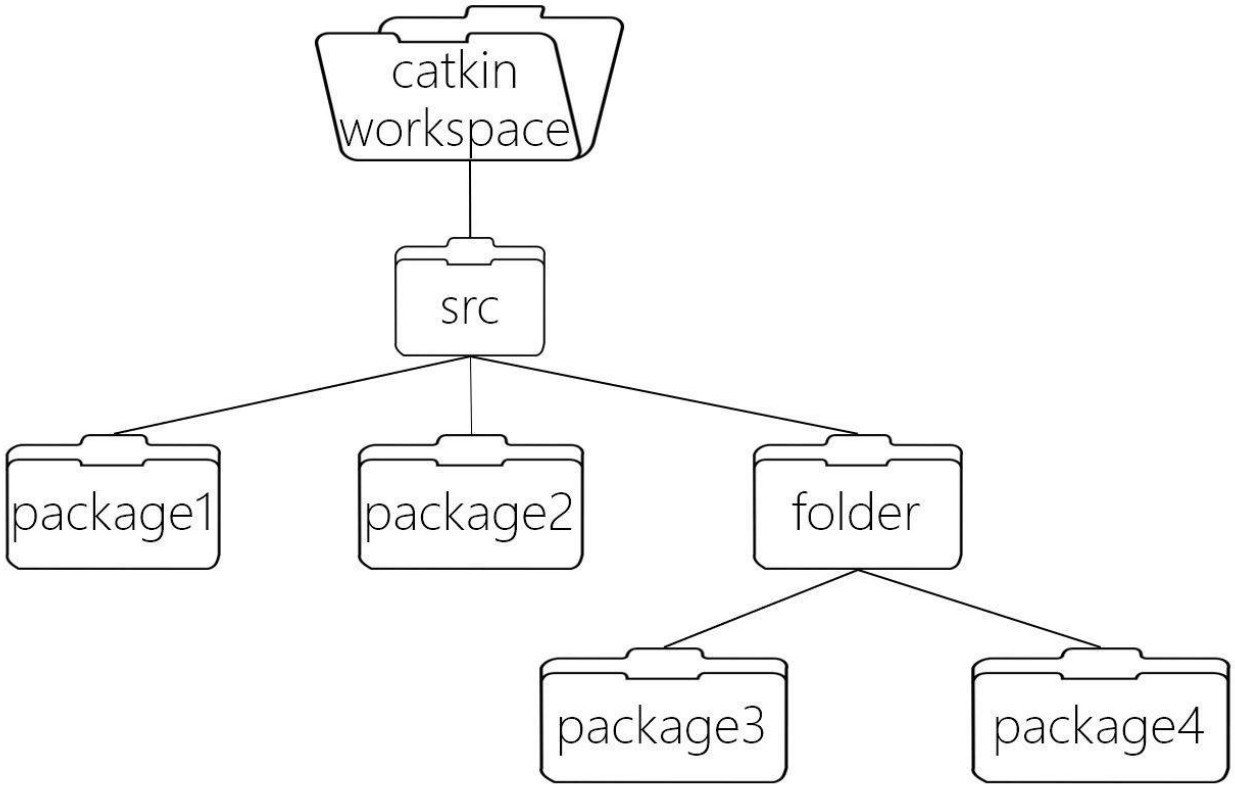
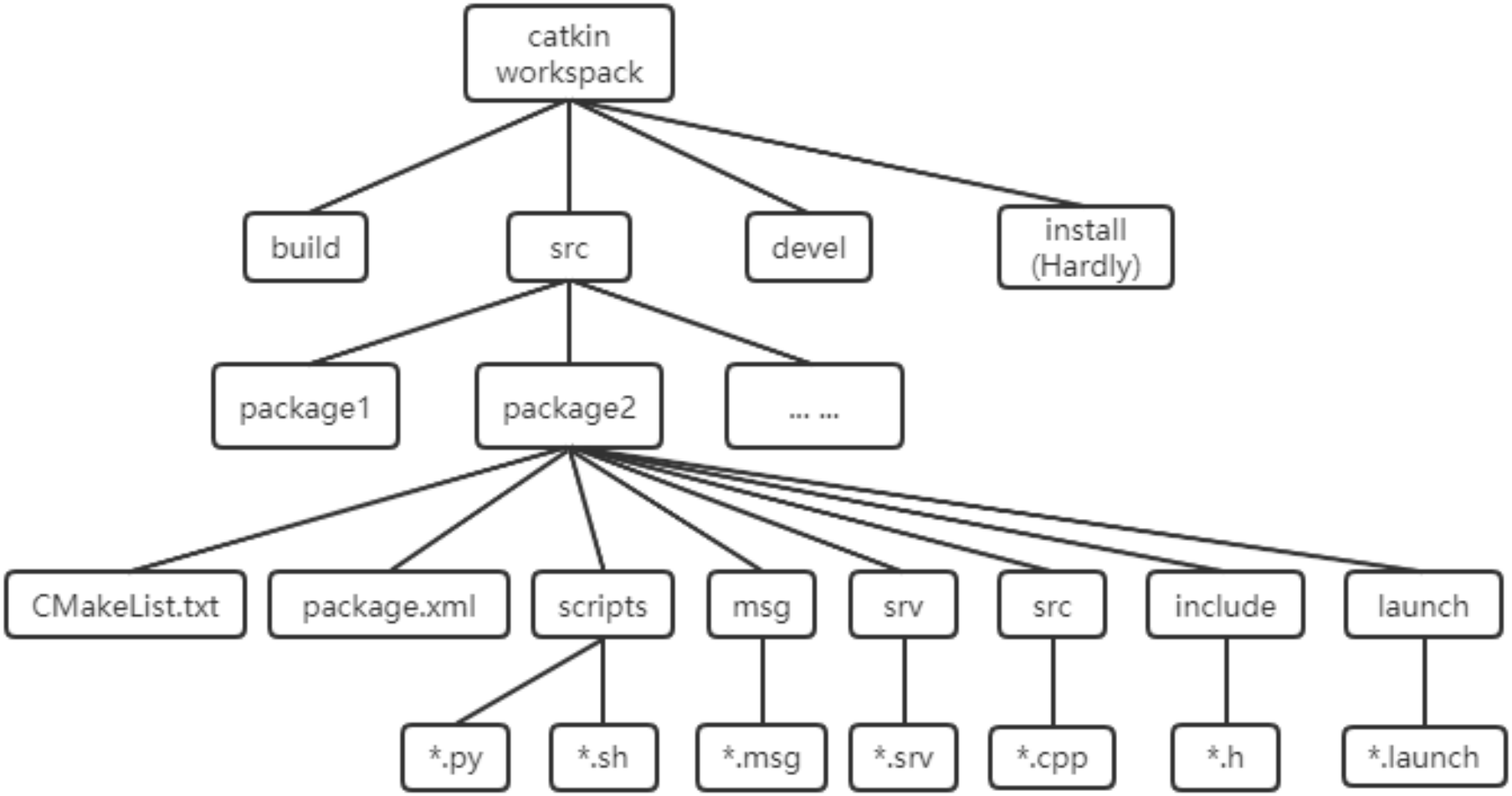
無人載具技術與應用
ROS

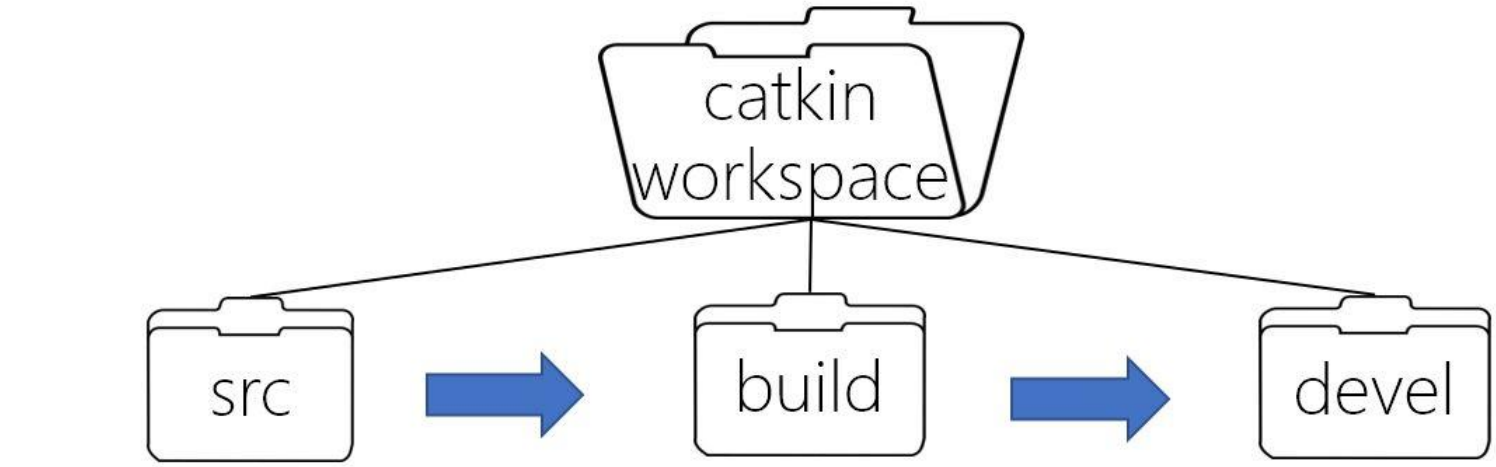
徐瑋隆

wlhsu304@gmail.com

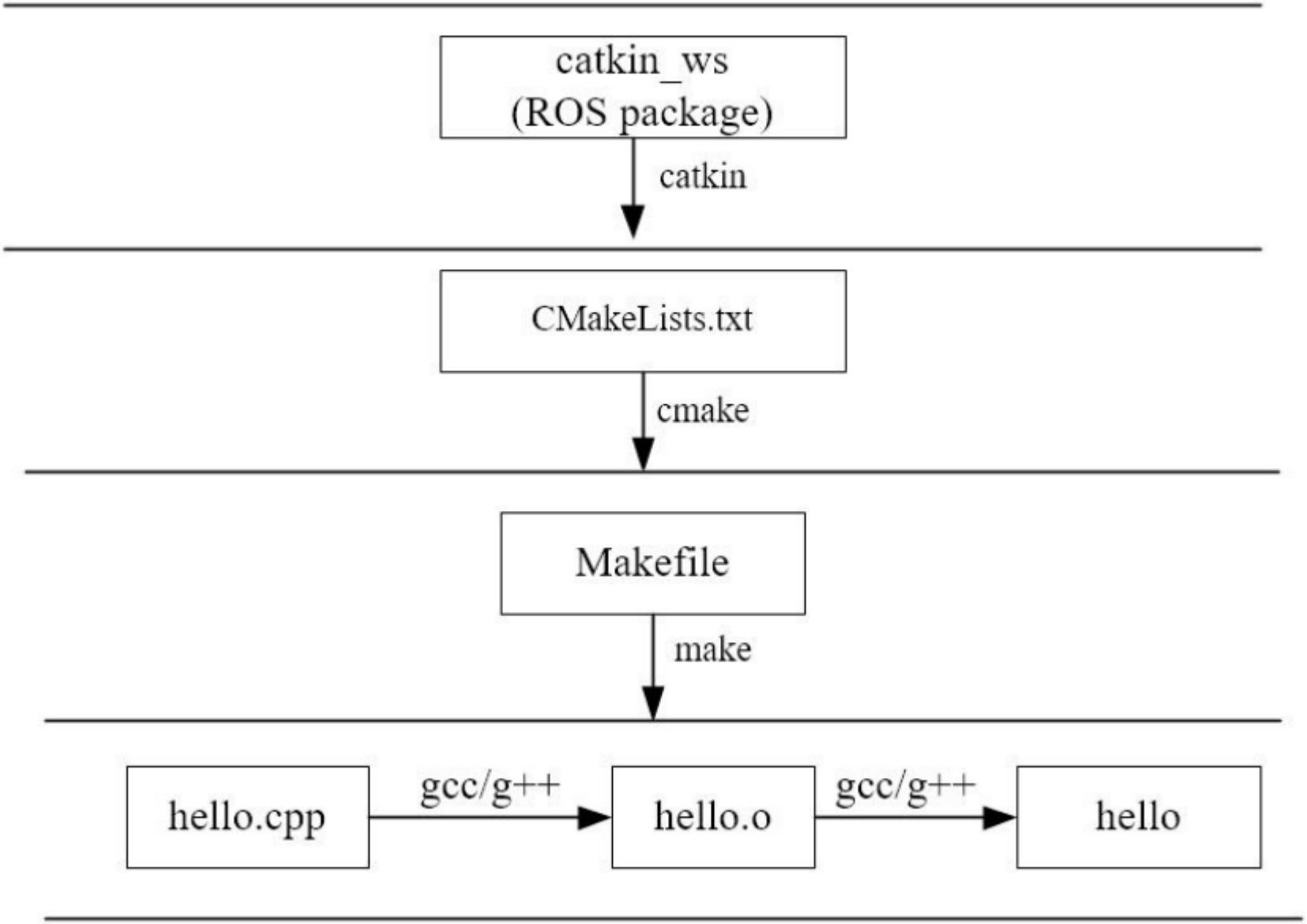
一般 程式


catkin
workspace
src
package1
package2
folder
package3
package4

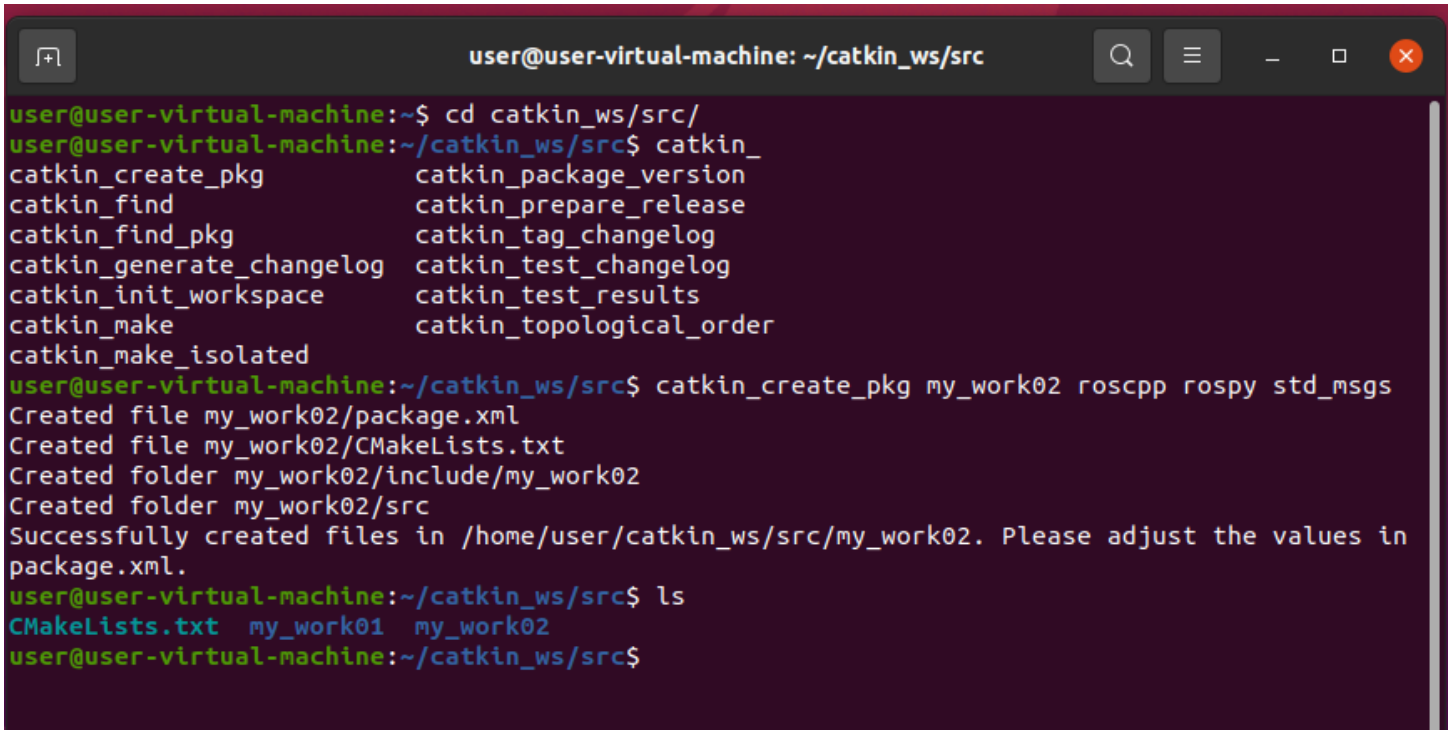

catkin
workspack
build
src
devel
install
(Hardly)
package1
package2
... ...
CMakeList.txt
package.xml
scripts
msg
srv
src
include
launch
*.py
*.sh
*.msg
*.srv
*.cpp
*.h
*.launch



package源代码包　　cmake&catkin缓存和中间文件　　目标文件

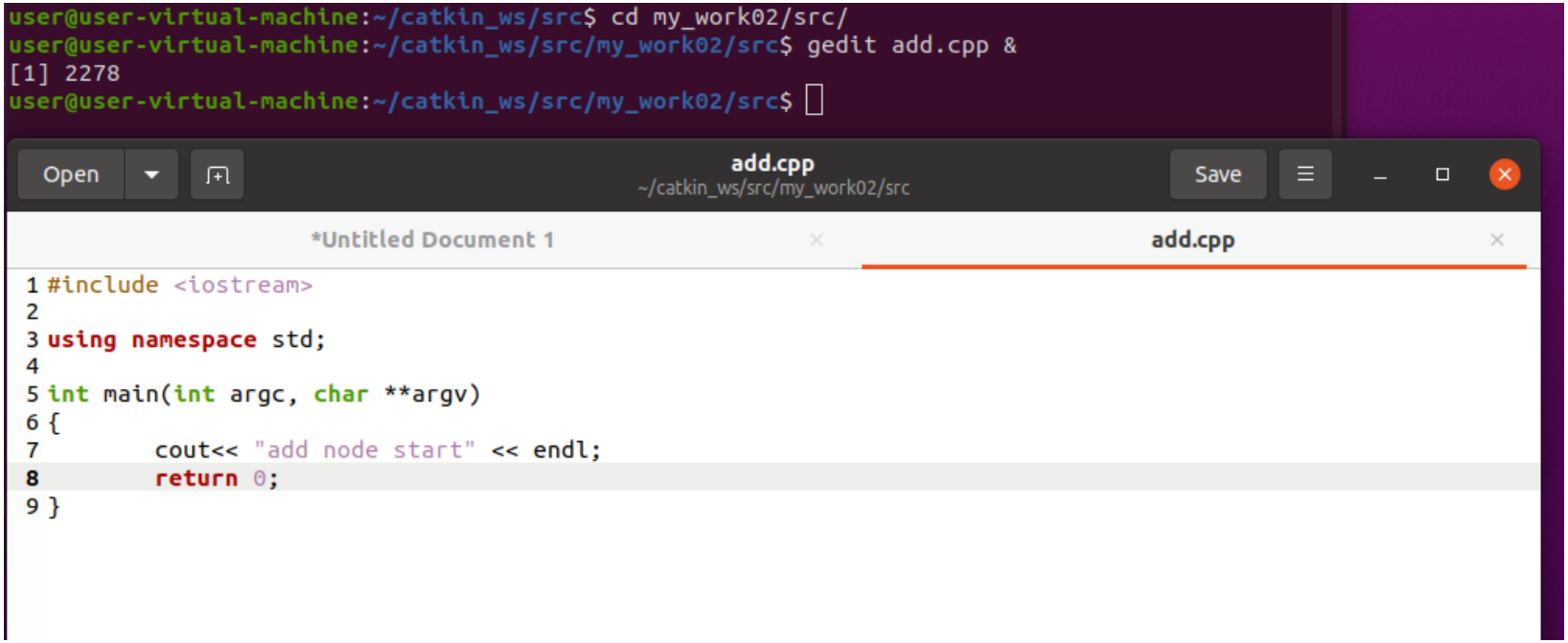

catkin_ws
(ROS package)
catkin
CMakeLists.txt
cmake
Makefile
make
hello.cpp
gcc/g++
hello.o
gcc/g++
hello

一般CPP程式01



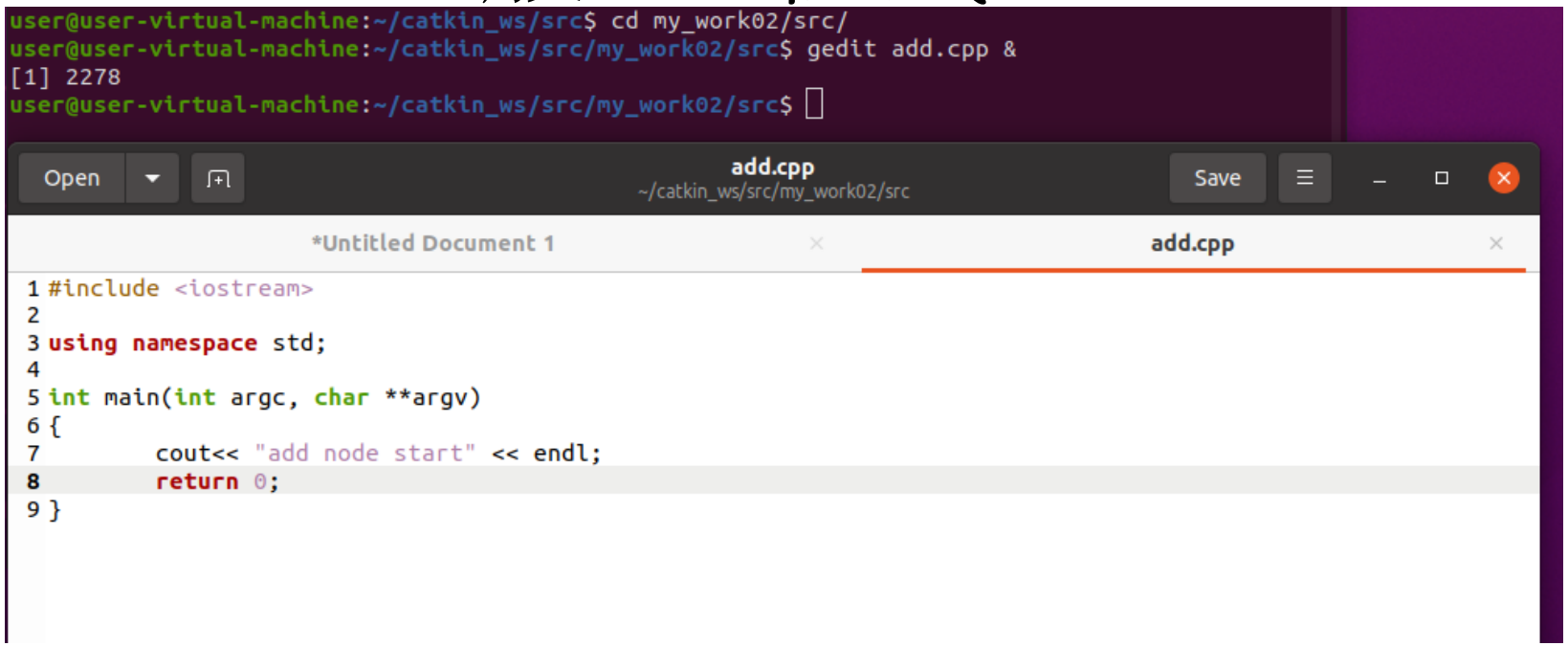
cd catkin_ws/src/
catkin_create_pkg my_work02 roscpp rospy std_msgs
ls

一般CPP程式02



cd my_work02/src/
gedit add.cpp &

一般CPP程式03




```
#include <iostream>

using namespace std;

int main(int argc, char **argv)
{
        cout<< "add node start" << endl;
        return 0;
}
```

一般CPP程式04

```
user@user-virtual-machine:~/catkin_ws/src/my_work02/src$ g++ -o add add.cpp
[1]+  Done                    gedit add.cpp
user@user-virtual-machine:~/catkin_ws/src/my_work02/src$ ls
add  add.cpp
user@user-virtual-machine:~/catkin_ws/src/my_work02/src$ ./add
add node start
user@user-virtual-machine:~/catkin_ws/src/my_work02/src$
```

g++ -o add add.cpp　　編譯程式
ls
./add　　執行程式

一般 程式 增加 ROS功能

一般CPP程式→ros node 01

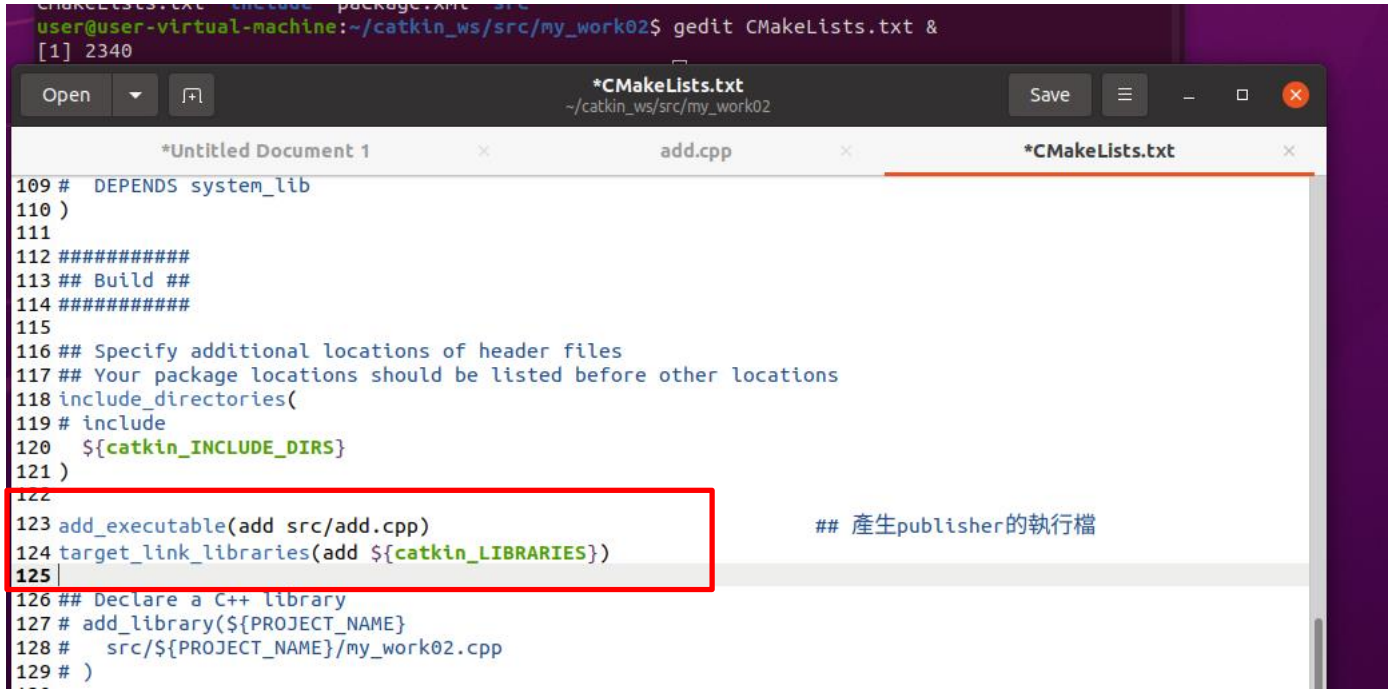
```
user@user-virtual-machine:~/catkin_ws/src/my_work02/src$ gedit add.cpp
```

Open | add.cpp ~/catkin_ws/src/my_work02/src | Save

*Untitled Document 1 | add.cpp

```cpp
#include <iostream>
#include "ros/ros.h"            // 加入ROS公用程序
#include "std_msgs/String.h"    // 所要publish的message header, 在此是std_msgs package底下的String.msg
#include <sstream>

using namespace std;

void cmdCallback(const std_msgs::String::ConstPtr& msg)
{
  ROS_INFO("add heard: [%s]", msg->data.c_str());
}

int main(int argc, char **argv)
{
        ros::init(argc, argv, "adder");
        ros::NodeHandle n;
        cout<< "add node start" << endl;

        ros::Subscriber sub = n.subscribe("cmd", 1000, cmdCallback);
        ros::spin();

        return 0;
}
```

一般CPP程式→ros node 02



add_executable(add src/add.cpp)
target_link_libraries(add ${catkin_LIBRARIES})

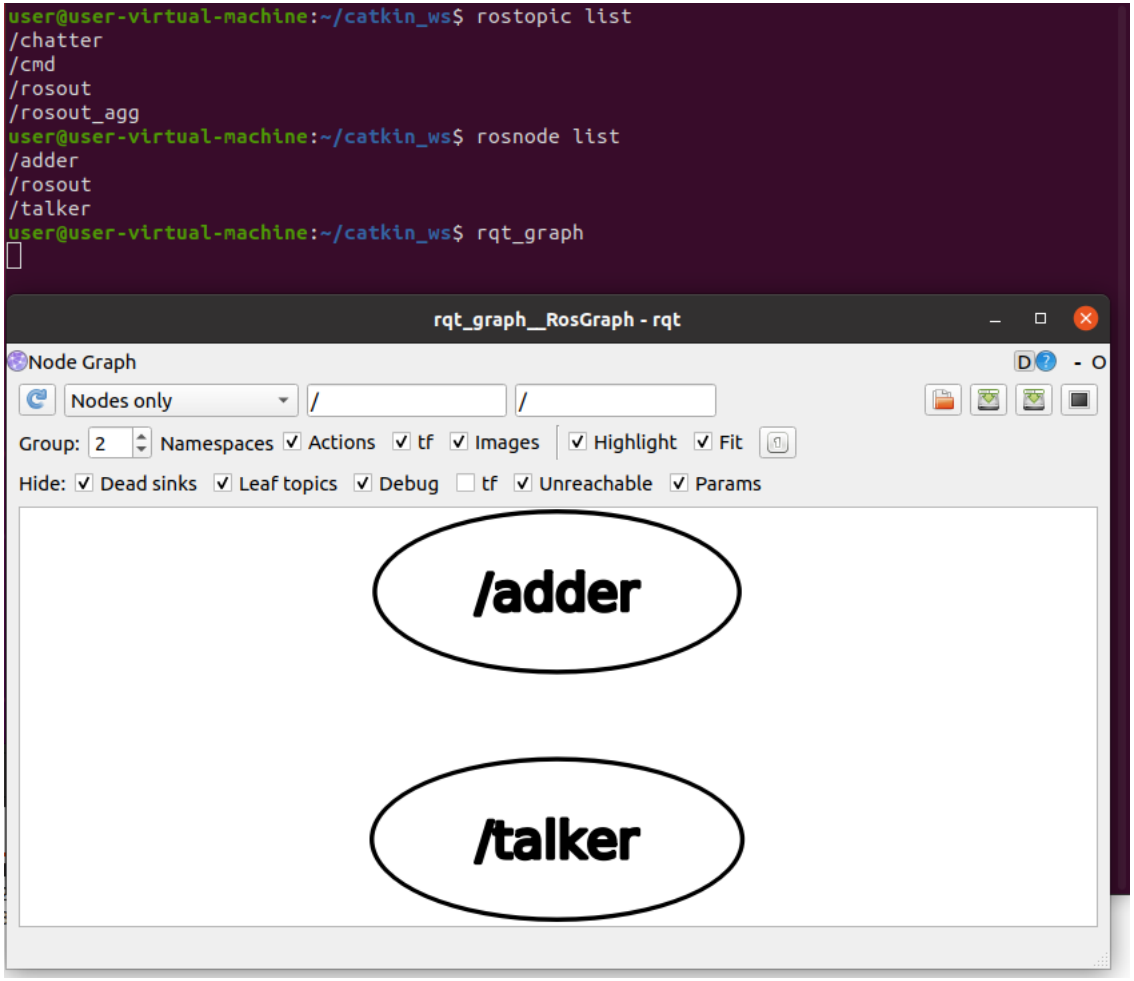
一般CPP程式→ros node 03

```
user@user-virtual-machine:~/catkin_ws/src/my_work02$ cd ../..
[1]+  Done                    gedit CMakeLists.txt  (wd: ~/catkin_ws/src/my_work02)
(wd now: ~/catkin_ws)
user@user-virtual-machine:~/catkin_ws$ catkin_make
Base path: /home/user/catkin_ws
Source space: /home/user/catkin_ws/src
Build space: /home/user/catkin_ws/build
Devel space: /home/user/catkin_ws/devel
Install space: /home/user/catkin_ws/install
####
#### Running command: "cmake /home/user/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home/user/catkin_
ws/devel -DCMAKE_INSTALL_PREFIX=/home/user/catkin_ws/install -G Unix Makefiles" in "/home/user
/catkin_ws/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/user/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /home/user/catkin_ws/devel;/opt/ros/noetic
-- This workspace overlays: /home/user/catkin_ws/devel;/opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10", minimum required is
"3")
```

```
-- ==> add_subdirectory(my_work01)
-- +++ processing catkin package: 'my_work02'
-- ==> add_subdirectory(my_work02)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/user/catkin_ws/build
####
#### Running command: "make -j4 -l4" in "/home/user/catkin_ws/build"
####
[100%] Built target listener
[100%] Built target talk
user@user-virtual-machine:~/catkin_ws$
```

cd ../..
catkin_make

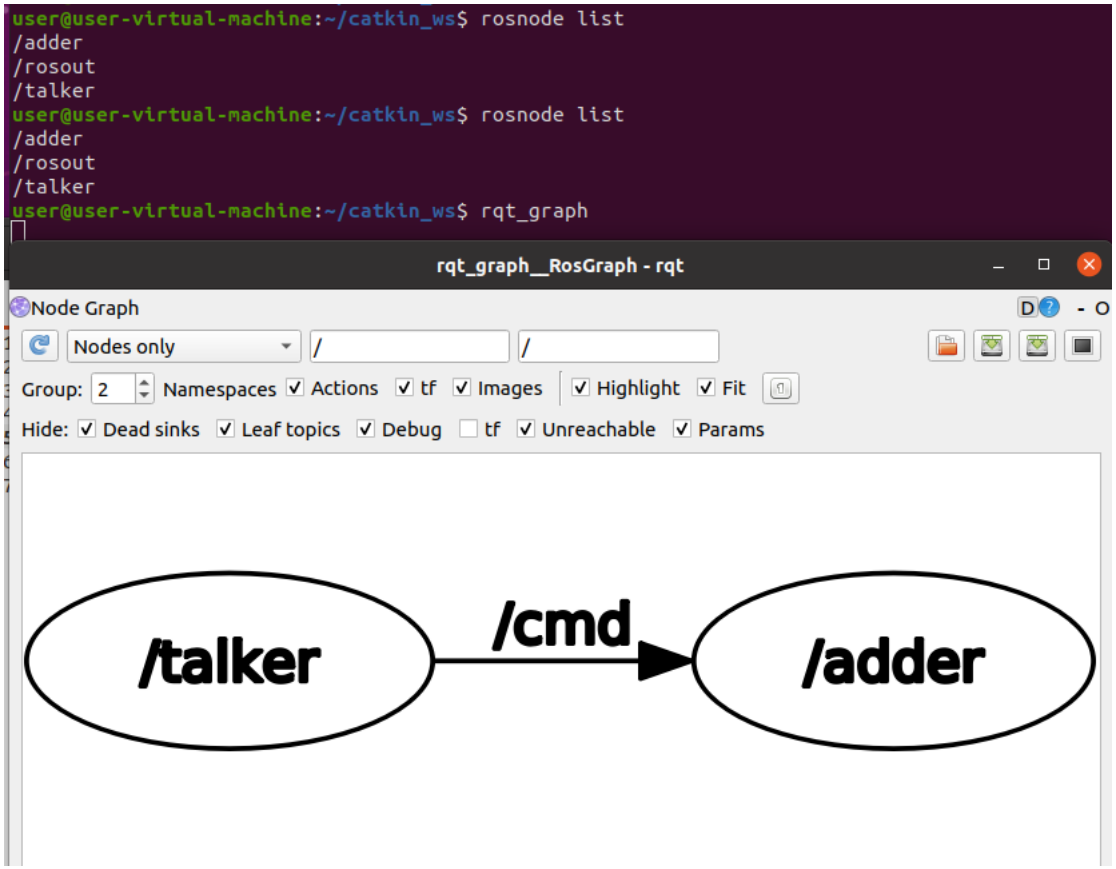
一般CPP程式→ros node 04



roscore
rosrun my_work02 add
rosrun my_work01 talk

rostopic list
rosnode list
rqt_graph

一般CPP程式→ros node 05

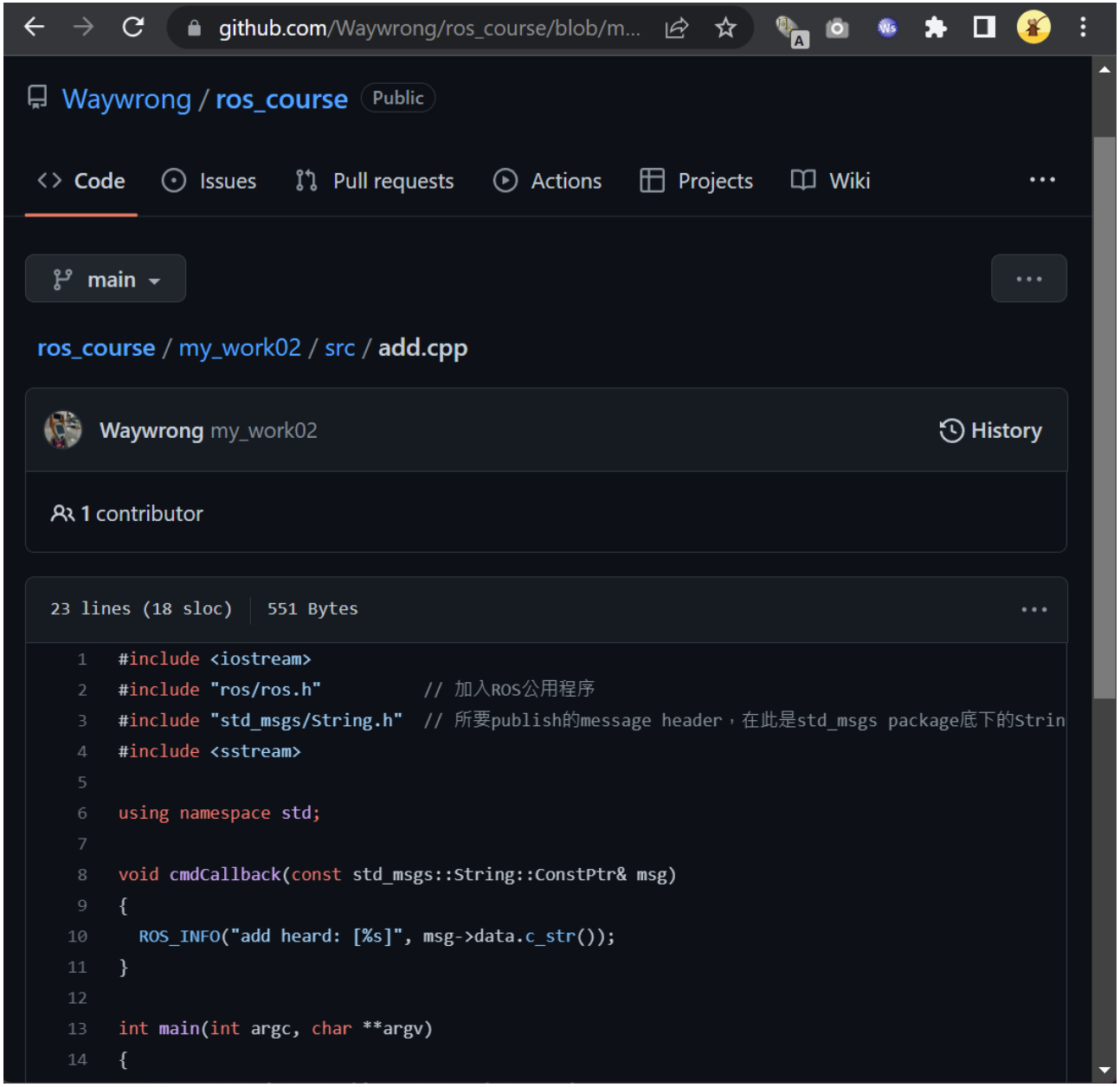


roscore
rosrun my_work02 add
rosrun my_work01 talk /chatter:=/cmd

rostopic list
rosnode list
rqt_graph

参考

https://github.com/Waywrong/ros_course



每次專案有改變時 皆須執行以下(只改設定檔 則不須)

cd ~/catkin_ws

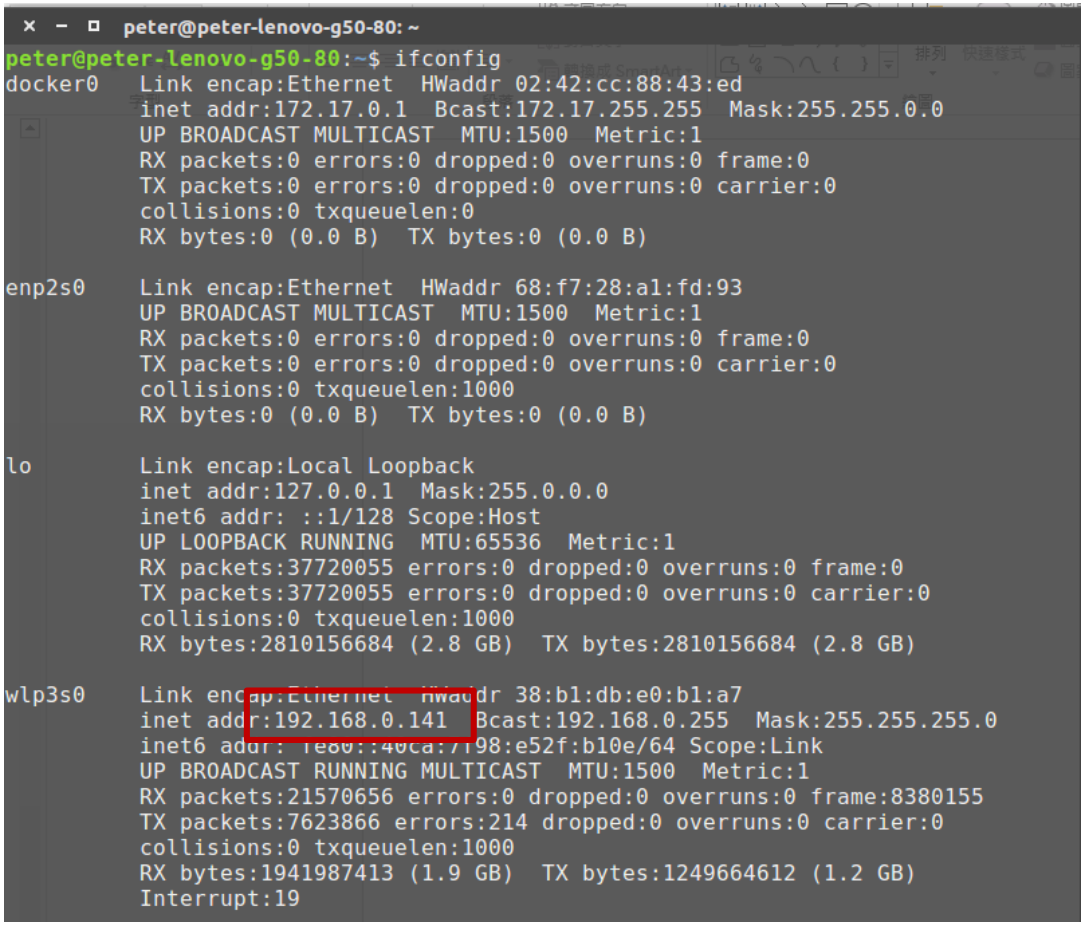
catkin_make

```
peter@peter-lenovo-g50-80:~/catkin_ws/src$ cd ~/catkin_ws/
peter@peter-lenovo-g50-80:~/catkin_ws$ catkin_make
```

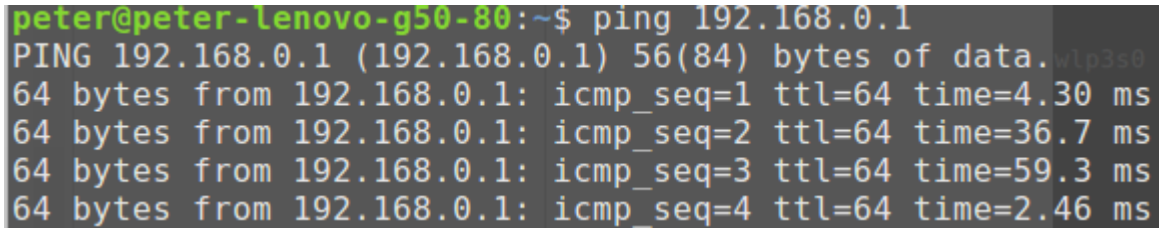
```
peter@peter-lenovo-g50-80:~/catkin_ws$ catkin_make
Base path: /home/peter/catkin_ws
Source space: /home/peter/catkin_ws/src
Build space: /home/peter/catkin_ws/build
Devel space: /home/peter/catkin_ws/devel
Install space: /home/peter/catkin_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/peter/catkin_ws/
build"
####
####
#### Running command: "make -j4 -l4" in "/home/peter/catkin_ws/build"
####
Scanning dependencies of target test_talk
[ 33%] Built target listener
[ 66%] Built target talk
[ 83%] Building CXX object my_work01/CMakeFiles/test_talk.dir/src/test.cpp.o
[100%] Linking CXX executable /home/peter/catkin_ws/devel/lib/my_work01/test_tal
k
[100%] Built target test_talk
peter@peter-lenovo-g50-80:~/catkin_ws$
```

多主機通訊

Ifconfig 取得IP資訊



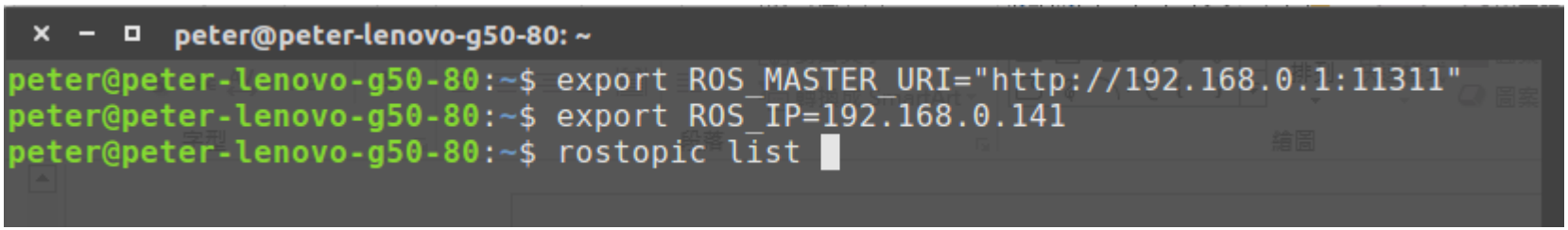
ping 啟動roscore的電腦IP，確認雙方通訊正常

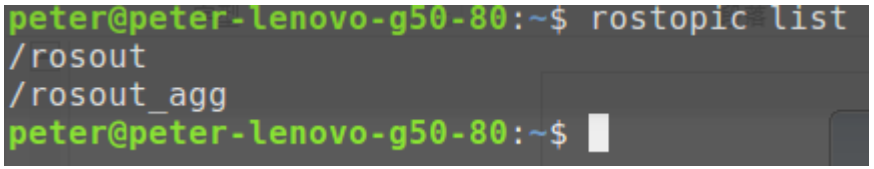


export ROS_MASTER_URI="http://192.168.0.1:11311"　開啟roscore電腦IP
export ROS_IP=192.168.0.141　所在電腦IP

開啟roscore電腦(192.168.0.1) e.g.機器人

所在電腦(192.168.0.141) e.g.筆電



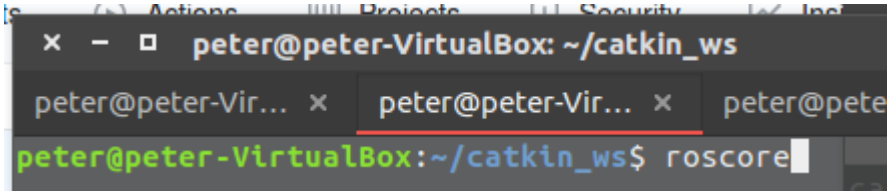


export ROS_MASTER_URI="http://192.168.50.68:11311"
export ROS_IP=192.168.50.XX

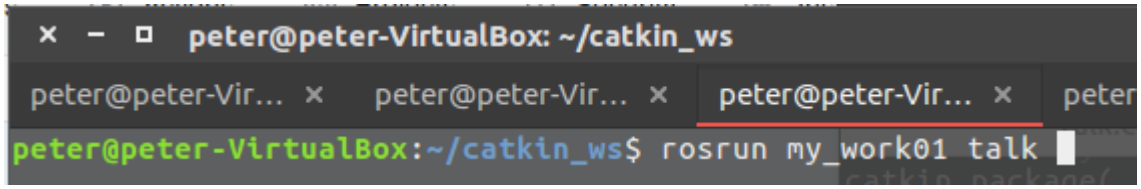
rostopic list

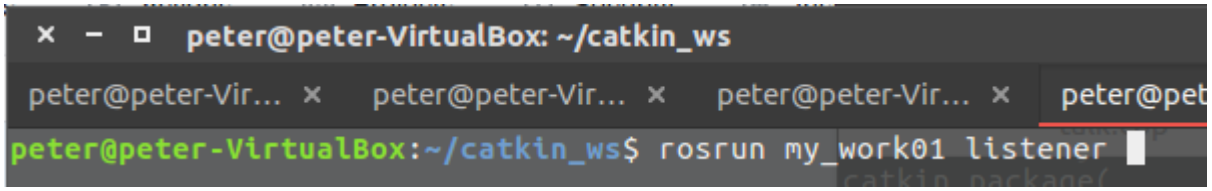
ROS多機運作

- 1. 兩人一組
- 2. 其中一人先執行roscore
- 3. 一台機器執行 talk
  一台機器執行listener
4. 兩人角色對調



source ~/catkin_ws/devel/setup.bash





作業2

- 建立專案(my_hw02)，並上傳相關檔案

- 參考3/28上課內容，"ROS-Class-5.pdf“
- 模仿 my_work01/src/talk.cpp ，寫一個會發布訊息(rostopic)的專案(my_hw02)
- 計分部分包含
  - 1. 修改rostopic 名稱
  - 2.修改rosnode 名稱
  - 3. 接收訊息的listener, 可用my_work01/src/listener.cpp(如自行修改另有計分)
  - 4. 紀錄實驗過程於word檔，紀錄所下的命令與回應，可多利用截圖(圖文並茂加分)
    - rosnode list
    - rostopic list
    - rqt_graph

- 上傳作業包含 (4/11 前上傳):
- 1. CMakeLists.txt
- 2. package.xml
- 3. XXX.launch (檔名自訂)
- 4. 修改過的cpp檔
- 5. 實驗紀錄word檔