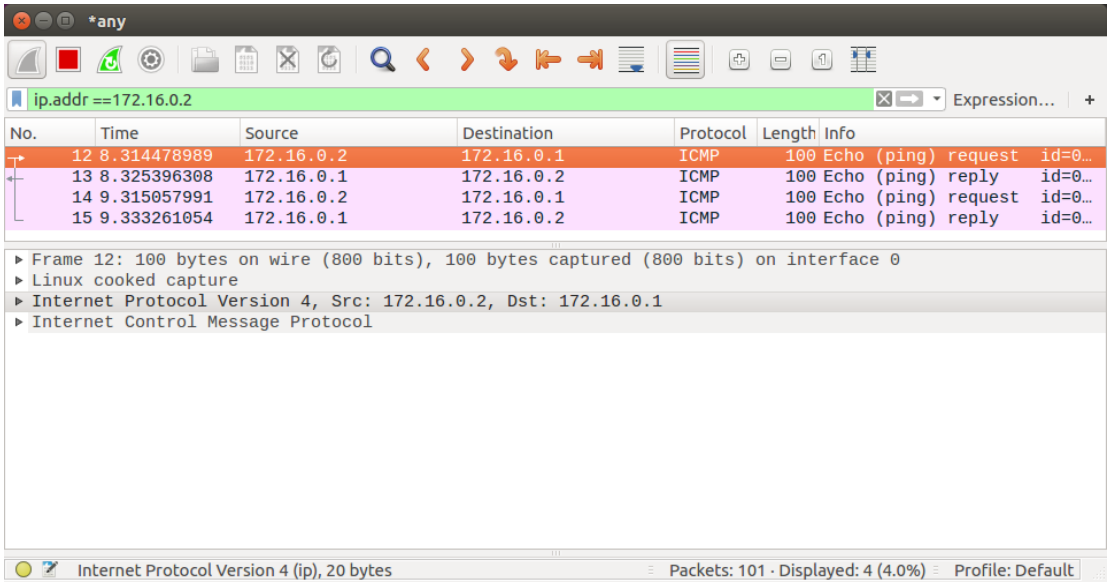
Wireshark-any介面

- 若是選擇any介面，會將電腦所有封包皆顯示，
  不僅顯示自己的封包，同時可能會顯示同一個區域網路的封包



Wireshark-filter

- 利用filter，過濾出自己想要看的封包
- ip.addr == 172.16.0.2 #符合ip位址為172.16.0.2的封包
- ip.dst == 172.16.0.1 #符合ip目的位址為172.16.0.1的封包
- ip.src == 172.16.0.2 #符合ip來源位址為172.16.0.2的封包
- tcp #符合協定為tcp的封包
- udp #符合協定為udp的封包
- eth.addr == 12:34:56:78:90:aa #符合MAC位址的封包



Outline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- 基本 Linux 指令
  - 檔案相關指令
  - 網路相關指令
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
- 執行程式暨測試
- 總結

總結

- 讓學生熟悉基本 Linux 的環境
- 使學生知道如何透過指令執行動作
- 讓學生熟悉及建置 srsLTE 的實驗環境