

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。

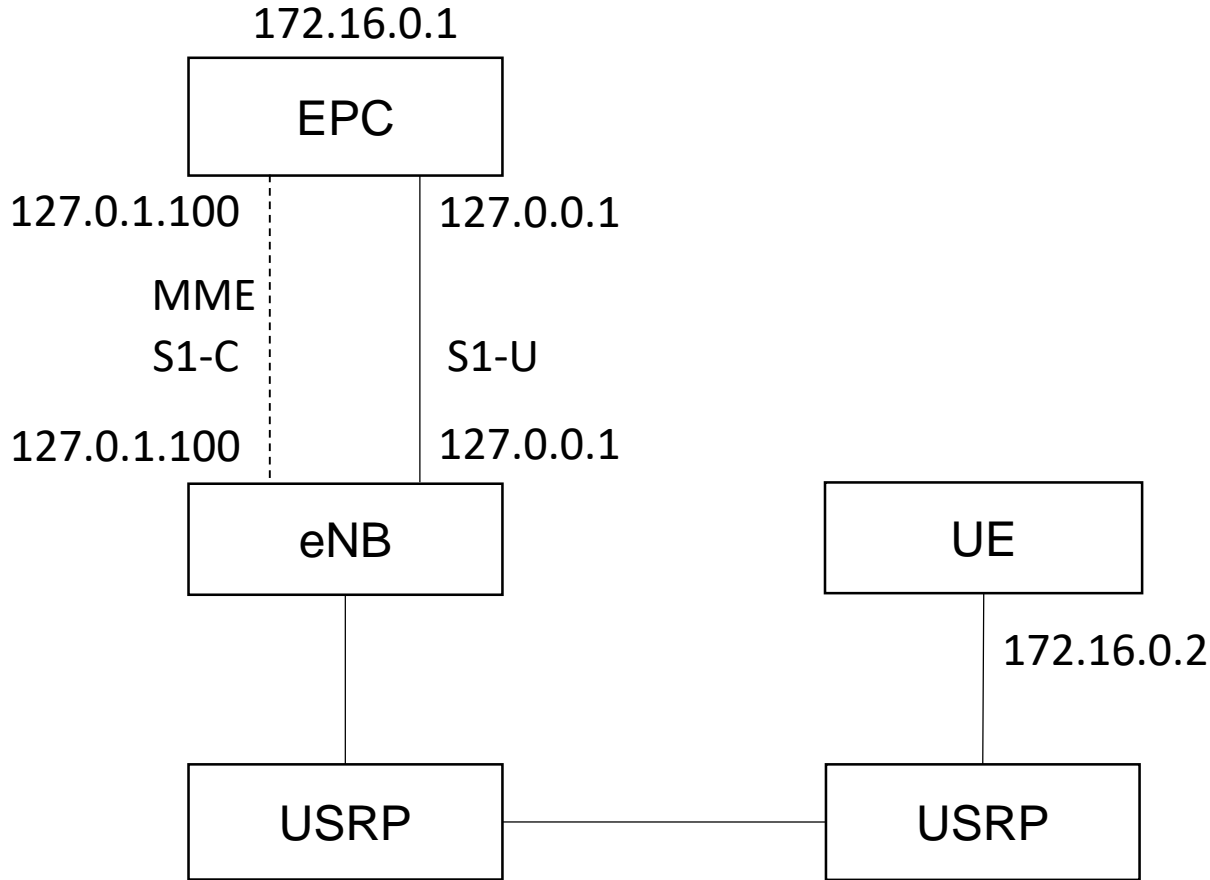
實驗內容

- 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-lowlatency
	srsLTE	srsLTE 19.03 5343b33f8ab2edf7319b6abb07bbc3970541517a
UE	OS : Ubuntu	Ubuntu 16.04
		Kernel : 4.15.0-041500-lowlatency
	srsLTE	srsLTE 19.03 5343b33f8ab2edf7319b6abb07bbc3970541517a

下載及安裝Kernel

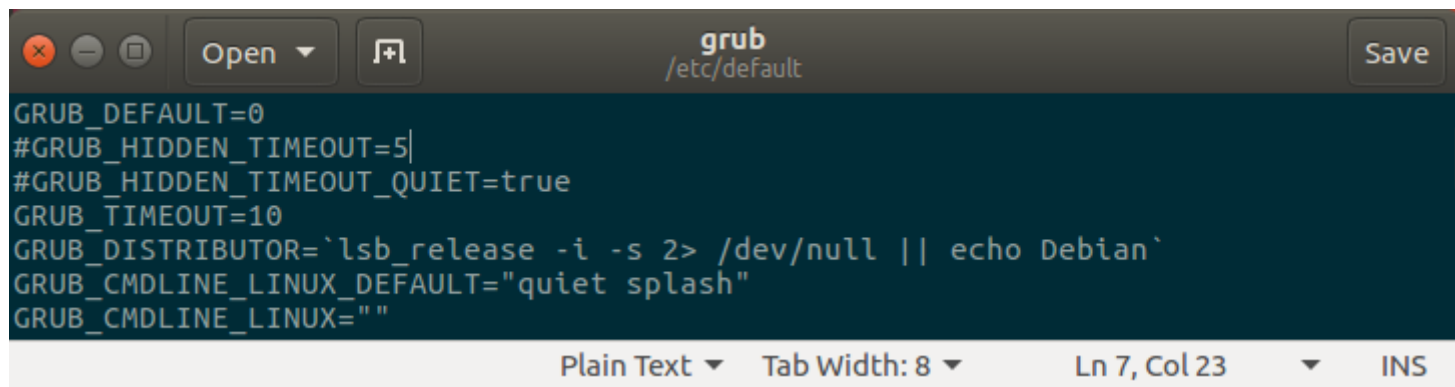
- 開啟一個終端機，並且依序輸入
- `wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-headers-4.4.15-040415-040415.201607111333_all.deb`
- `wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-headers-4.4.15-040415-lowlatency_4.4.15-040415.201607111333_amd64.deb`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



```
grub
/etc/default

GRUB_DEFAULT=0
#GRUB_HIDDEN_TIMEOUT=5
#GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

Plain Text ▾ Tab Width: 8 ▾ Ln 7, Col 23 ▾ INS
```

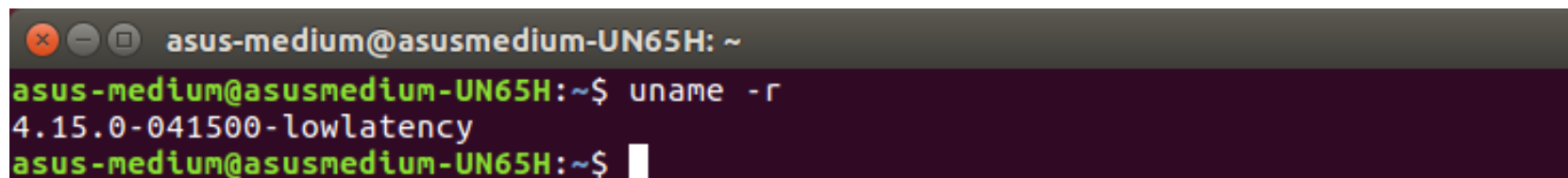
更新grub設定

- 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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ sudo update-grub2  
[sudo] password for asus-medium:  
Generating grub configuration file ...  
Found linux image: /boot/vmlinuz-4.15.0-041500-lowlatency  
Found initrd image: /boot/initrd.img-4.15.0-041500-lowlatency  
Found linux image: /boot/vmlinuz-4.15.0-041500-lowlatency  
Found initrd image: /boot/initrd.img-4.15.0-041500-lowlatency  
Found linux image: /boot/vmlinuz-4.15.0-47-generic  
Found initrd image: /boot/initrd.img-4.15.0-47-generic  
Found linux image: /boot/vmlinuz-4.15.0-46-generic  
Found initrd image: /boot/initrd.img-4.15.0-46-generic  
Found linux image: /boot/vmlinuz-4.15.0-30-lowlatency  
Found initrd image: /boot/initrd.img-4.15.0-30-lowlatency  
Adding boot menu entry for EFI firmware configuration  
done  
asus-medium@asusmedium-UN65H:~$
```

檢查Kernel版本

- 重新開機後在終端機輸入指令，確認版本
- `uname -r`



```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ uname -r  
4.15.0-041500-lowlatency  
asus-medium@asusmedium-UN65H:~$
```

Outline

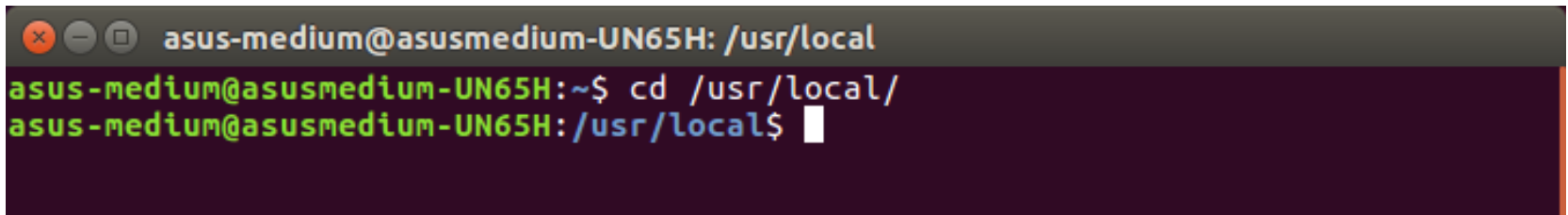
- 實驗目的及實驗內容
- srsLTE 實驗環境
 - srsLTE Small Cell 架構
 - 軟硬體環境
- 基本 Linux 指令
 - 檔案相關指令
 - 網路相關指令
- srsLTE 網路實驗平台建置
 - 安裝所需套件
 - 安裝 srsLTE 網路環境
- 執行程式暨測試
- 總結

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

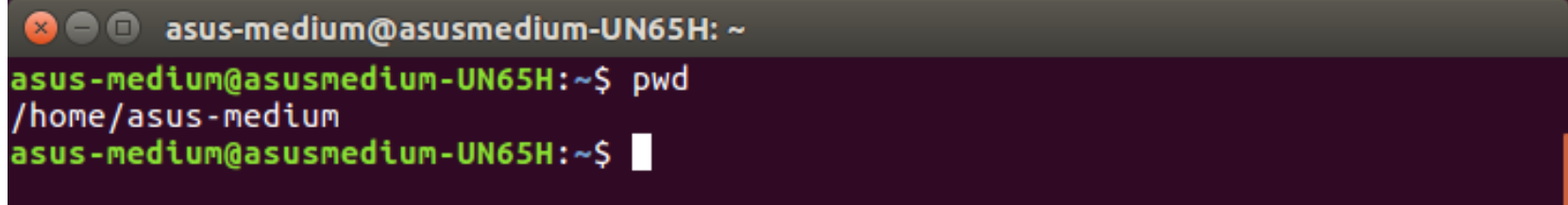
- 更改工作路徑指令
- 相對路徑 vs. 絕對路徑
- `cd ..` #回到上一層
- `cd ~` #在ubuntu 預設是 `/home/user_name/`
- `cd /usr/bin` #移動到指定路徑



```
asus-medium@asusmedium-UN65H: /usr/local
asus-medium@asusmedium-UN65H:~$ cd /usr/local/
asus-medium@asusmedium-UN65H:/usr/local$
```


pwd

- 顯示目前的工作路徑



```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ pwd  
/home/asus-medium  
asus-medium@asusmedium-UN65H:~$
```

A terminal window with a dark purple background. The title bar shows window control buttons and the text 'asus-medium@asusmedium-UN65H: ~'. The terminal content shows the command 'pwd' being executed, resulting in the output '/home/asus-medium'. The prompt 'asus-medium@asusmedium-UN65H:~\$' is shown before and after the command.

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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ ls -al  
total 212  
drwxr-xr-x 27 asus-medium asus-medium 4096 七  4 14:59 .  
drwxr-xr-x  3 root         root         4096 七 19 18:45 ..  
-rw-----  1 asus-medium asus-medium 22657 七  4 16:20 .bash_history  
-rw-r--r--  1 asus-medium asus-medium   220 七 19 18:45 .bash_logout  
-rw-r--r--  1 asus-medium asus-medium  3771 七 19 18:45 .bashrc  
drwx----- 22 asus-medium asus-medium 4096 七  1 15:35 .cache  
drwx-----  3 asus-medium asus-medium 4096 七 23 16:18 .compiz  
drwx----- 25 asus-medium asus-medium 4096 七  1 15:35 .config  
drwx-----  3 root         root         4096 三 14 09:27 .dbus
```

mkdir

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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ ls  
Desktop    Downloads    Music    Pictures    scitools    srsGUI    Understand_project  
Documents  examples.desktop  Mytest  Public    SoapySDR    Templates  Videos  
asus-medium@asusmedium-UN65H:~$ mkdir Mydir  
asus-medium@asusmedium-UN65H:~$ ls  
Desktop    examples.desktop  Mytest    scitools    Templates  
Documents  Music             Pictures  SoapySDR    Understand_project  
Downloads  Mydir             Public    srsGUI      Videos  
asus-medium@asusmedium-UN65H:~$
```

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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ ls  
Desktop      examples.desktop  Mytest    scitools  Templates  
Documents    Music             Pictures  SoapySDR  Understand_project  
Downloads    Mydir            Public    srsGUI    Videos  
asus-medium@asusmedium-UN65H:~$ rm -rf Mydir/  
asus-medium@asusmedium-UN65H:~$ ls  
Desktop      Downloads         Music     Pictures  scitools  srsGUI    Understand_project  
Documents    examples.desktop  Mytest    Public    SoapySDR  Templates  Videos  
asus-medium@asusmedium-UN65H:~$
```

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

```
asus-medium@asusmedium-UN65H: ~/command_dir
asus-medium@asusmedium-UN65H:~$ ls
command_dir  Documents      Music          Public         srsGUI          Videos
command_test Downloads      Mytest         scitools       Templates
Desktop      examples.desktop Pictures        SoapySDR       Understand_project
asus-medium@asusmedium-UN65H:~$ mv command_test ./command_dir/
asus-medium@asusmedium-UN65H:~$ cd command_dir/
asus-medium@asusmedium-UN65H:~/command_dir$ ls
command_test
asus-medium@asusmedium-UN65H:~/command_dir$ mv command_test command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$ ls
command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$
```

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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~/command_dir$ cp command_test_rename ~/  
asus-medium@asusmedium-UN65H:~/command_dir$ cd ../  
asus-medium@asusmedium-UN65H:~$ ls  
command_dir      Documents      Music          Public         srsGUI         Videos  
command_test_rename Downloads      Mytest         scitools       Templates  
Desktop          examples.desktop Pictures        SoapySDR       Understand_project  
asus-medium@asusmedium-UN65H:~$
```

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

```
asus-medium@asusmedium-UN65H: ~/command_dir
asus-medium@asusmedium-UN65H:~/command_dir$ find ./ command_test_rename
./
./command_test_rename
command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$
```

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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ grep "Hello" ./command_dir/command_test_rename  
Hello test for command  
asus-medium@asusmedium-UN65H:~$ ls | grep com  
command_dir  
command_test_rename  
asus-medium@asusmedium-UN65H:~$
```


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			
	a	=(設定)	x	

```
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]`

```
asus-medium@asusmedium-UN65H: ~/command_dir
asus-medium@asusmedium-UN65H:~/command_dir$ sudo chown asus-medium:test1 command_test_rename
[sudo] password for asus-medium:
asus-medium@asusmedium-UN65H:~/command_dir$ ls -l
total 4
-rw-r----- 1 asus-medium test1 23  7  4 16:36 command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$
```

chgrp

- 改變檔案的擁有群組
- `chgrp group [file]`

```
asus-medium@asusmedium-UN65H: ~/command_dir
asus-medium@asusmedium-UN65H:~/command_dir$ ls -l
total 4
-rw-r----- 1 asus-medium test1 23  7  4 16:36 command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$ sudo chgrp asus-medium command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$ ls -l
total 4
-rw-r----- 1 asus-medium asus-medium 23  7  4 16:36 command_test_rename
asus-medium@asusmedium-UN65H:~/command_dir$
```

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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ ifconfig  
dongle    Link encap:Ethernet  HWaddr 98:de:d0:13:9b:1a  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
eth0      Link encap:Ethernet  HWaddr 78:24:af:04:55:03  
          inet addr:192.168.128.101  Bcast:192.168.128.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:10009 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:3767 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:9801668 (9.8 MB)  TX bytes:292349 (292.3 KB)
```

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 (Adap  
tive 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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ netstat -s  
Ip:  
  9778 total packets received  
    2 with invalid addresses  
    0 forwarded  
    0 incoming packets discarded  
  9775 incoming packets delivered  
  4234 requests sent out  
    40 outgoing packets dropped
```


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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ nslookup www.google.com  
Server:          127.0.1.1  
Address:         127.0.1.1#53  
  
Non-authoritative answer:  
Name:   www.google.com  
Address: 172.217.24.4  
  
asus-medium@asusmedium-UN65H:~$
```

traceroute

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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ traceroute www.nuk.edu.tw  
traceroute to www.nuk.edu.tw (140.127.234.77), 30 hops max, 60 byte packets  
 1  192.168.128.1 (192.168.128.1)  0.168 ms  0.202 ms  0.236 ms  
 2  10.1.208.254 (10.1.208.254)  16.947 ms  17.253 ms  17.594 ms  
 3  192.168.1.254 (192.168.1.254)  2.094 ms  2.231 ms  2.258 ms  
 4  192.168.249.253 (192.168.249.253)  0.773 ms  0.768 ms  0.776 ms
```

traceroute

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ traceroute --help  
Usage:  
  traceroute [ -4dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [ -m max_ttl ] [ -N squeries ] [ -p port ] [ -t tos ] [ -l flow_label ] [ -w wait time ] [ -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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ route  
Kernel IP routing table  
Destination      Gateway           Genmask           Flags  Metric  Ref    Use  Iface  
default          192.168.128.1    0.0.0.0           UG      100      0        0  eth0  
link-local       *                255.255.0.0       U        1000     0        0  eth0  
192.168.128.0    *                255.255.255.0     U        100      0        0  eth0  
asus-medium@asusmedium-UN65H:~$
```

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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ ping 192.168.128.1 -c 10  
PING 192.168.128.1 (192.168.128.1) 56(84) bytes of data.  
64 bytes from 192.168.128.1: icmp_seq=1 ttl=64 time=0.253 ms  
64 bytes from 192.168.128.1: icmp_seq=2 ttl=64 time=0.217 ms  
64 bytes from 192.168.128.1: icmp_seq=3 ttl=64 time=0.231 ms  
64 bytes from 192.168.128.1: icmp_seq=4 ttl=64 time=0.224 ms  
64 bytes from 192.168.128.1: icmp_seq=5 ttl=64 time=0.256 ms  
64 bytes from 192.168.128.1: icmp_seq=6 ttl=64 time=0.218 ms  
64 bytes from 192.168.128.1: icmp_seq=7 ttl=64 time=0.226 ms  
64 bytes from 192.168.128.1: icmp_seq=8 ttl=64 time=0.223 ms  
64 bytes from 192.168.128.1: icmp_seq=9 ttl=64 time=0.221 ms  
64 bytes from 192.168.128.1: icmp_seq=10 ttl=64 time=0.235 ms  
  
--- 192.168.128.1 ping statistics ---  
10 packets transmitted, 10 received, 0% packet loss, time 9218ms  
rtt min/avg/max/mdev = 0.217/0.230/0.256/0.018 ms  
asus-medium@asusmedium-UN65H:~$
```

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:~$
```

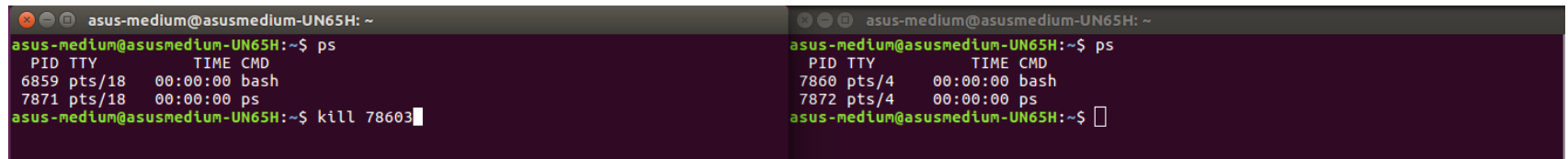

ps

- 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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ ps -l  
F S    UID    PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD  
0 S    1000    7572  6855  0  80   0  -  5823 wait  pts/4        00:00:00 bash  
0 R    1000    7769  7572  0  80   0  -  7379 -      pts/4        00:00:00 ps  
asus-medium@asusmedium-UN65H:~$
```

kill

- 結束執行中的程式的指令
- kill [PID]



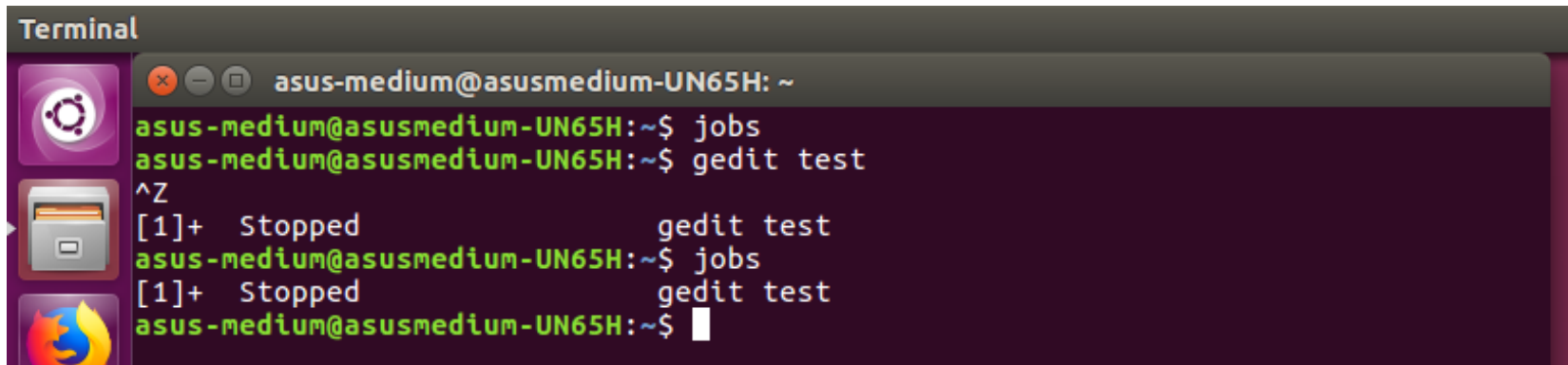
The image shows two terminal windows side-by-side. The left window shows the output of the 'ps' command, listing processes with their PIDs, TTYs, times, and commands. The right window shows the output of the 'ps' command after the 'kill' command has been executed, showing that the process with PID 7860 has been terminated.

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ ps  
  PID TTY          TIME CMD  
 6859 pts/18    00:00:00 bash  
 7871 pts/18    00:00:00 ps  
asus-medium@asusmedium-UN65H:~$ kill 78603
```

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ ps  
  PID TTY          TIME CMD  
 7860 pts/4      00:00:00 bash  
 7872 pts/4      00:00:00 ps  
asus-medium@asusmedium-UN65H:~$
```

jobs

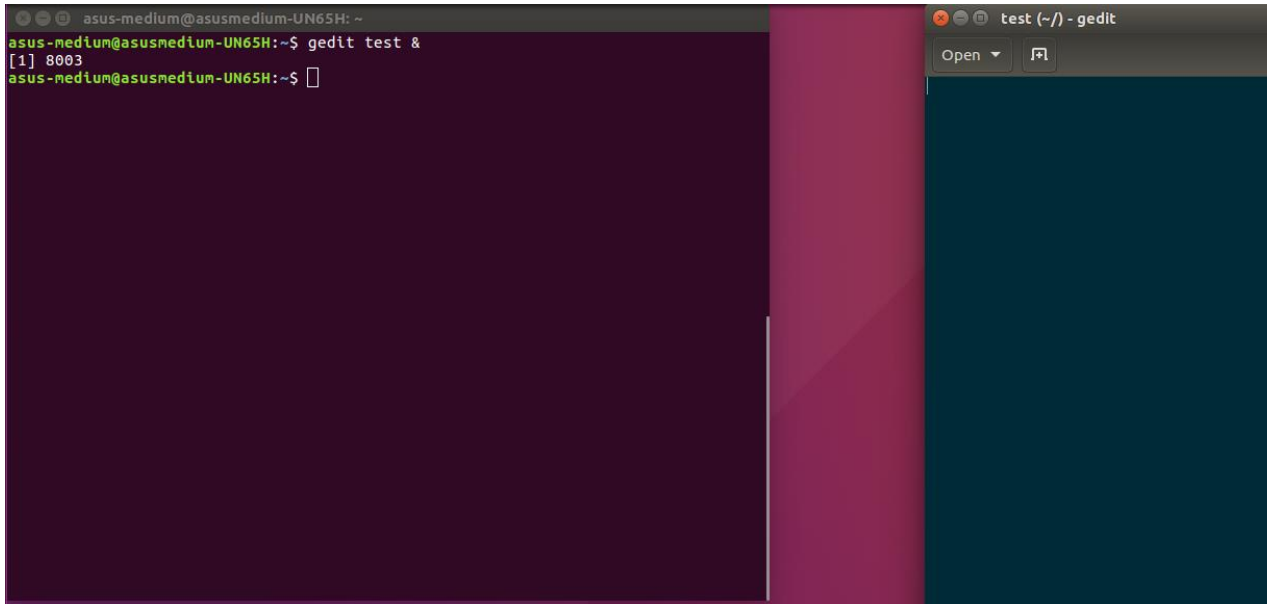
- 查看目前背景有哪些工作
- jobs #顯示工作狀態
- jobs -l #顯示工作狀態外，同時顯示PID



```
Terminal
asus-medium@asusmedium-UN65H: ~
asus-medium@asusmedium-UN65H:~$ jobs
asus-medium@asusmedium-UN65H:~$ gedit test
^Z
[1]+  Stopped                  gedit test
asus-medium@asusmedium-UN65H:~$ jobs
[1]+  Stopped                  gedit test
asus-medium@asusmedium-UN65H:~$
```

&,[Ctrl]+[Z]

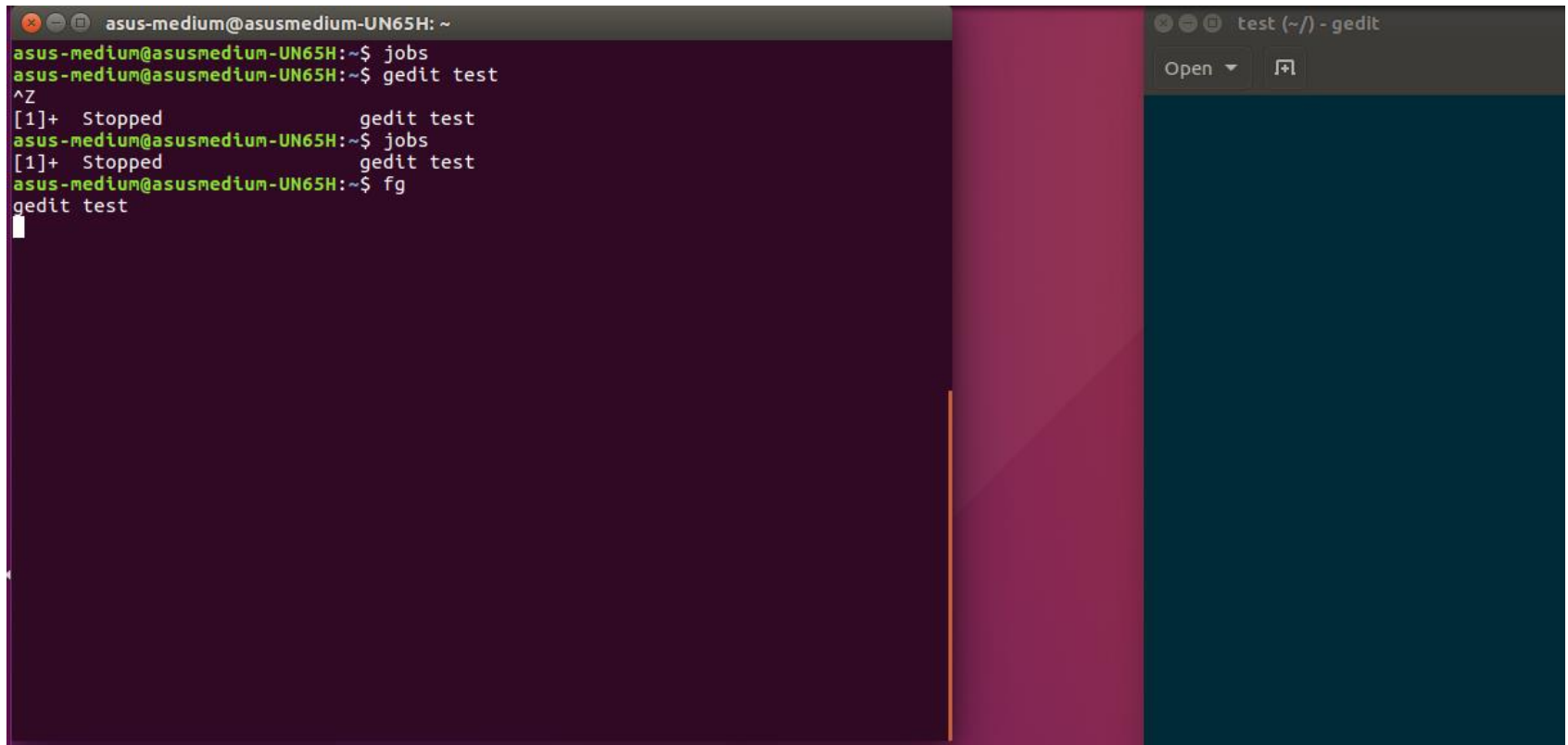
- 在指令後面加上”&”，代表將工作放在背景執行
#工作狀態為”Running”
- 輸入完指令後，按下[Ctrl]+[Z]，代表把工作放置背景暫停
#工作狀態為”Stopped”



```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ gedit test &  
[1] 8003  
asus-medium@asusmedium-UN65H:~$
```

fg

- 將背景工作拿至前景處理

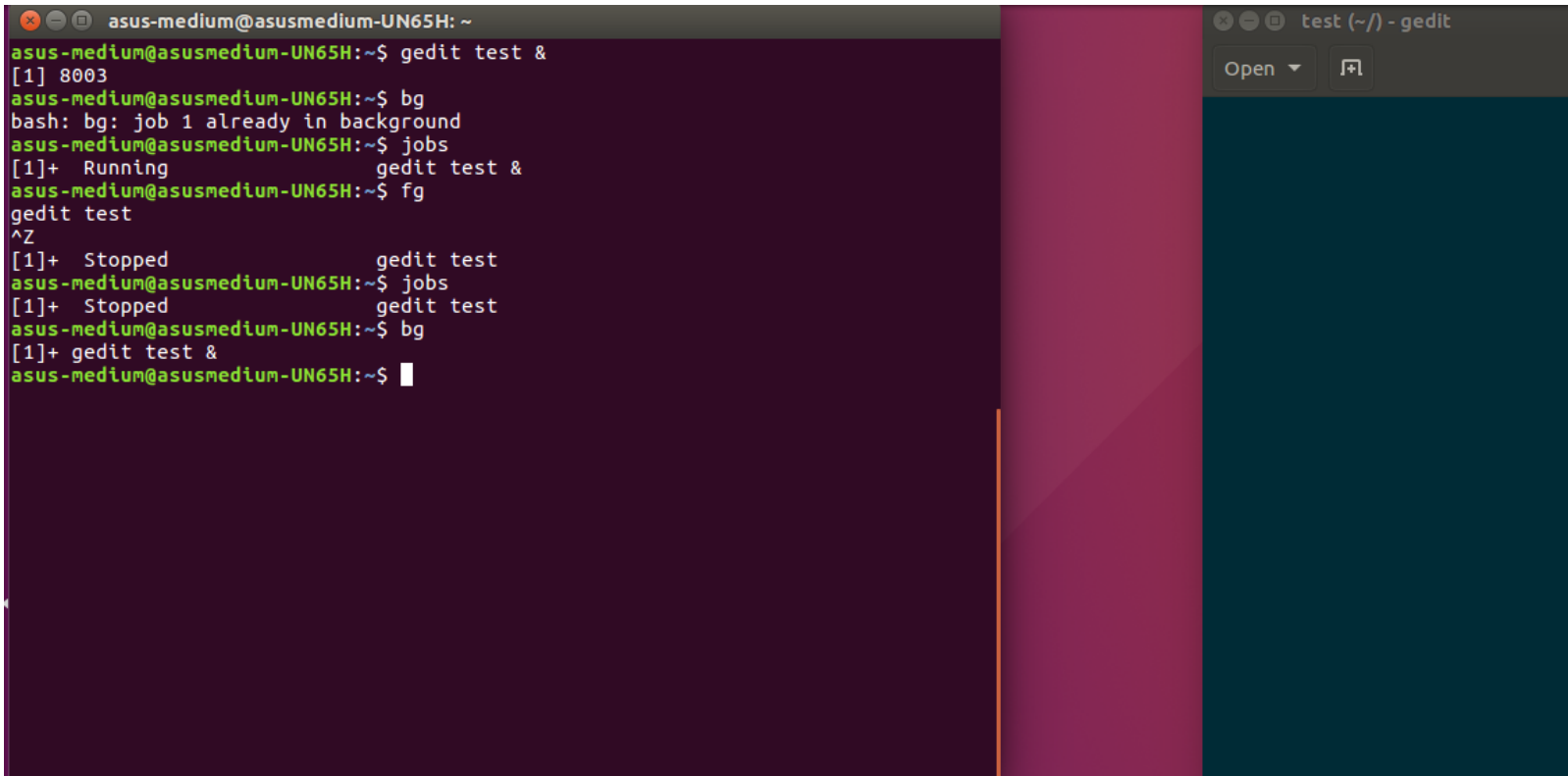


```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ jobs  
asus-medium@asusmedium-UN65H:~$ gedit test  
^Z  
[1]+  Stopped                  gedit test  
asus-medium@asusmedium-UN65H:~$ jobs  
[1]+  Stopped                  gedit test  
asus-medium@asusmedium-UN65H:~$ fg  
gedit test
```

The screenshot shows two windows. The left window is a terminal with a dark purple background. It displays the execution of the 'gedit test' command in the background, which is then paused with Ctrl-Z and brought to the foreground using the 'fg' command. The right window is a gedit editor titled 'test (~/) - gedit', showing a dark blue editor area with a toolbar at the top containing 'Open' and a file icon.

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”&”



```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ gedit test &  
[1] 8003  
asus-medium@asusmedium-UN65H:~$ bg  
bash: bg: job 1 already in background  
asus-medium@asusmedium-UN65H:~$ jobs  
[1]+  Running                  gedit test &  
asus-medium@asusmedium-UN65H:~$ fg  
gedit test  
^Z  
[1]+  Stopped                  gedit test  
asus-medium@asusmedium-UN65H:~$ jobs  
[1]+  Stopped                  gedit test  
asus-medium@asusmedium-UN65H:~$ bg  
[1]+ gedit test &  
asus-medium@asusmedium-UN65H:~$
```

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`

```
ue@ue-X580VD: ~  
ue@ue-X580VD:~$ sudo add-apt-repository ppa:wireshark-dev/stable  
[sudo] password for ue:  
Latest stable Wireshark releases back-ported from Debian package versions.  
  
Back-porting script is available at https://github.com/rbalint/pkg-wireshark-ubuntu-ppa  
  
From Ubuntu 16.04 you also need to enable "universe" repository, see:  
http://askubuntu.com/questions/148638/how-do-i-enable-the-universe-repository  
  
The packaging repository for Debian and Ubuntu is at: https://salsa.debian.org/debian/wireshark  
More info: https://launchpad.net/~wireshark-dev/+archive/ubuntu/stable  
Press [ENTER] to continue or ctrl-c to cancel adding it  
  
gpg: keyring `/tmp/tmp52204kx4/secring.gpg' created  
gpg: keyring `/tmp/tmp52204kx4/pubring.gpg' created  
gpg: requesting key 14ECA0F0 from hkp server keyserver.ubuntu.com  
gpg: /tmp/tmp52204kx4/trustdb.gpg: trustdb created  
gpg: key 14ECA0F0: public key "Launchpad PPA for Wireshark Developers" imported  
gpg: Total number processed: 1  
gpg: imported: 1 (RSA: 1)  
OK  
ue@ue-X580VD:~$
```


- `sudo apt-get update`

```
ue@ue-X580VD: ~  
Get:27 http://tw.archive.ubuntu.com/ubuntu xenial-backports/main amd64 DEP-11 Me  
tadata [3328 B]  
Get:28 http://tw.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 DEP-1  
1 Metadata [5104 B]  
Get:29 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [70  
0 kB]  
Get:30 http://security.ubuntu.com/ubuntu xenial-security/main i386 Packages [572  
kB]  
Get:31 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [27  
9 kB]  
Get:32 http://security.ubuntu.com/ubuntu xenial-security/main amd64 DEP-11 Metad  
ata [73.9 kB]  
Get:33 http://security.ubuntu.com/ubuntu xenial-security/main DEP-11 64x64 Icons  
[73.2 kB]  
Get:34 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 M  
etadata [121 kB]  
Get:35 http://security.ubuntu.com/ubuntu xenial-security/universe DEP-11 64x64 I  
cons [179 kB]  
Get:36 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 DEP-11  
Metadata [2464 B]  
Fetched 7566 kB in 4s (1683 kB/s)  
Reading package lists... Done  
ue@ue-X580VD:~$
```

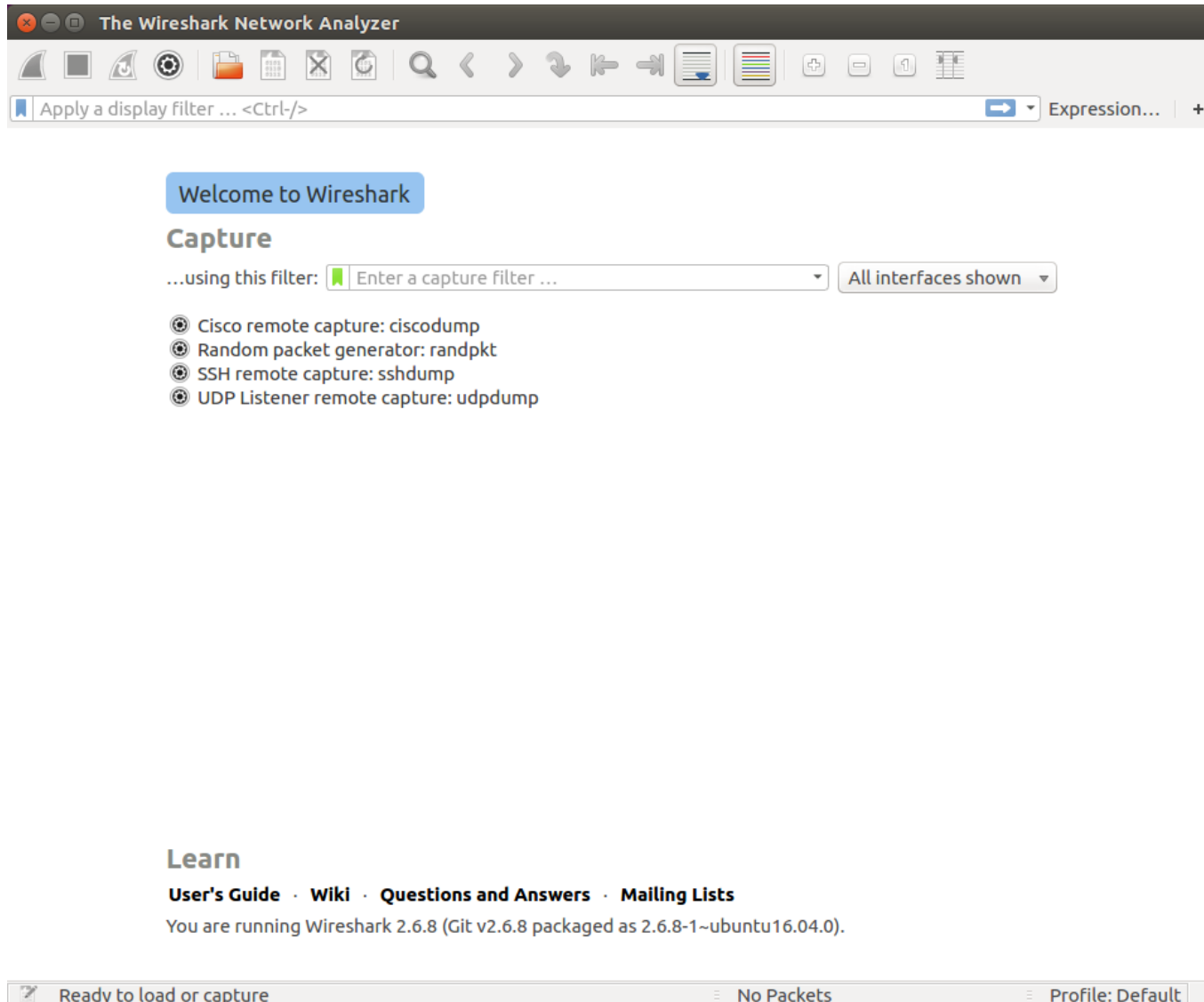
- `sudo apt-get install wireshark`

```
ue@ue-X580VD: ~  
ue@ue-X580VD:~$ sudo apt-get install wireshark  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  libllvm5.0 linux-headers-4.13.0-36 linux-headers-4.13.0-36-generic  
  linux-headers-4.15.0-46 linux-headers-4.15.0-46-generic  
  linux-headers-4.15.0-47 linux-headers-4.15.0-47-generic  
  linux-headers-4.4.0-143 linux-headers-4.4.0-143-generic  
  linux-headers-4.4.0-148 linux-headers-4.4.0-148-generic  
  linux-image-4.13.0-36-generic linux-image-4.15.0-46-generic  
  linux-image-4.15.0-47-generic linux-image-4.4.0-143-generic  
  linux-image-4.4.0-148-generic linux-image-extra-4.13.0-36-generic  
  linux-modules-4.15.0-46-generic linux-modules-4.15.0-47-generic  
  linux-modules-4.4.0-143-generic linux-modules-4.4.0-148-generic  
  linux-modules-extra-4.15.0-47-generic linux-modules-extra-4.4.0-148-generic  
  snapd-login-service  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  wireshark-gtk wireshark-qt  
The following NEW packages will be installed:  
  wireshark wireshark-gtk wireshark-qt  
0 upgraded, 3 newly installed, 0 to remove and 27 not upgraded.  
Need to get 4201 kB of archives.  
After this operation, 10.7 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

- 輸入“y”後按下enter，接著繼續

```
ue@ue-X580VD: ~  
k-gtk amd64 2.6.8-1~ubuntu16.04.0 [680 kB]  
Get:2 http://tw.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 wireshar  
k-qt amd64 2.6.8-1~ubuntu16.04.0 [3517 kB]  
Get:3 http://tw.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 wireshar  
k amd64 2.6.8-1~ubuntu16.04.0 [4426 B]  
Fetched 4201 kB in 0s (4689 kB/s)  
Selecting previously unselected package wireshark-gtk.  
(Reading database ... 438754 files and directories currently installed.)  
Preparing to unpack .../wireshark-gtk_2.6.8-1~ubuntu16.04.0_amd64.deb ...  
Unpacking wireshark-gtk (2.6.8-1~ubuntu16.04.0) ...  
Selecting previously unselected package wireshark-qt.  
Preparing to unpack .../wireshark-qt_2.6.8-1~ubuntu16.04.0_amd64.deb ...  
Unpacking wireshark-qt (2.6.8-1~ubuntu16.04.0) ...  
Selecting previously unselected package wireshark.  
Preparing to unpack .../wireshark_2.6.8-1~ubuntu16.04.0_amd64.deb ...  
Unpacking wireshark (2.6.8-1~ubuntu16.04.0) ...  
Processing triggers for desktop-file-utils (0.22-1ubuntu5.2) ...  
Processing triggers for bamfdaemon (0.5.3~bzip0+16.04.20180209-0ubuntu1) ...  
Rebuilding /usr/share/applications/bamf-2.index...  
Processing triggers for gnome-menus (3.13.3-6ubuntu3.1) ...  
Processing triggers for mime-support (3.59ubuntu1) ...  
Processing triggers for man-db (2.7.5-1) ...  
Setting up wireshark-gtk (2.6.8-1~ubuntu16.04.0) ...  
Setting up wireshark-qt (2.6.8-1~ubuntu16.04.0) ...  
Setting up wireshark (2.6.8-1~ubuntu16.04.0) ...  
ue@ue-X580VD:~$
```

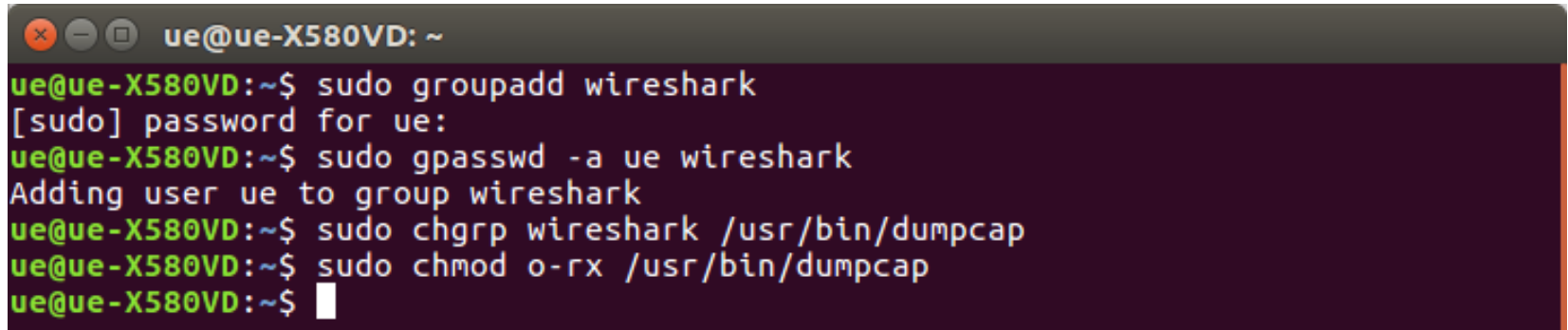
- 安裝完成後，重新登入，並開啟wireshark，若是無法正確顯示網路介面卡資訊，請按照以下步驟解決



- cd 到 /usr/bin
- 查看 dumpcap 檔案的屬性

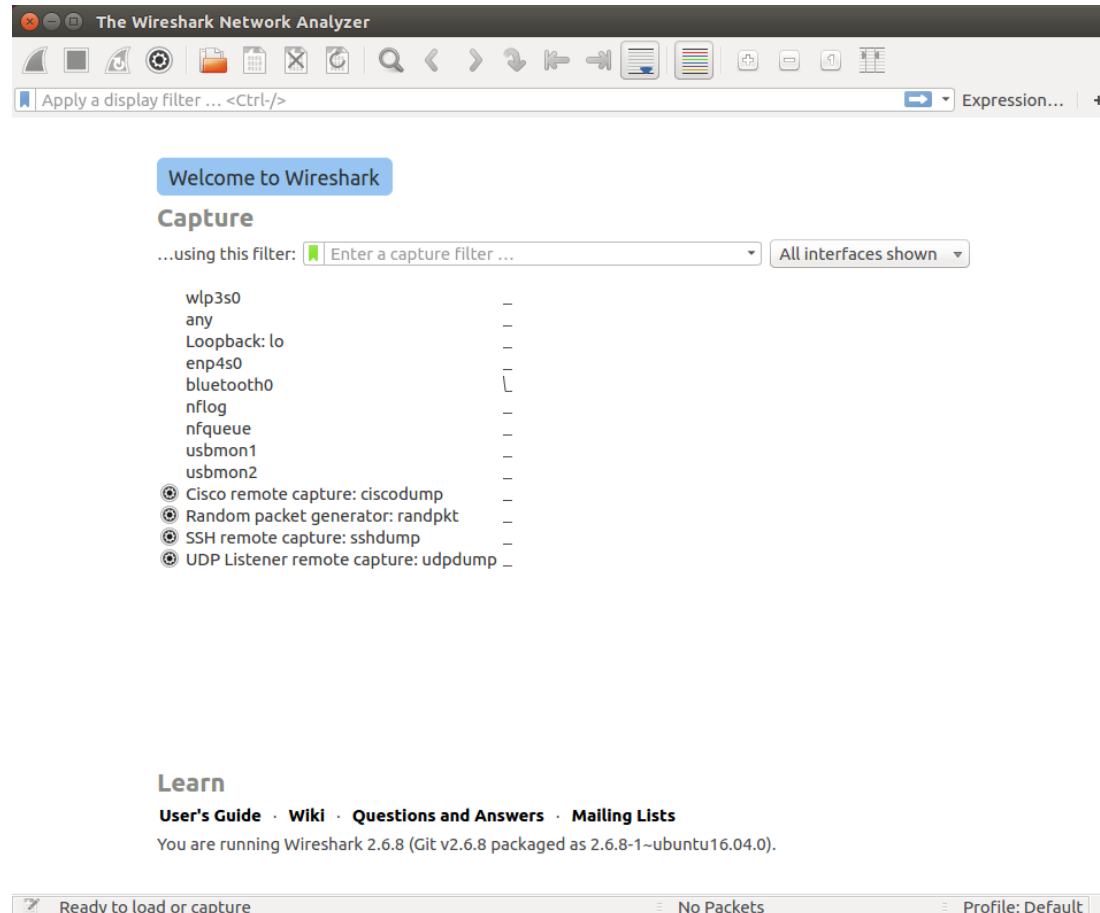
```
ue@ue-X580VD: /usr/bin
ue@ue-X580VD:~$ cd /usr/bin/
ue@ue-X580VD:/usr/bin$ ls -al | grep dumpcap
-rwxr-xr-- 1 root 130 104688 五 16 04:11 dumpcap
ue@ue-X580VD:/usr/bin$
```

- `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



```
ue@ue-X580VD: ~  
ue@ue-X580VD:~$ sudo groupadd wireshark  
[sudo] password for ue:  
ue@ue-X580VD:~$ sudo gpasswd -a ue wireshark  
Adding user ue to group wireshark  
ue@ue-X580VD:~$ sudo chgrp wireshark /usr/bin/dumpcap  
ue@ue-X580VD:~$ sudo chmod o-rx /usr/bin/dumpcap  
ue@ue-X580VD:~$
```

- 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`

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>

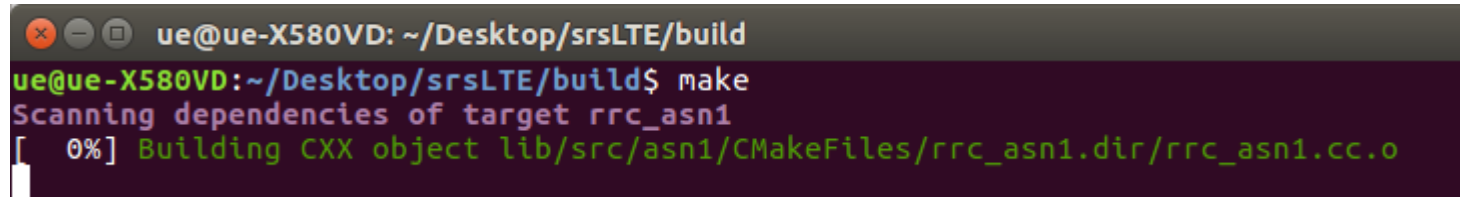
編譯 srsLTE - 1

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

```
ue@ue-X580VD: ~/Desktop/srsLTE/build
ue@ue-X580VD:~$ cd ~/Desktop/srsLTE/
ue@ue-X580VD:~/Desktop/srsLTE$ mkdir build
ue@ue-X580VD:~/Desktop/srsLTE$ cd build/
ue@ue-X580VD:~/Desktop/srsLTE/build$ cmake ../
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
```


編譯 srsLTE - 2

- make



```
ue@ue-X580VD: ~/Desktop/srsLTE/build
ue@ue-X580VD:~/Desktop/srsLTE/build$ make
Scanning dependencies of target rrc_asn1
[ 0%] Building CXX object lib/src/asn1/CMakeFiles/rrc_asn1.dir/rrc_asn1.cc.o
```

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
```

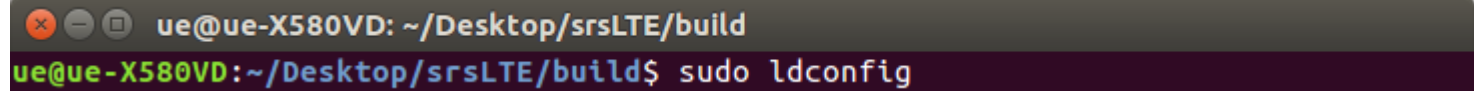
編譯 srsLTE - 4

- sudo make install

```
ue@ue-X580VD: ~/Desktop/srsLTE/build
ue@ue-X580VD:~/Desktop/srsLTE/build$ sudo make install
[sudo] password for ue:
[ 1%] Built target rrc_asn1
[ 2%] Built target srslte_asn1
-- Generating build_info.h
[ 2%] Built target gen_build_info
```

編譯 srsLTE - 5

- sudo ldconfig



```
ue@ue-X580VD: ~/Desktop/srsLTE/build  
ue@ue-X580VD:~/Desktop/srsLTE/build$ 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

```
asus-medium@asusmedium-UN65H: ~/Desktop/srsLTE/srsepc
asus-medium@asusmedium-UN65H:~/Desktop/srsLTE/srsepc$ sudo srsepc
Built in Release mode using commit 5343b33 on branch master.
--- Software Radio Systems EPC ---
Reading configuration file /home/asus-medium/.config/srslte/epc.conf...
HSS Initialized.
MME S11 Initialized
MME GTP-C Initialized
MME Initialized. MCC: 0xf001, MNC: 0xff01
SPGW GTP-U Initialized.
SPGW S11 Initialized.
SP-GW Initialized.
```

顯示目前使用的版本

執行程式-eNB

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

```
asus-medium@asusmedium-UN65H: ~/Desktop/srsLTE/srsenb
asus-medium@asusmedium-UN65H:~/Desktop/srsLTE/srsenb$ sudo srsenb

Built in Release mode using commit 5343b33 on branch master.

--- Software Radio Systems LTE eNodeB ---

Reading configuration file /home/asus-medium/.config/srslte/enb.conf...
[INFO] [UHD] linux; GNU C++ version 5.4.0 20160609; Boost_105800; UHD_3.14.0.0-release
[INFO] [LOGGING] Fastpath logging disabled at runtime.
Opening USRP with args: type=b200, master_clock_rate=30.72e6
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 30.720000 MHz...
[INFO] [B200] Actually got clock rate 30.720000 MHz.
Setting frequency: DL=2685.0 Mhz, UL=2565.0 MHz
[INFO] [B200] Asking for clock rate 11.520000 MHz...
[INFO] [B200] Actually got clock rate 11.520000 MHz.
Setting Sampling frequency 11.52 MHz

==== eNodeB started ====
Type <t> to view trace
```


執行程式

- 在啟動eNB後，EPC會接著顯示S1連線的相關資訊

```
asus-medium@asusmedium-UN65H: ~/Desktop/srsLTE/srsepc
--- exiting ---
asus-medium@asusmedium-UN65H:~/Desktop/srsLTE/srsepc$ clear

asus-medium@asusmedium-UN65H:~/Desktop/srsLTE/srsepc$ sudo srsepc

Built in Release mode using commit 5343b33 on branch master.

--- Software Radio Systems EPC ---

Reading configuration file /home/asus-medium/.config/srslte/epc.conf...
HSS Initialized.
MME S11 Initialized
MME GTP-C Initialized
MME Initialized. MCC: 0xf001, MNC: 0xff01
SPGW GTP-U Initialized.
SPGW S11 Initialized.
SP-GW Initialized.
Received S1 Setup Request.
S1 Setup Request - eNB Name: srsenb01, eNB id: 0x19b
S1 Setup Request - MCC:001, MNC:01, PLMN: 61712
S1 Setup Request - TAC 7, B-PLMN 0
S1 Setup Request - Paging DRX 2
Sending S1 Setup Response
█
```

執行程式-UE

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

```
ue@ue-X580VD: ~/Desktop/srsLTE/srsue
ue@ue-X580VD:~/Desktop/srsLTE/srsue$ sudo srsue
Reading configuration file /home/ue/.config/srslte/ue.conf...

Built in Release mode using commit 5343b33 on branch master.

--- Software Radio Systems LTE UE ---

Opening 1 RF devices with 1 RF channels...
[INFO] [UHD] linux; GNU C++ version 5.4.0 20160609; Boost_105800; UHD_3.14.0.0-release
[INFO] [LOGGING] Fastpath logging disabled at runtime.
Opening USRP with args: type=b200, master_clock_rate=30.72e6
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 30.720000 MHz...
[INFO] [B200] Actually got clock rate 30.720000 MHz.
Waiting PHY to initialize...
...
Attaching UE...
Searching cell in DL EARFCN=3400, f_dl=2685.0 MHz, f_ul=2565.0 MHz
.
Found Cell: Mode=FDD, PCI=1, PRB=50, Ports=1, CFO=0.7 KHz
[INFO] [B200] Asking for clock rate 11.520000 MHz...
[INFO] [B200] Actually got clock rate 11.520000 MHz.
Found PLMN: Id=00101, TAC=7
Random Access Transmission: seq=8, ra-rnti=0x2
RRC Connected
Random Access Complete. c-rnti=0x46, ta=1
Network attach successful. IP: 172.16.0.2
Software Radio Systems LTE (srsLTE)
```

顯示由epc分配給enb
的ip_addr

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: ~  
asus-medium@asusmedium-UN65H:~$ sudo netstat -alpWn | 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/srsnec  
unix 2      [ ]          DGRAM          1700867  5267/srsnec      @mme_s11@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
unix 2      [ ]          DGRAM          1700873  5267/srsnec      @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

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 78:24:af:04:55:03  
          inet addr:192.168.128.101  Bcast:192.168.128.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1  
          RX packets:31202 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:15571 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:22414352 (22.4 MB)  TX bytes:4384829 (4.3 MB)  
  
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:2920 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:2920 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:377570 (377.5 KB)  TX bytes:377570 (377.5 KB)  
  
srs_spgw_sgi Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  
          inet addr:172.16.0.1  P-t-P:172.16.0.1  Mask:255.255.255.0  
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1  
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:500  
          RX bytes:1680 (1.6 KB)  TX bytes:1936 (1.9 KB)  
  
asus-medium@asusmedium-UN65H:~$
```

- 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 [v] [i] 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 slap dissector and with DLT 150.
# To use the dissector, edit the preferences for DLT_USER to
# add an entry with DLT=150, Payload Protocol=slap.
#
# enable: Enable or disable the PCAP.
# filename: File name where to save the PCAP.
#
#####
[pcap]
enable = false
filename = /tmp/epc.pcap

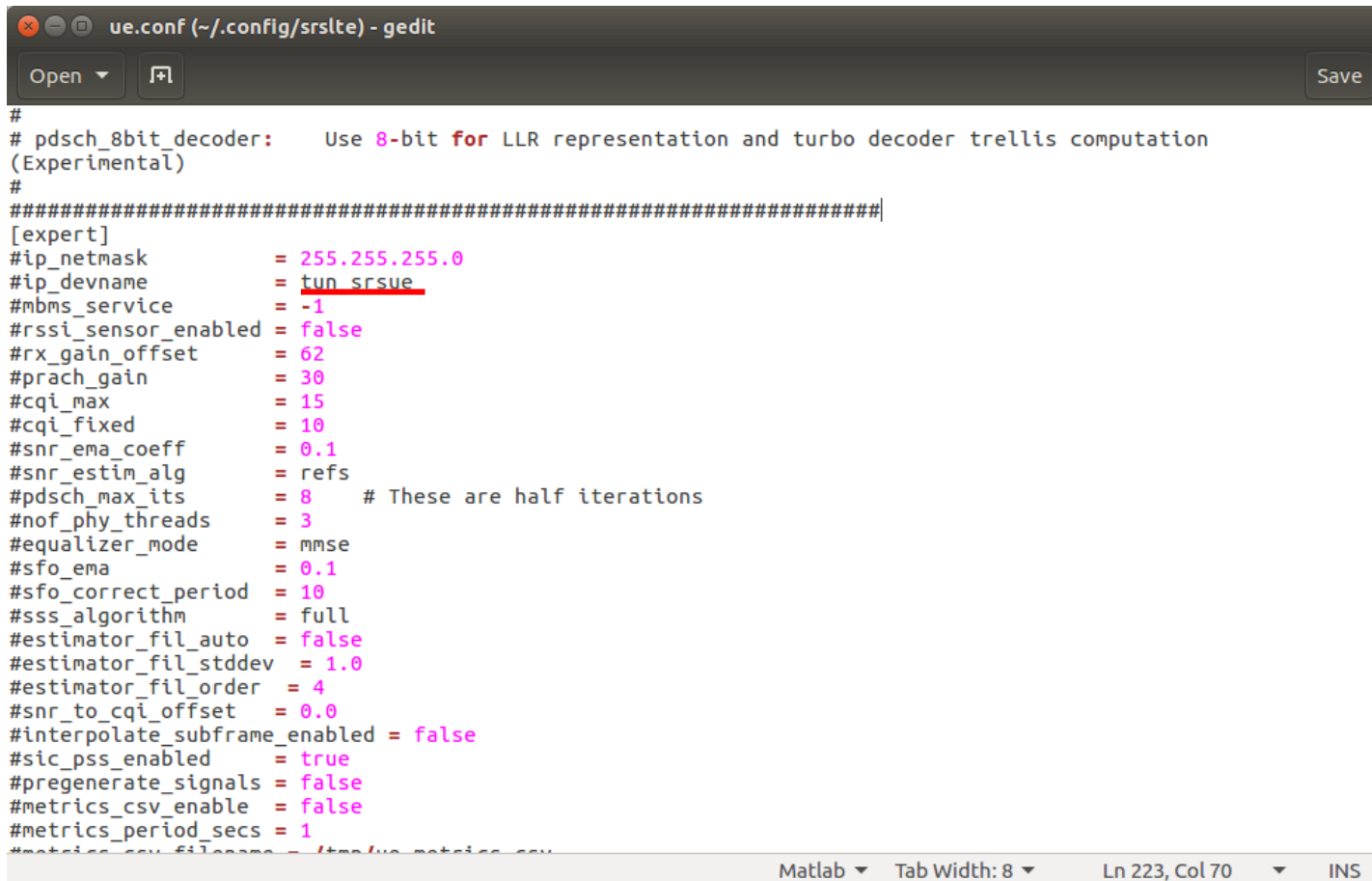
Matlab Tab Width: 8 Ln 1, Col 1 INS
```

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



```
#
# 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
#metrics_csv_filename    = /tmp/ue-metrics.csv
```


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

```
ue@ue-X580VD: ~/Desktop/srsLTE/srsue
ue@ue-X580VD:~/Desktop/srsLTE/srsue$ sudo srsue
[sudo] password for ue:
Reading configuration file /home/ue/.config/srslte/ue.conf...

Built in Release mode using commit 5343b33 on branch master.

--- Software Radio Systems LTE UE ---

Opening 1 RF devices with 1 RF channels...
[INFO] [UHD] linux; GNU C++ version 5.4.0 20160609; Boost_105800; UHD_3.14.0.0-release
[INFO] [LOGGING] Fastpath logging disabled at runtime.
Opening USRP with args:
Error opening UHD: code 11
No Soapy devices found.
/home/ue/Desktop/srsLTE/lib/src/phy/rf/rf_imp.c.126: No compatible RF frontend found

/home/ue/Desktop/srsLTE/lib/src/radio/radio.cc.38: Error opening RF device

Failed to find device auto with args auto
ue@ue-X580VD:~/Desktop/srsLTE/srsue$
```

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`

```
asus-medium@asusmedium-UN65H: ~  
asus-medium@asusmedium-UN65H:~$ /usr/lib/uhd/utils/b2xx_fx3_utils -D  
Device opened (VID=0x2500,PID=0x0020)  
B2xx detected... Control of B2xx granted...  
  
Operation complete! I did it! I did it!  
asus-medium@asusmedium-UN65H:~$
```

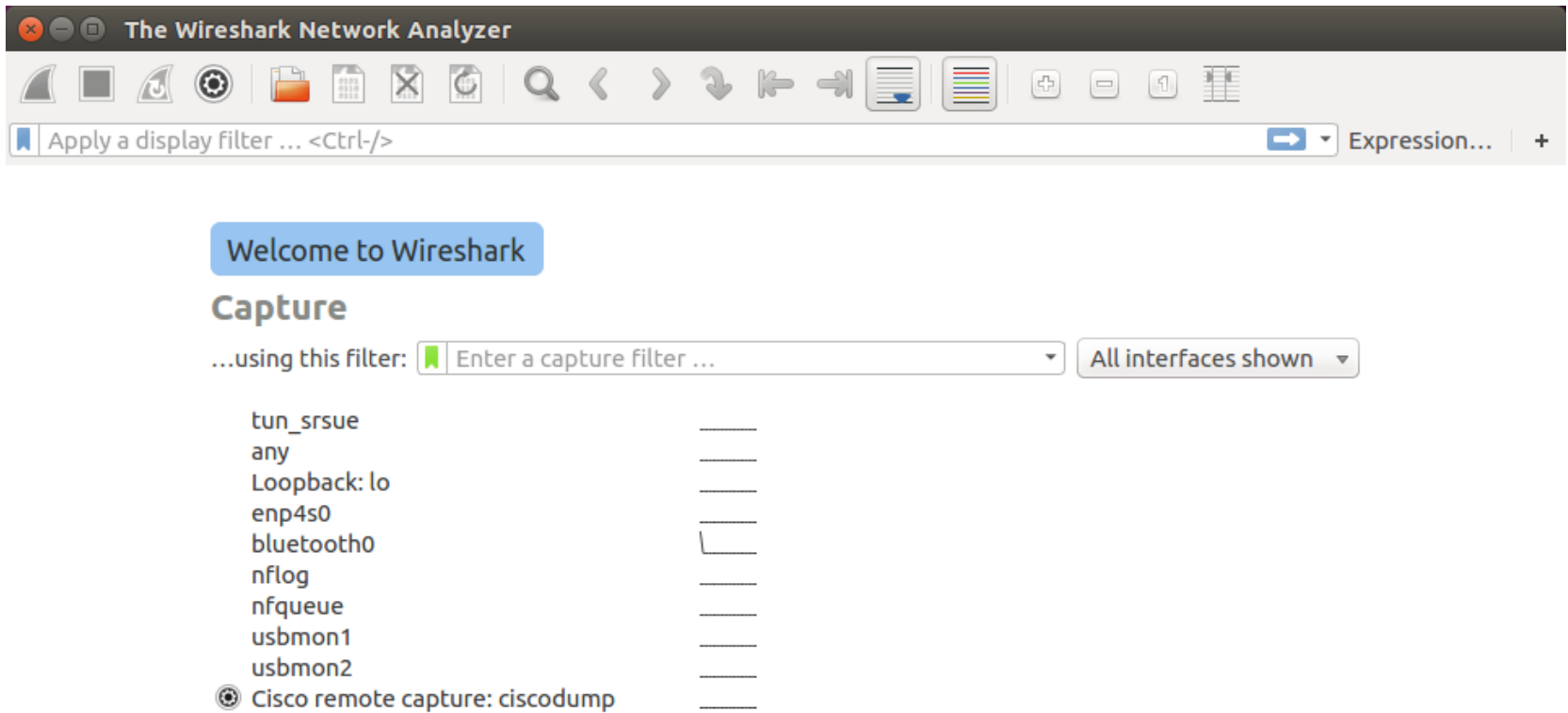
互通測試

- 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

```
ue@ue-X580VD: ~  
ue@ue-X580VD:~$ ping 172.16.0.1 -c 10  
PING 172.16.0.1 (172.16.0.1) 56(84) bytes of data.  
64 bytes from 172.16.0.1: icmp_seq=1 ttl=64 time=179 ms  
64 bytes from 172.16.0.1: icmp_seq=2 ttl=64 time=16.0 ms  
64 bytes from 172.16.0.1: icmp_seq=3 ttl=64 time=14.0 ms  
64 bytes from 172.16.0.1: icmp_seq=4 ttl=64 time=11.8 ms  
64 bytes from 172.16.0.1: icmp_seq=5 ttl=64 time=18.0 ms  
64 bytes from 172.16.0.1: icmp_seq=6 ttl=64 time=15.8 ms  
64 bytes from 172.16.0.1: icmp_seq=7 ttl=64 time=12.8 ms  
64 bytes from 172.16.0.1: icmp_seq=8 ttl=64 time=9.98 ms  
64 bytes from 172.16.0.1: icmp_seq=9 ttl=64 time=16.9 ms  
64 bytes from 172.16.0.1: icmp_seq=10 ttl=64 time=13.8 ms  
  
--- 172.16.0.1 ping statistics ---  
10 packets transmitted, 10 received, 0% packet loss, time 9012ms  
rtt min/avg/max/mdev = 9.983/30.951/179.973/49.728 ms  
ue@ue-X580VD:~$
```

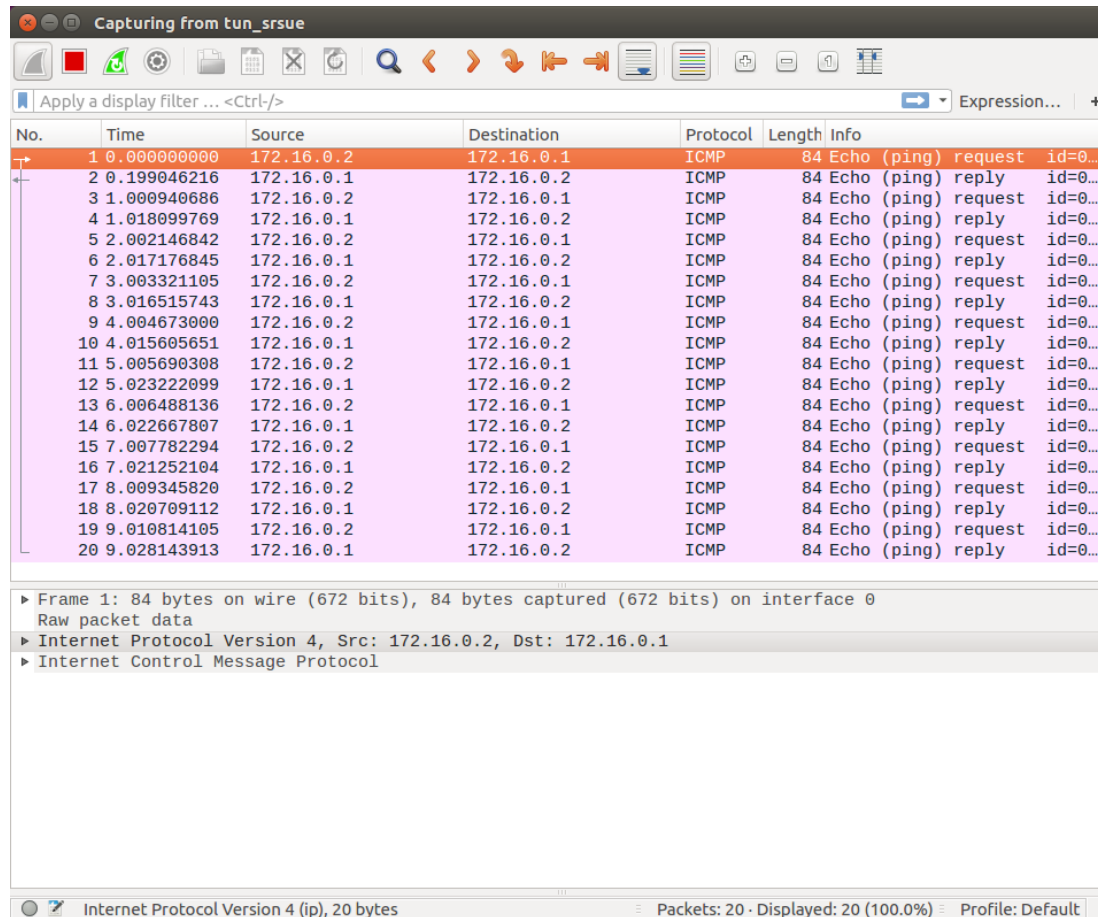
Wireshark 介面選擇

- 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



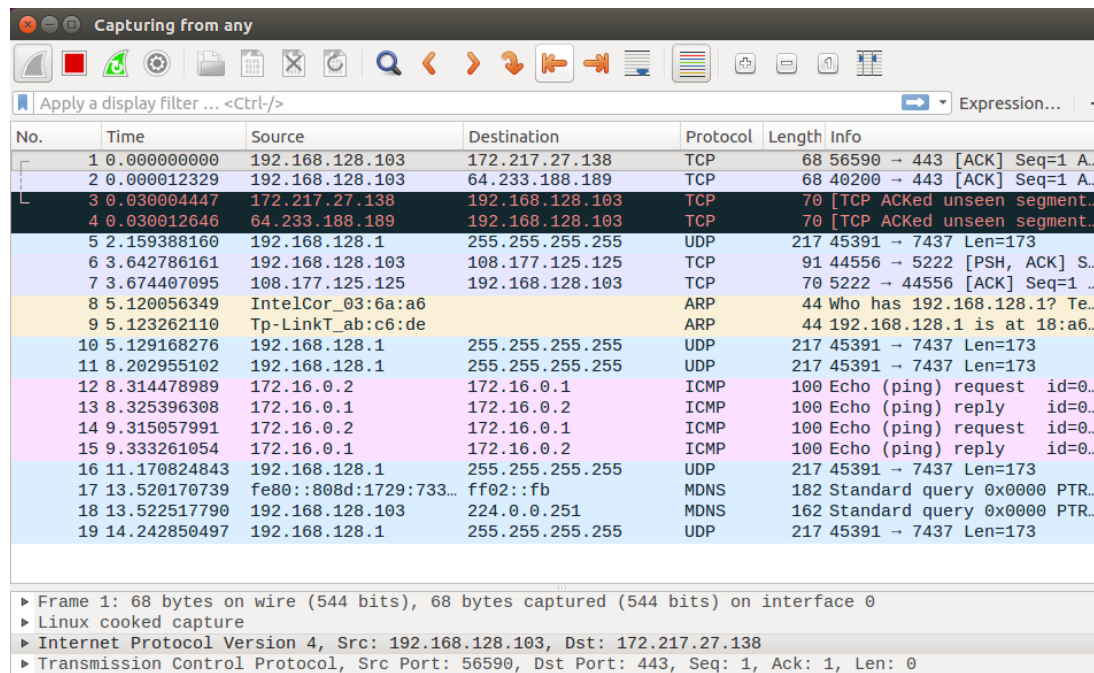
Wireshark-srs 介面

- 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

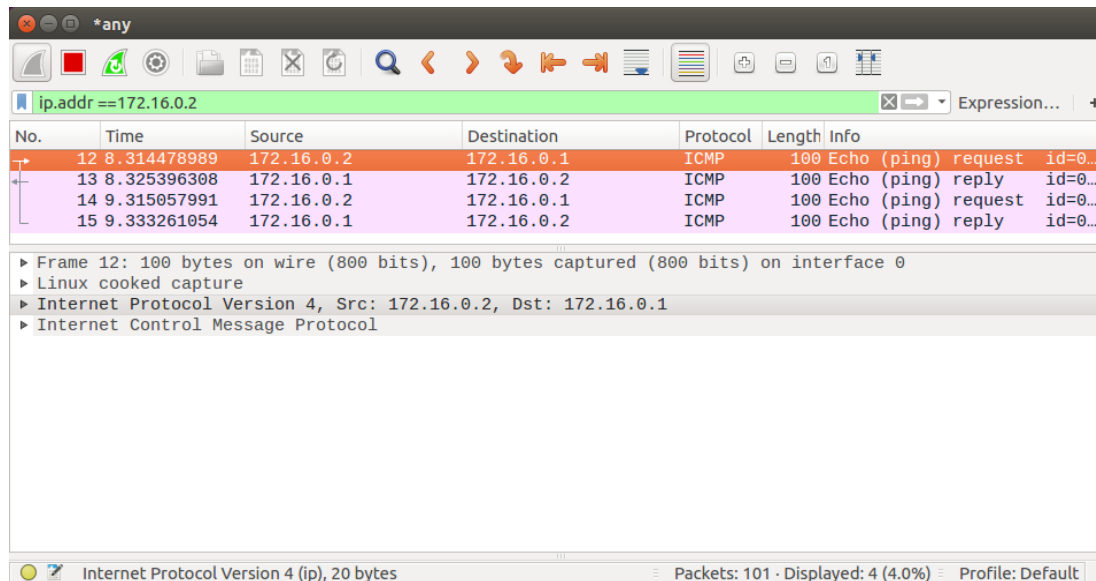


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.128.103	172.217.27.138	TCP	68	56590 → 443 [ACK] Seq=1 A...
2	0.000012329	192.168.128.103	64.233.188.189	TCP	68	40200 → 443 [ACK] Seq=1 A...
3	0.030004447	172.217.27.138	192.168.128.103	TCP	70	[TCP ACKed unseen segment...
4	0.030012646	64.233.188.189	192.168.128.103	TCP	70	[TCP ACKed unseen segment...
5	2.159388160	192.168.128.1	255.255.255.255	UDP	217	45391 → 7437 Len=173
6	3.642786161	192.168.128.103	108.177.125.125	TCP	91	44556 → 5222 [PSH, ACK] S...
7	3.674407095	108.177.125.125	192.168.128.103	TCP	70	5222 → 44556 [ACK] Seq=1 ...
8	5.120056349	IntelCor_03:6a:a6		ARP	44	Who has 192.168.128.1? Te...
9	5.123262110	Tp-LinkT_ab:c6:de		ARP	44	192.168.128.1 is at 18:a6...
10	5.129168276	192.168.128.1	255.255.255.255	UDP	217	45391 → 7437 Len=173
11	8.202955102	192.168.128.1	255.255.255.255	UDP	217	45391 → 7437 Len=173
12	8.314478989	172.16.0.2	172.16.0.1	ICMP	100	Echo (ping) request id=0...
13	8.325396308	172.16.0.1	172.16.0.2	ICMP	100	Echo (ping) reply id=0...
14	9.315057991	172.16.0.2	172.16.0.1	ICMP	100	Echo (ping) request id=0...
15	9.333261054	172.16.0.1	172.16.0.2	ICMP	100	Echo (ping) reply id=0...
16	11.170824843	192.168.128.1	255.255.255.255	UDP	217	45391 → 7437 Len=173
17	13.520170739	fe80::808d:1729:733...	ff02::fb	MDNS	182	Standard query 0x0000 PTR...
18	13.522517790	192.168.128.103	224.0.0.251	MDNS	162	Standard query 0x0000 PTR...
19	14.242850497	192.168.128.1	255.255.255.255	UDP	217	45391 → 7437 Len=173

▶ Frame 1: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 192.168.128.103, Dst: 172.217.27.138
▶ Transmission Control Protocol, Src Port: 56590, Dst Port: 443, Seq: 1, Ack: 1, Len: 0

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