

# 第5章 LINUX常用指令及C程式語言

# 5.1 LINUX常用指令

Linux屬於開放原始碼的系統，開發人員不用花費到任何的費用就可以獲得Linux原始程式碼，並可以根據自己的需要使用或進行必要的修改，而Linux以它的高效率和高靈活性著稱，Linux系統不僅只有作業系統，還包含了文字編輯器、高階語言編輯器等應用程式，並提供帶有多個視窗管理器的圖形化使用者介面，使用者可以使用它像在使用Windows系統一樣，使用視窗或是圖示或是選單對於初學者來說比較輕易上手，由於本系統開發平台以Linux為主體所以我們在此針對系統平台常用的Linux指令做說明使用方式，可以詳細了解指令後，在而後的實驗中實際操作並深刻了解相關指令的用法。

## 5.1.1 系統管理

- ◎ 在此要特別說明的是，本章節僅提供實驗中有出現的相關指令作一說明，若需要深入了解其他指令用法使用系統內建之文件查詢指令“man”來查詢。以下是常用Linux相關指令的介紹。

# 1.CLEAR指令

clear指令	清除螢幕資訊
目標	能清除雜亂的螢幕資訊
輸入語法	clear
功能說明	輸入clear可以將雜亂的螢幕資訊清除乾淨
範例	說明
clear	清除畫面

## 2.DATE指令

date指令	顯示或設定系統時間
目標	能以指定格式顯示時間或設定目前系統時間
輸入語法	date[-參數][日期與時間格式]
功能說明	date指令可根據指定的格式顯示目前的時間或設定系統時間
範例	說明
date -d<"字串">	顯示字串時間
date -s<"字串">	設定字串時間

# 3.ECHO指令

echo指令	字串輸出
目標	將指定的字串值輸出到標準輸出
輸入語法	echo[-參數] 字串
功能說明	echo指令會將指定的字串值輸出到標準的輸出裝置
範例	
echo -n	最後不自動換行
echo -e	\b 刪除前一個字元 \n 換行且游標移至行首 \t 插入tab字元

# 4.REBOOT指令

reboot指令	重新啟動
目標	使用reboot指令將系統重新啟動
輸入語法	reboot[-參數]
功能說明	用reboot指令重新啟動電腦
範例	說明
reboot -f	強制重新開機
reboot -i	關閉網路後，再重新開機
reboot -n	儲存資料後，再進行重新開機

# 5.SHUTDOWN指令

shutdown指令	重新啟動
目標	將系統關機
輸入語法	shutdown [-參數][時間][警告訊息]
功能說明	使用shutdown指令可以將系統進行關機或重新啟動，
範例	說明
shutdown -c	中斷關機
shutdown -h	關機
shutdown -r	重新開機
shutdown -t<時間>	延遲指定時間
shutdown<時間>	指定一個時間關機



# 6.SUDO指令

sudo指令	用其他身份去執行指令
目標	不切換的情況下以系統中其他身份來執行指令
輸入語法	sudo[-參數][指令]
功能說明	sudo指令的預設身份是root，即可使用最高權限下執行指令。
範例	說明
sudo -l	列出其他身份可執行與不可執行的指令
sudo -l	顯示sudo的設定
sudo -i	將使用者權限切換至root

# 7. MAN指令

man指令	說明手冊
目標	檢視shell指令的說明資訊
輸入語法	man[-參數] 指令名
功能說明	man指令可以用來尋找系統中已安裝完成的指令說明手冊，尋找說明手冊的路徑是指定到設定檔/etc/man.config中。
範例	說明
man help	顯示說明手冊的資訊
man -a	列出所有符合條件的資訊
man -K<字串 >	在整個手冊裡尋找指定字串
man -M<路徑 >	指定搜尋手冊檔案的路徑，各路徑由冒號分隔
man -S<名稱 >	在指定的章節搜索指定並顯示手冊資訊

# 8. DMESG指令

dmesg指令	顯示核心資訊
目標	顯示核心中ring buffer緩衝區的資訊
輸入語法	dmesg[-參數]
功能說明	Dmesg指令可透過檢測核心的緩衝區，來顯示目前系統開機及執行中核心的輸出資訊
範例	說明
dmesg	輸出系統的開機資訊

# 可用參數及其說明如下

參數	說明
-c	顯示資訊後清空緩衝區中的資訊
-n<等級>	設定紀錄緩衝區中資訊的等級
-s<位元組數>	設定緩衝區中的大小

## 5.1.2 文件管理

# 1.LS指令

ls指令	列出文件或目錄下之文件名稱
目標	能清楚知道文件或目錄下有什麼檔案
輸入語法	ls
功能說明	輸入ls可以得到文件或目錄下有什麼檔案
範例	說明
ls -a	列出以.起始的所有文件隱藏檔案名稱
ls -l	列出目錄下所有文件的所有權、擁有者、文件大小、修改日期、及名稱
ls -R	顯示目錄下，以及所有子目錄之文件名稱

## 2.CD指令

cd指令	改變目錄位置
目標	使用以下指令可以改變目錄位置
輸入語法	cd [參數]
功能說明	輸入cd [參數]可以改變目錄位置
範例	說明
cd ~	使用者的起始目錄
cd ~ username	指定特定user的起始目錄
cd .	目前的工作目錄
cd ..	目前目錄的上一層

# 3.CP指令

cp指令	複製文件
目標	使用cp指令可以複製文件檔案
輸入語法	cp [參數] 來源目標
功能說明	輸入cp [參數]可以改變目錄位置
範例	說明
cp file1 file2	將file1複製成file2
cp file1 dir1	將file1複製到dir1路徑下，file1檔名不變。
cp -r dir1 dir2	將dir1整個目錄，複製到dir2的目錄下，若是沒有dir2目錄，會將dir1這個目錄複製為dir2。



# 可用參數及其說明如下

參數	說明
-r	將目錄下的所有檔案與子目錄一起處理

# 4.MV指令

mv指令	移動文件
目標	使用mv指令可以移動文件檔案
輸入語法	mv [參數] 來源目標
功能說明	輸入mv可以用來移動更改文件、目錄名稱
範例	說明
mv file1 file2	將file1改變文件名稱為file2
mv file1 dir1	將file1移動到dir1路徑下，file1檔名不變。
mv dir1 dir2	將dir1整個目錄，移動到dir2的目錄下，若是沒有dir2目錄，會將dir1這個目錄更名為dir2。

# 5. MKDIR指令

mkdir指令	建立新目錄
目標	使用mkdir指令可以讓使用者建立新目錄
輸入語法	mkdir [name]
功能說明	輸入mkdir [name]可以建立新目錄
範例	說明
mkdir dir1	建立新目錄，並將目錄名稱命名為dir1

# 6. RM指令

rm指令	刪除文件、目錄
目標	使用rm dir1指令可以讓使用者刪除文件、目錄
輸入語法	rm [-r] name
功能說明	輸入rm [name]可以刪除文件、目錄
範例	說明
rmdir dir1	刪除目錄dir1，不過文件下必須沒有文件存在
rm -r dir1	刪除目錄dir1，以及刪除所有子目錄

# 7. PWD指令

pwd指令	查出目前目錄的位置
目標	讓使用者知道目前的位置
輸入語法	pwd
功能說明	pwd指令可以獲得目前位置資訊
範例	說明
pwd	查出目前使用者所在位置

# 8.GREP 指令

grep指令	找尋文件
目標	文件之字串找尋
輸入語法	grep string file
功能說明	輸入一個字串在某個檔案下找尋
範例	說明
grep abcd file1	找尋file1文件中，並列出字串abcd之內容所在位置

## 5.1.3 文件模式設定

需要在最高權限下才可以順利執行以下指令

# 1. CHMOD指令

chmod指令	設定權限
目標	設定文件或目錄的權限
輸入語法	chmod [-R] 模式名稱
功能說明	改變文件或目錄可讀、可寫、可執行之權限 模式：rwx rwx rwx，r:讀，w:寫，x:可執行 使用者:user group other，u:使用者，g:群組，o:其他使用者
範例	說明
chmod 777 dir1	將目錄dir1，設定成任何使用者都皆可讀、可寫、可執行
chmod 700 file1	將文件file1，設定只有檔案擁有者才可讀、可寫、可執行
chmod o+x file2	將文件file2，增加其他使用者可執行的權限
chmod g-r dir2	將目錄dir2，減少群組可讀的權限



## 2.CHOWN 指令

chown指令	改變文件、目錄的擁有權
目標	設定擁有權
輸入語法	chown [-R] 使用者名稱 檔案名稱
功能說明	改變文件擁有權至指定的使用者
範例	說明
chown user file1	將文件file1，改成使用者user所擁有
chown -R user dir1	將目錄dir1及其子目錄，改成使用者user所擁有

## 5.2 C程式語言

由於嵌入式Linux系統開發皆以C語言為基礎，因此本節也針對C語言特性及開發流程分別作一個簡單的介紹。

## 5.2.1 C程式語言簡介

語言的用途是用來作為人與人之間溝通的橋樑，一般人用來溝通的語言稱為自然語言 (Natural Language)，電腦使用的語言稱為程式語言 (Programming Language)，而程式語言的種類大致上分為低階語言 (Low-level Language) 及高階語言 (High-level Language) 之分，機器語言 (Machine Language) 是屬於低階語言，為電腦硬體實際執行的語言，就是一連串的0、1二進位數字所組合而成。

## 5.2.1 C程式語言簡介

組合語言(Assembly Language)的表示方法接近機器語言，使用我們較容易記憶的形式，來對應一連串的0、1二進位的數值組合，雖然組合語言不易學習，不過寫出來的程式碼，其執行效率較高階語言好很多。不過，由於高階語言最接近人類的語言習慣，如C語言、C++、BASIC、JAVA語言等，這些語言較容易學習。本節從C程式語言介紹開始，讓使用者了解C程式語言的特性及開發流程，進而實際學習C語言程式的架構與編譯器操作。

## 5.2.2 C程式語言介紹

C程式語言是最被常使用在系統底層的高階語言，於1970年為UNIX系統所開發，因而發展出來的一種程式語言，不過在1980年，為了避免各個開發廠商所使用的C語言語法會有誤差，美國國家標準局替C語言訂定一套完整的標準語法，統稱ANSI C，而目前開發工具都支援ANSI C語法的程式。

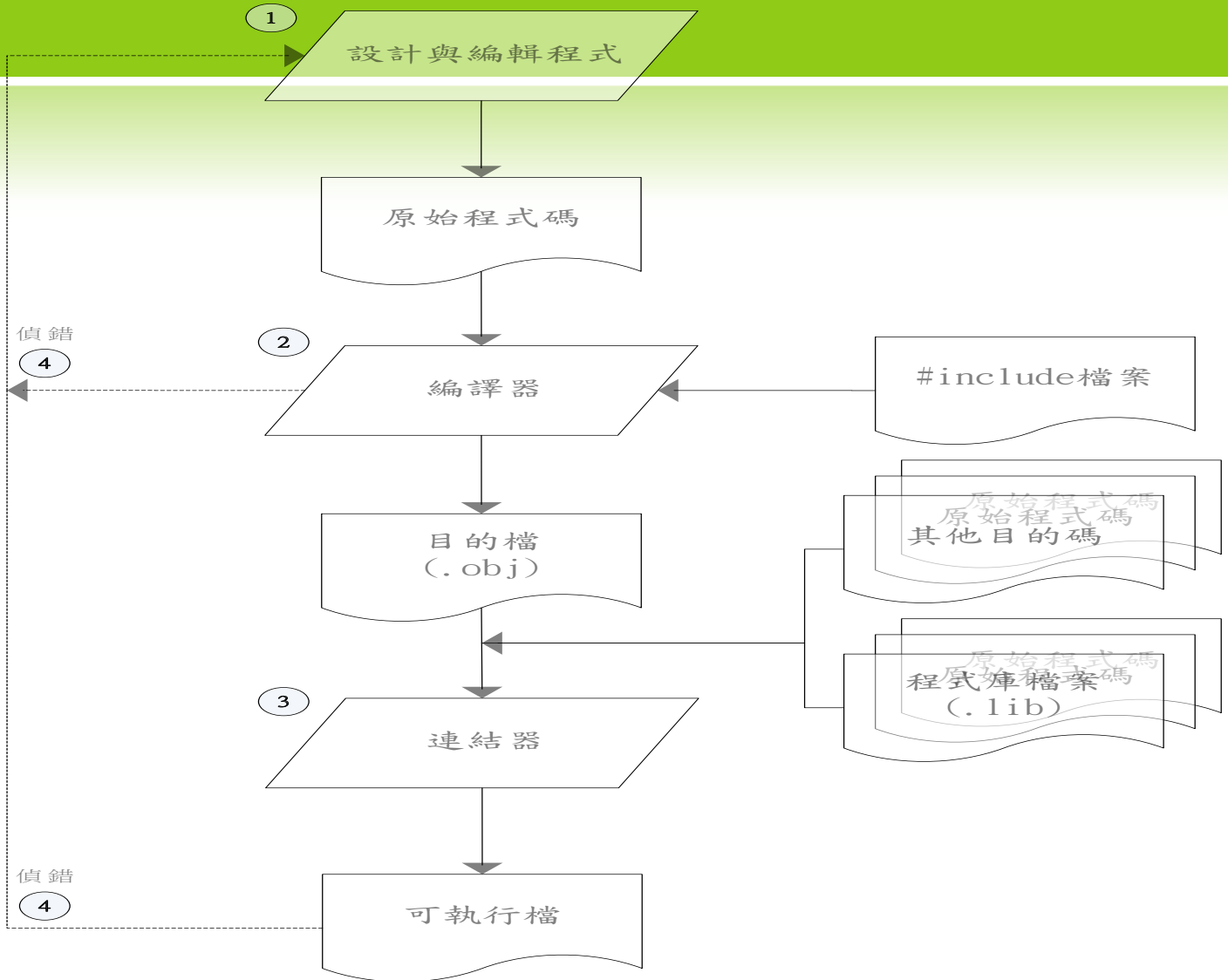
## 5.2.3 C程式語言的開發流程

- ◎ 設計與編輯程式，可以使用任何的純文字編輯器來編輯C程式語言，如Windows的筆記本或Wordpad、Linux的Gedit或Vim等，需要將副檔名存成(.c)的原始程式檔。
- ◎ 使用編譯器來編輯原始程式檔，如GCC編譯器的gcc/g++，編譯結果會產生目的檔。

## 5.2.3 C程式語言的開發流程

- ◎ 目的檔程式及其他目的碼程式或是程式庫使用連結器連結成一個可執行檔，目前大多數的編譯器都必須具備連接器的功能。
- ◎ 編譯過程或連接過程發生錯誤或是邏輯錯誤而無法完成程式，都必須進入偵錯階段，所以程式工程師需要不斷的除錯(Debug)，直到程式完成為止，圖5.1是整個C程式語言的開發流程圖

# 5.2.3 C程式語言的開發流程





## 5.2.4 C程式語言基本架構

C程式語言基本架構主要是由程式註解、前置處理指令以及程式內容所組合而成的，如圖5.2 C程式語言基本架構圖，而最基本的簡單範例，請詳見附錄。

## C程式語言架構

```
/*註解*/
```

```
#include <...>  
#define ....
```

```
main(...)
```

```
{
```

程式內容

```
}
```

## 簡單範例 V.S. 基本架構

```
/*檔名:Hello-UBIKIT-6612*/  
/*功能:簡單的C程式範例*/
```

```
#include <stdio.h>
```

```
main(void)
```

```
{
```

```
Printf("Hello, UBIKIT-6612!!\n");
```

```
}
```

圖5.2 C程式語言基本架構圖

## 5.2.4 C程式語言基本架構

- ◎ 程式註解部分:在/\*與\*/之間的文字符號都將會被編譯器忽略，所以說這些註解並不會影響程式的正確與否，這些文字可以用來當作說明該程式或該片斷，有助於日後快速了解程式，而註解符號為/\*註解內容\*/。
- ◎ 前置處理指令部分:前置處理指令是指前面出現#符號的指令，比如說#include的功用在於引入標頭檔，標頭檔就是放置已經撰寫完成函式內容的函式庫資料，範例中的stdio.h的標頭檔中定義了printf()與scanf()函式的內容，所以我們如果要使用到printf()與scanf()函式，則需要使用#include將這些內建函式庫檔案引入程式中。

## 5.2.4 C程式語言基本架構

- ◎ 程式內容部分:由各個不同功能的函式所組成，並且函式與函式之間可以透過呼叫以及回傳值方式加以聯繫，main函式有以下兩個特點:
  - ◎ 1.唯一性:在C函式，任何內建函數都需要具備有唯一的特性。
  - ◎ 2.必要性:為了讓作業系統能夠找到程式的進入點，因此不可以省略及缺少main函式。

# 5.2.5 C程式語言編譯器

- ◎ GCC是一個開源程式，可編譯C/C++的編譯器，通常我們使用gcc指令來編譯C 語言程式，而使用g++來編譯C++語言，gcc與g++的參數大同小異，我們可以透過各種參數，指定gcc只做某個步驟的動作，或者更改輸出執行的檔案名稱，gcc語法如下：

- ◎ gcc語法 

gcc [ option   filename ].....
--------------------------------

## 5.2.5 C程式語言編譯器

- ◎ 語法說明: option是個參數，通常是以“-“當作起始符號，filename是個檔案名稱，他可以是輸入檔名或輸出檔名，由於gcc參數過多，只介紹幾個重要的參數加以說明。
- ◎ gcc參數分類
- ◎ gcc參數(option)分類:
  - ◎ 一般參數(-o, -S, -c)、前置處理參數(-E, -C)、最佳化參數(-O)、警告參數(-w, -Wall)、除錯參數(-g)、標頭檔目錄參數(-I)、連接檔目錄或名稱參數(-L, -l)。

# GCC常用參數

參數	用途
-o filename	指定編譯輸出的目的檔檔案名稱或執行檔檔案名稱
-S	產生組合語言程式碼
-c	將程式編譯成目的檔而不做程式庫的連接。
-On	指定最佳化程度，n為程度數字
-Wall	顯示所有警告資訊
-g	加入gdb的除錯功能
-ldir	設定標頭檔的預設目錄
-Ldir	設定函式庫的預設目錄
-lname	指定連接韓式庫的檔案名稱

範例:HELLO-UBIKIT-6612.c的編譯，  
原始檔在附錄A中第五章的目錄中。

◎ Step1:

首先，先建立Hello-UBIKIT-6612.c

Hello-UBIKIT-6612.c程式碼如下:

```
1 /*檔名:Hello-UBIKIT-6612*/  
2 /*功能:簡單的C程式範例*/  
3 #include <stdio.h>  
4 main(void)  
5 {  
6     printf("Hello, UBIKIT-6612!!\n");  
7 }
```



範例:HELLO-UBIKIT-6612.c的編譯，  
原始檔在附錄A中第五章的目錄中。

- ◎ Step2:在開發的客戶端(PC端) Ubuntu系統終端機中輸入指令:

gcc Hello-UBIKIT-6612.c -o hello-ubikit-6612 ,  
如圖5.3所示。

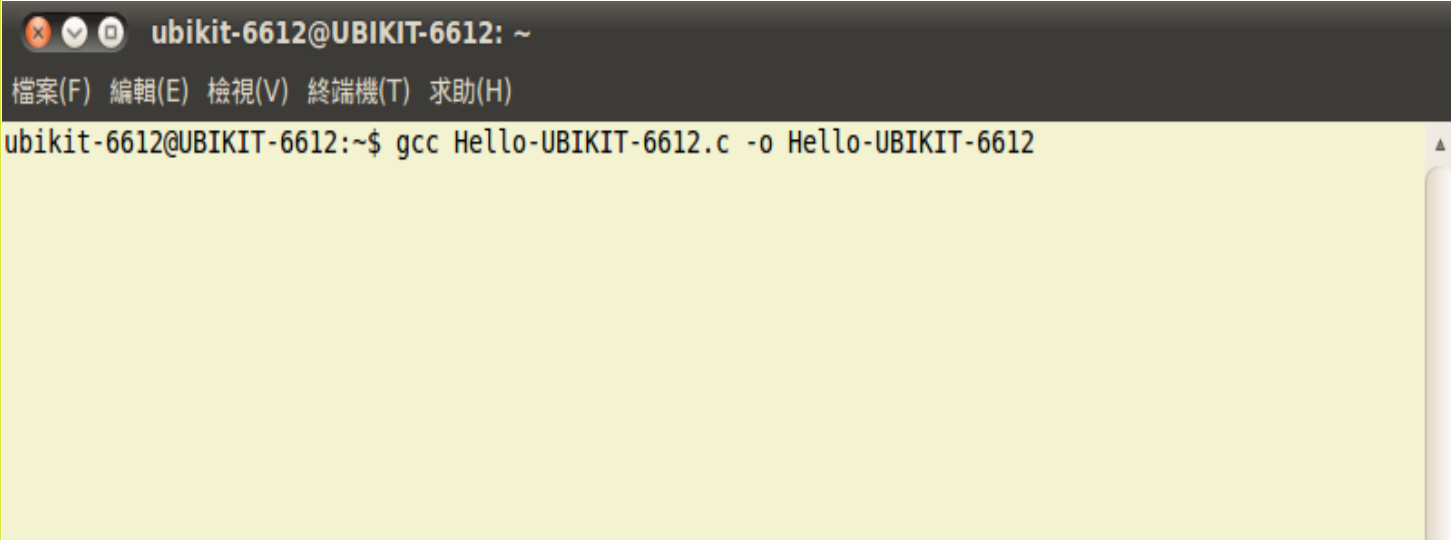
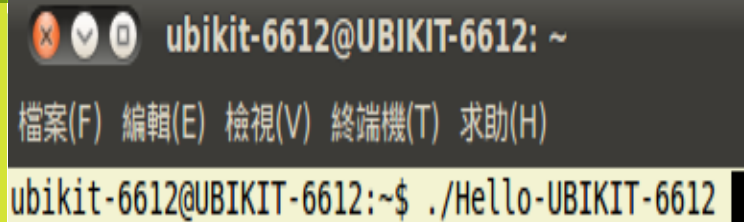
A screenshot of a terminal window. The title bar shows 'ubikit-6612@UBIKIT-6612: ~'. Below the title bar, there are menu options: '檔案(F) 編輯(E) 檢視(V) 終端機(T) 求助(H)'. The main content of the terminal shows the command 'ubikit-6612@UBIKIT-6612:~\$ gcc Hello-UBIKIT-6612.c -o Hello-UBIKIT-6612' entered at the prompt. The terminal background is light yellow, and the text is black.

圖5.3輸入編譯指令

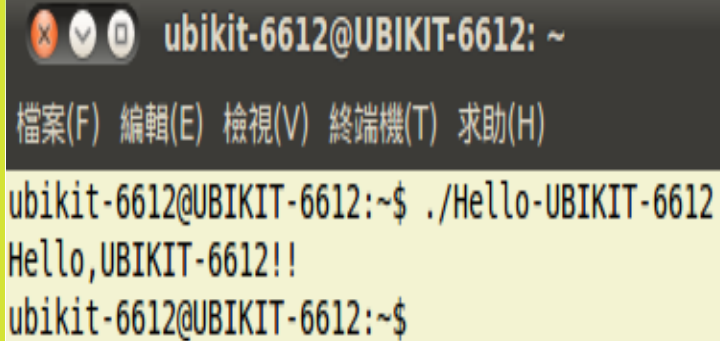
Step3:編譯完成後輸入指令: ./Hello-UBIKIT-6612 , 如圖5.4所示。

A terminal window with a dark grey title bar. The title bar contains three window control icons (close, maximize, and refresh) on the left and the text 'ubikit-6612@UBIKIT-6612: ~' on the right. Below the title bar, the text '檔案(F) 編輯(E) 檢視(V) 終端機(T) 求助(H)' is displayed. The main area of the terminal is light yellow and shows the command 'ubikit-6612@UBIKIT-6612:~\$ ./Hello-UBIKIT-6612' followed by a black cursor bar.

```
ubikit-6612@UBIKIT-6612:~$ ./Hello-UBIKIT-6612
```

圖5.4輸入執行Hello-UBIKIT-6612指令

Step4:執行結果如圖5.5。

A terminal window with a dark title bar containing the text 'ubikit-6612@UBIKIT-6612: ~'. Below the title bar is a menu bar with the text '檔案(F) 編輯(E) 檢視(V) 終端機(T) 求助(H)'. The main area of the terminal shows the following text:

```
ubikit-6612@UBIKIT-6612:~$ ./Hello-UBIKIT-6612  
Hello,UBIKIT-6612!!  
ubikit-6612@UBIKIT-6612:~$
```

圖5.5執行Hello-UBIKIT-6612

## 5.3 C語言教學資源導引

以下是一些免費的線上網路教學網站

- ◎ C語言入門教室

([http://imil.au.edu.tw/~hsichcl/C\\_OnLine.htm](http://imil.au.edu.tw/~hsichcl/C_OnLine.htm))

- ◎ C語言程式教學

(<http://www.cmlab.csie.ntu.edu.tw/~perng/course/>)

- ◎ C Language Tutorial

([http://einstein.drexel.edu/courses/Comp\\_Phys/General/C\\_basics/](http://einstein.drexel.edu/courses/Comp_Phys/General/C_basics/))

- ◎ C程式簡介

(<http://140.127.40.1/~jwu/c/cpg.htm>)

# 問題與討論：

- ◎ 請各列出兩個高階語言與一個低階語言？
- ◎ 請列出兩個C語言的前置處理器指令？
- ◎ 使用陣列的結構寫出一個九九乘法表？
- ◎ 用迴圈的方式畫出一個菱型？
- ◎ 使用C語言寫一個遞迴的程式來求出 $n!$ 的值？