

lex

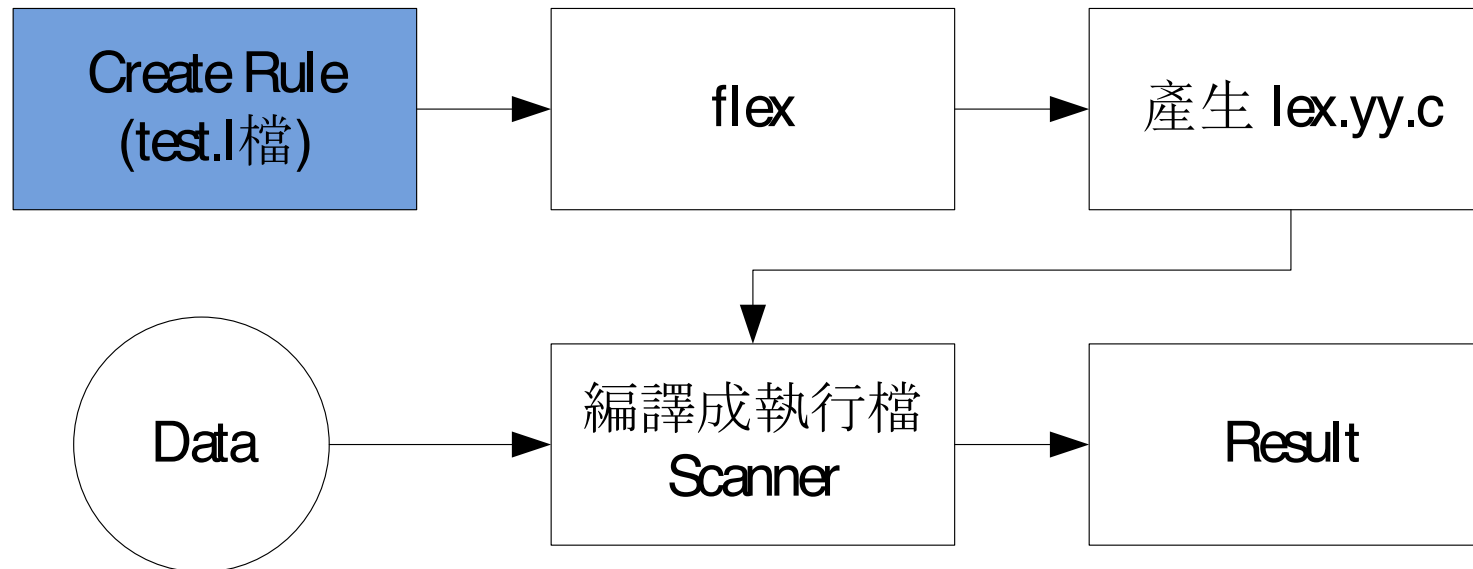
-A scanner generator

Outline

- Introduction
- Installation
- Homework

Introduction (1)

- What is lex
 - lex是linux系統中標準的詞法分析器產生程式



lex建立規則流程

Introduction(2)

- Format of Input File
 - 分為3個部分，並以%%隔開

```
definition //標頭檔 宣告變數  
%%  
rules //規則建立  
%%  
user code //主程式
```

Introduction(3)

- Definitions Section:
 - declarations of simple *name* definitions to simplify the scanner specification, and declarations of *start conditions*
 - *start conditions* 必須用%{ %}包住

```
DIGIT    [0-9]
ID       [a-z][a-z0-9]*
```

```
%{
    #include<stdio.h>
    #include<stdlib.h>
    int Total_Num = 0 ;
}%
```

Introduction(4)

- Rules Section
 - *Form: Pattern action*

```
[0-9]+ {  
        printf("integer : %d\n",yytext);  
        TotalNum++;  
    }
```

Introduction(5)

- Patterns

| 型式 | 規則描述 | 範例 |
|-----------------------|------------------------|-----------|
| ABC...abc...123 .. | 構成了部份的字串以及數字 | printf |
| . | 代表任意字串or數字，(\n除外) | . |
| - | 用來指定範圍 | A-Z |
| [] | 一個字串的集合，只要符合[]中任一個字便成立 | [abc] |
| * | 符合0個或多個 | [0-9]*A |
| + | 符合1個或多個 | [0-9]+A |
| ? | 符合0個或1個 | [0-9]?A |
| {} | 用來表示某個字可能出現的次數 | A{1,3} |
| ^ | 否定符號 | ^[A-Z] |
| | 或 | Abc 123 |

Introduction(6)

- Patterns example:
 - `[0-9]+[A-Z]?` : 09998A 、 123C 、 22.....
 - `A{1,3}[abc]` : Aa,Ab,Ac,AAAa,AAAb,AAAc
 - `Printf[abc]` : Printf a 、 Printf b 、 Printf c
 - `\(+` : (、 ((、 (((、 ((((、
 - `[A-Z0-9]+` : N95 、 RS232 、 Z454E.....

Introduction(7)

- User Code
 - 利用main()來執行

```
int main()
{
    yylex();
    return 0;
}
```

Introduction(8)

- Global Variable
- 下列是lex預設的變數，在寫.l檔時，我們可以直接使用這些變數：
 - **yytext**：當Rule中有符合的字串時，符合的字串就會存放在**yytext**中，型態為**char***。
 - **yylen**：記錄著**yytext**的長度。
 - **yylineno**：記錄目前的**yyin**讀到第幾行。
 - **yyin**：是lex的input來源，型態為**FILE**，預設為**stdin**。

Introduction(9)

- 數字統計

```
%{  
    #include <stdio.h>  
    int Total_Num=0;  
%}  
%option noyywrap ← 分析單一文件  
%%  
[0-9]+ {  
    printf("Find a Number: %s\n",yytext);  
    Total_Num++;  
}  
.  
\n {  
%%  
int main()  
{  
    yylex();  
    printf("Total_Num = %d\n",Total_Num);  
    return 0;  
}
```

Installation(1)

- MSYS2
 - <https://www.msys2.org/>

Installation

1. Download the installer

`msys2-x86_64-20231026.exe`

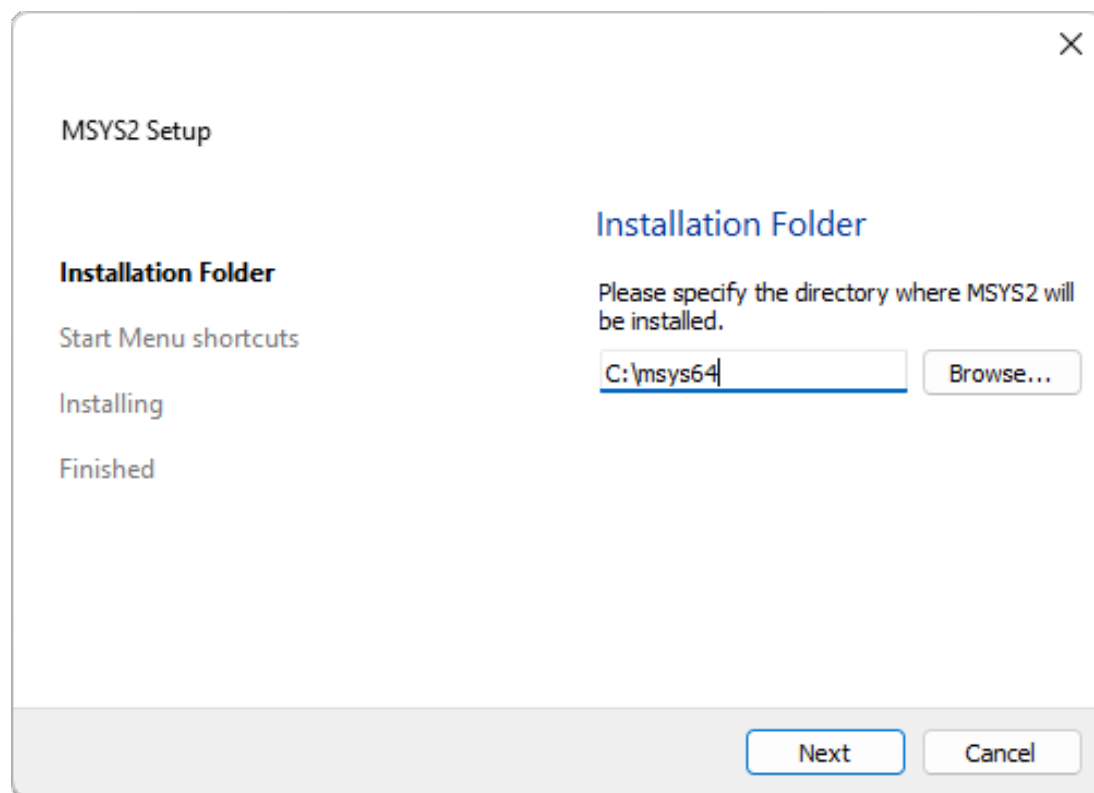
For more information on the installer, like command line options, or how to verify the checksum and signature of the installer, see the [installer guide](#).

2. Run the installer. MSYS2 requires 64 bit Windows 8.1 or newer.

3. Enter your desired **Installation Folder** (short ASCII-only path on a NTFS volume, no accents, no spaces, no symlinks, no subst or network drives, no FAT).

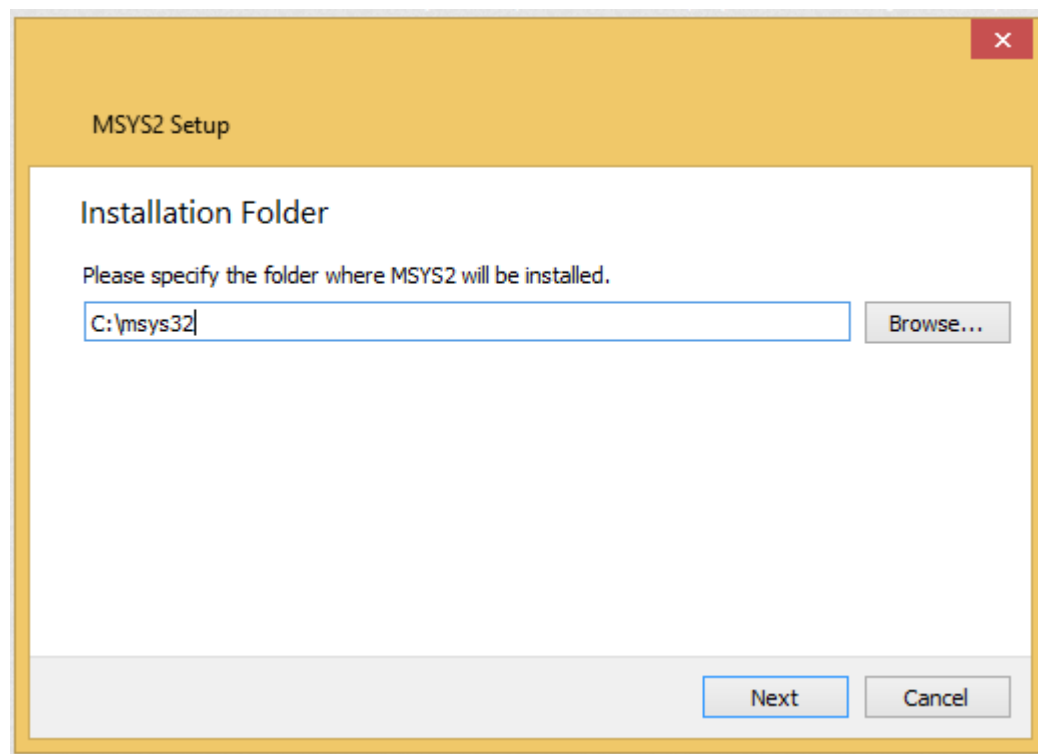
Installation(2)

- 選擇下一步



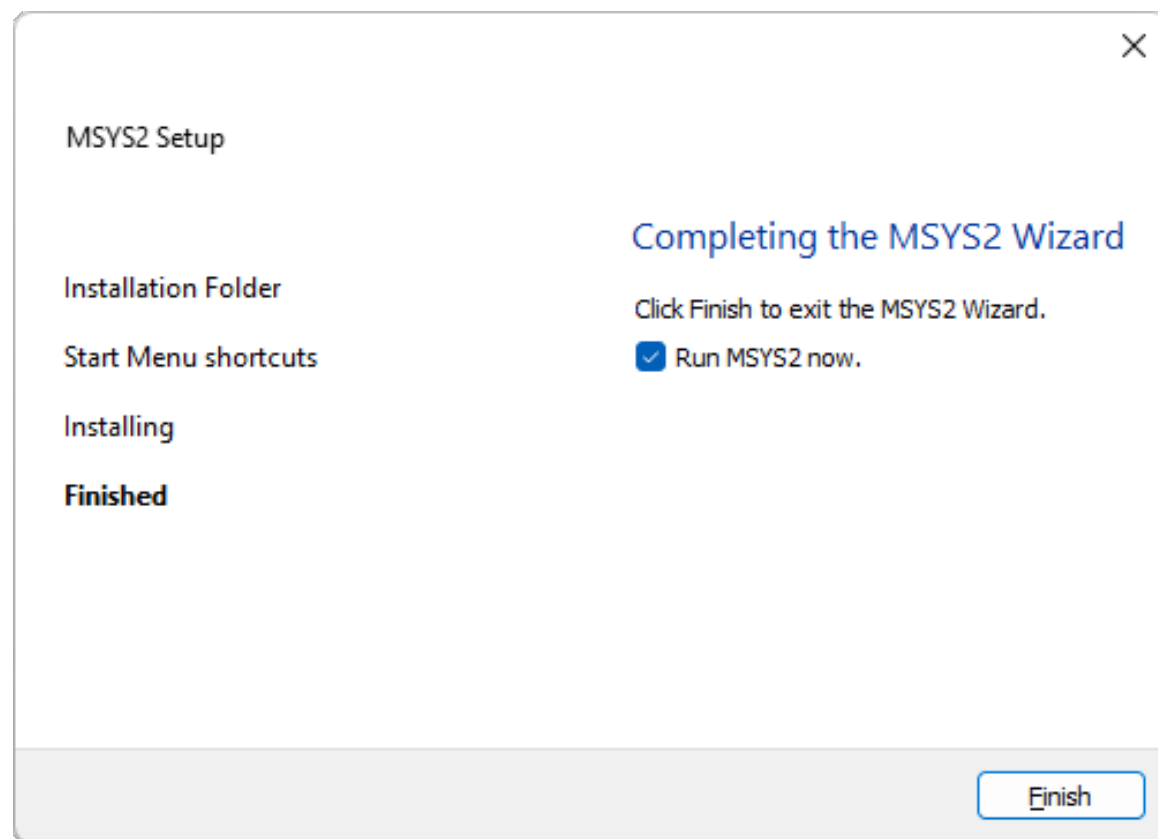
Installation(3)

- 選擇安裝的路徑



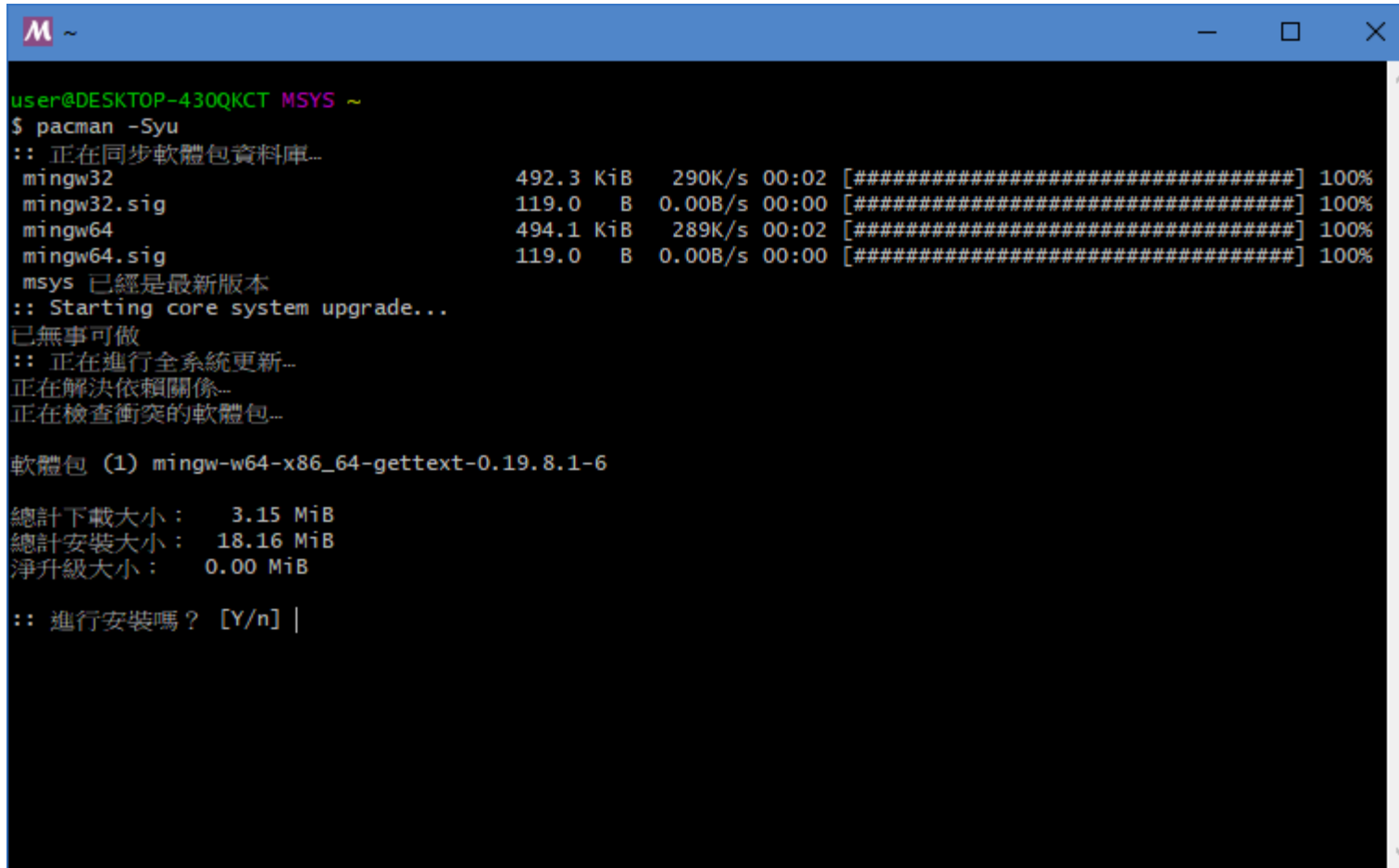
Installation(3)

- 安裝完成後，便可以開啟MSYS。



Installation(4)

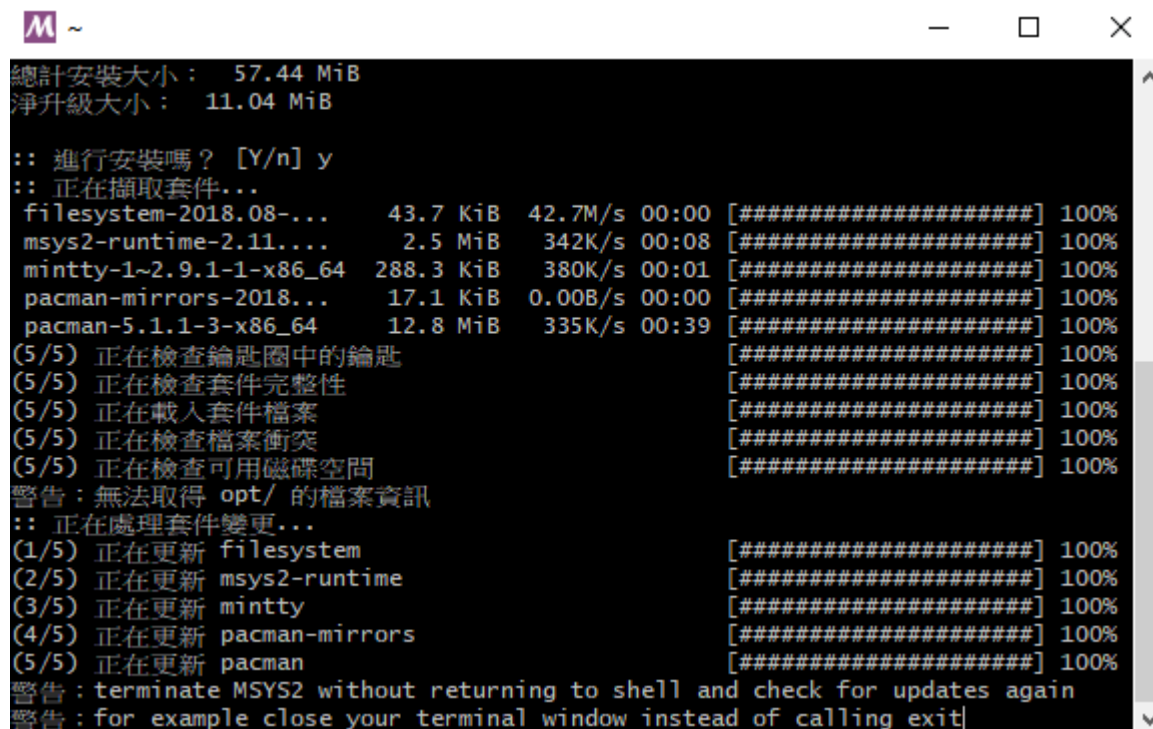
- 接著需要進行更新，請輸入 `$ pacman -Syu`



```
user@DESKTOP-430QKCT MSYS ~  
$ pacman -Syu  
:: 正在同步軟體包資料庫...  
mingw32                492.3 KiB   290K/s  00:02 [#####] 100%  
mingw32.sig            119.0 B    0.00B/s  00:00 [#####] 100%  
mingw64                494.1 KiB   289K/s  00:02 [#####] 100%  
mingw64.sig            119.0 B    0.00B/s  00:00 [#####] 100%  
msys 已經是最新版本  
:: Starting core system upgrade...  
已無事可做  
:: 正在進行全系統更新...  
正在解決依賴關係...  
正在檢查衝突的軟體包...  
  
軟體包 (1) mingw-w64-x86_64-gettext-0.19.8.1-6  
  
總計下載大小：    3.15 MiB  
總計安裝大小：   18.16 MiB  
淨升級大小：     0.00 MiB  
  
:: 進行安裝嗎？ [Y/n] |
```


Installation(5)

- 看到以下畫面，請重新開啟MSYS



```
M ~
總計安裝大小： 57.44 MiB
淨升級大小： 11.04 MiB

:: 進行安裝嗎？ [Y/n] y
:: 正在擷取套件...
filesystem-2018.08-... 43.7 KiB 42.7M/s 00:00 [#####] 100%
msys2-runtime-2.11.... 2.5 MiB 342K/s 00:08 [#####] 100%
mintty-1~2.9.1-1-x86_64 288.3 KiB 380K/s 00:01 [#####] 100%
pacman-mirrors-2018... 17.1 KiB 0.00B/s 00:00 [#####] 100%
pacman-5.1.1-3-x86_64 12.8 MiB 335K/s 00:39 [#####] 100%
(5/5) 正在檢查鑰匙圈中的鑰匙 [#####] 100%
(5/5) 正在檢查套件完整性 [#####] 100%
(5/5) 正在載入套件檔案 [#####] 100%
(5/5) 正在檢查檔案衝突 [#####] 100%
(5/5) 正在檢查可用磁碟空間 [#####] 100%
警告：無法取得 opt/ 的檔案資訊
:: 正在處理套件變更...
(1/5) 正在更新 filesystem [#####] 100%
(2/5) 正在更新 msys2-runtime [#####] 100%
(3/5) 正在更新 mintty [#####] 100%
(4/5) 正在更新 pacman-mirrors [#####] 100%
(5/5) 正在更新 pacman [#####] 100%
警告：terminate MSYS2 without returning to shell and check for updates again
警告：for example close your terminal window instead of calling exit
```

Installation(6)

- 當開啟完成時，請輸入 `$ pacman -Su`

```
hugh@DESKTOP-QE760B3 MSYS ~  
$ pacman -Su  
:: Starting core system upgrade...  
已無事可做  
:: 正在進行全系統更新--  
正在解決依賴關係--  
正在檢查衝突的軟體包--  
  
軟體包 (62) bash-completion-2.8-2 brotli-1.0.6-1 bsdcpio-3.3.3-3  
bsdtar-3.3.3-3 coreutils-8.30-1 curl-7.61.0-2 dash-0.5.10.2-1  
dtc-1.4.7-1 file-5.34-1 gawk-4.2.1-1 gcc-libs-7.3.0-3  
glib2-2.54.3-1 gnupg-2.2.9-1 grep-3.1-1 heimdal-libs-7.5.0-2  
icu-62.1-1 info-6.5-2 less-530-1 libarchive-3.3.3-3  
libargp-20110921-2 libassuan-2.5.1-1 libcrypto-2.1-2  
libcurl-7.61.0-2 libexpat-2.2.6-1 libffi-3.2.1-3  
libgcrypt-1.8.3-2 libgnutls-3.6.3-1 libgpg-error-1.32-1  
libgpgme-1.11.1-1 libhogweed-3.4-1 libidn2-2.0.5-1  
libksba-1.3.5-1 liblz4-1.8.2-1 liblzma-5.2.4-1 liblz02-2.10-2  
libnettle-3.4-1 libnpt-1.6-1 libopenssl-1.0.2.p-2  
libpcre-8.42-1 libpcre16-8.42-1 libpcre32-8.42-1  
libpcrecpp-8.42-1 libpcreposix-8.42-1 libpsl-0.20.2-1  
libreadline-7.0.005-1 libsqlite-3.21.0-4 libunistring-0.9.10-1  
libutil-linux-2.32.1-1 libxml2-2.9.8-1 m4-1.4.18-2  
ncurses-6.1.20180908-1 nettle-3.4-1 openssl-1.0.2.p-2  
pcre-8.42-1 pinentry-1.1.0-2 rebase-4.4.4-1 sed-4.5-1  
time-1.9-1 ttyrec-1.0.8-2 util-linux-2.32.1-1 wget-1.19.5-1  
xz-5.2.4-1  
  
總計下載大小： 34.08 MiB  
總計安裝大小： 153.95 MiB  
淨升級大小： 12.42 MiB  
  
:: 進行安裝嗎？ [Y/n]
```

- 這裡請輸入y

Installation(7)

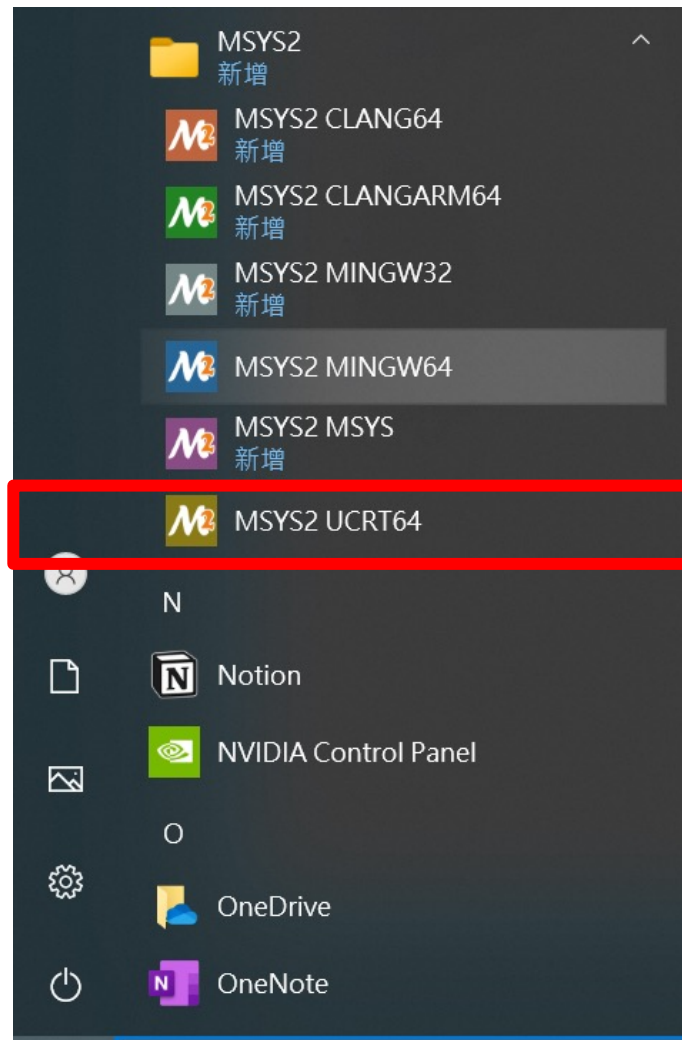
- 安裝GCC 輸入
- \$ pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain

```
$ pacman -S git mingw-w64-x86_64-gcc base-devel
:: 在群組 base-devel 中有 56 個成員：
:: 軟體庫 msys
 1) asciidoc  2) autoconf  3) autoconf2.13  4) autogen  5) automake-wrapper  6
15) automake1.8 16) automake1.9 17) bison 18) diffstat 19) diffutils 20)
33) libtool 34) libunrar 35) libunrar-devel 36) m4 37) make 38) man-db
52) texinfo-tex 53) ttyprec 54) unrar 55) wget 56) xmlto
輸入某個選擇（預設=全部選定）：
```

- Enter


Installation(8)

- 開啟MSYS2 UCRT64



Installation(9)

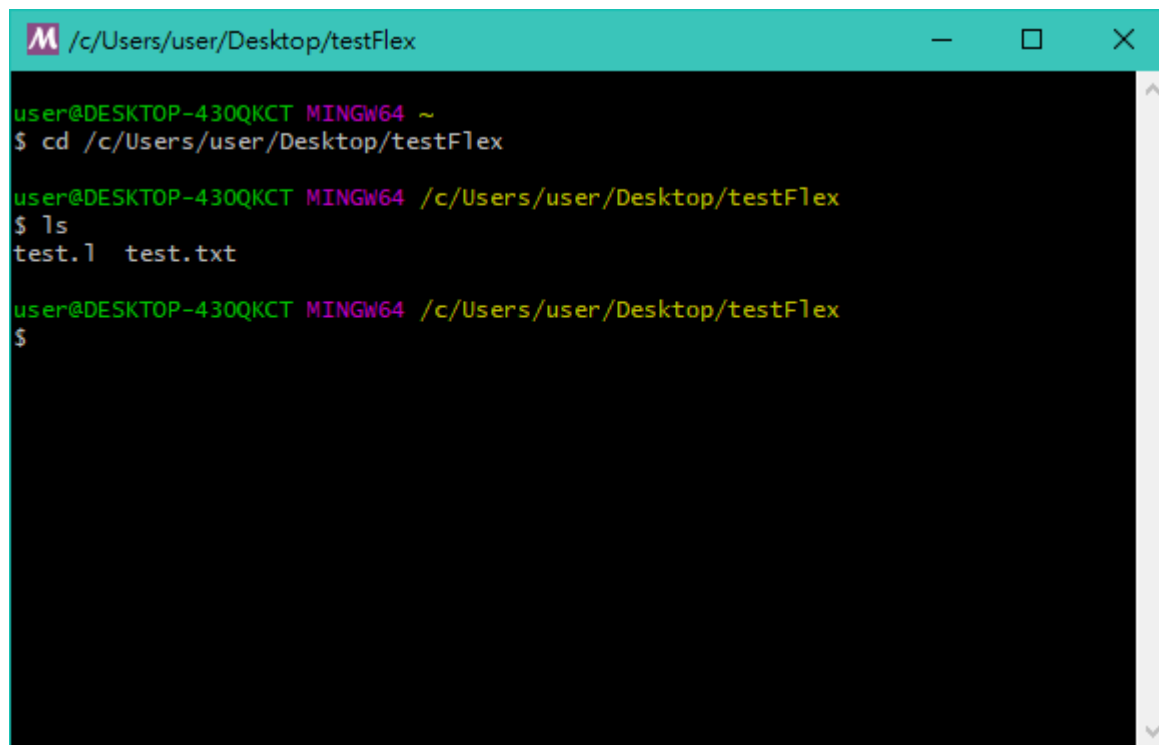
- 可以透過輸入以下指令檢查是否安裝成功
- \$ gcc --version
- \$ flex --version



```
hugh@DESKTOP-QE760B3 MINGW64 ~  
$ gcc --version  
gcc.exe (Rev3, Built by MSYS2 project) 8.2.0  
Copyright (C) 2018 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  
  
hugh@DESKTOP-QE760B3 MINGW64 ~  
$ flex --version  
flex 2.6.4  
  
hugh@DESKTOP-QE760B3 MINGW64 ~  
$ yacc --version  
bison (GNU Bison) 3.1  
由 Robert Corbett 與 Richard Stallman 編寫。  
  
版權所有 (C) 2018 自由軟體基金會。  
這是自由軟體；請參考原始碼的版權聲明。本軟體不提供任何保證，甚至不會包括  
可售性或適用於任何特定目的的保證。  
  
hugh@DESKTOP-QE760B3 MINGW64 ~  
$
```

實作方式

- 我們可以利用vi指令撰寫xxx.l檔，也可以用Windows上的記事本或VS code撰寫後，再放入自己創建資料夾中。



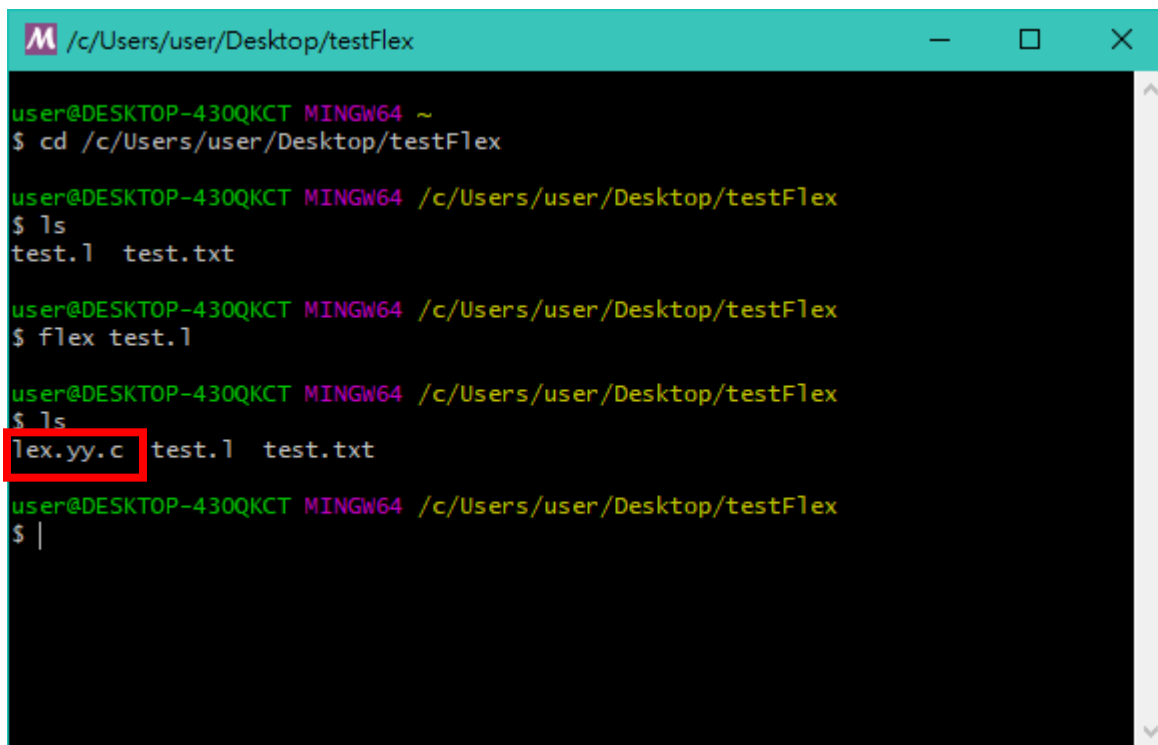
```
M /c/Users/user/Desktop/testFlex
user@DESKTOP-430QKCT MINGW64 ~
$ cd /c/Users/user/Desktop/testFlex

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ ls
test.l  test.txt

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$
```

實作方式

- 在我們撰寫好test.l檔的規則後，我們可以利用flex命令使得test.l檔轉換為lex.yy.c檔。
 - flex test.l



```
M /c/Users/user/Desktop/testFlex
user@DESKTOP-430QKCT MINGW64 ~
$ cd /c/Users/user/Desktop/testFlex

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ ls
test.l  test.txt

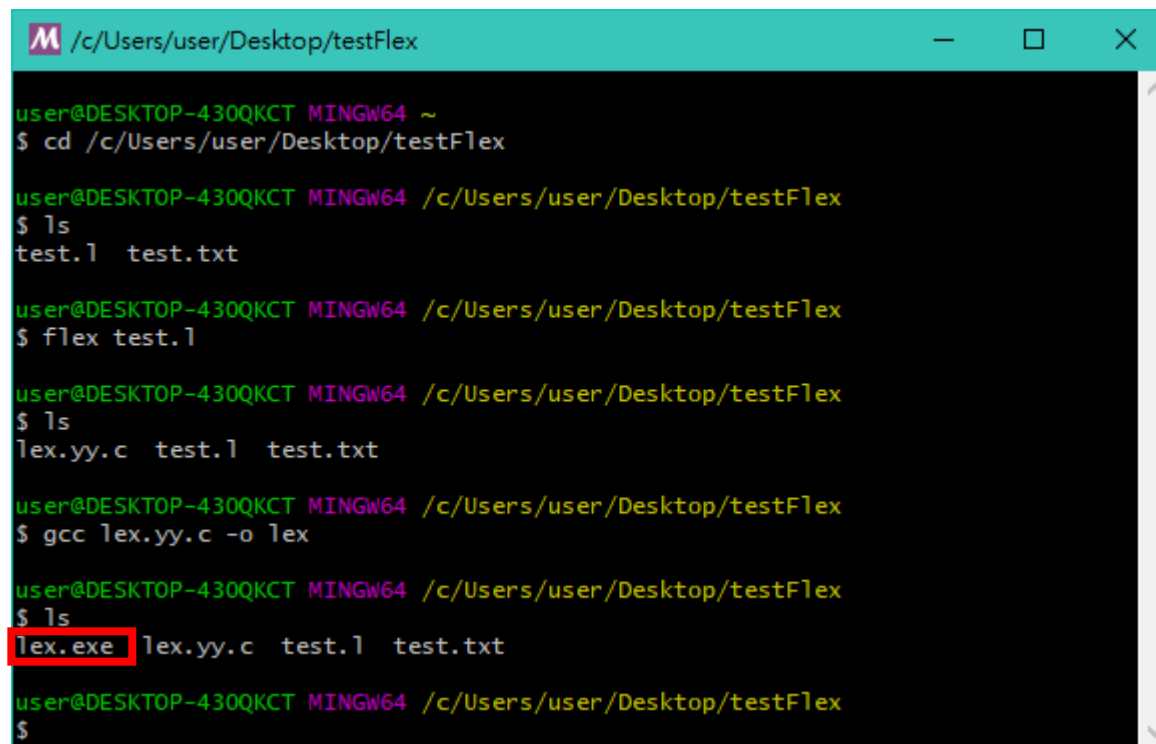
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ flex test.l

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ ls
lex.yy.c  test.l  test.txt

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ |
```

實作方式

- 產生了lex.yy.c檔後，我們便可以將它編譯成執行檔。
 - gcc lex.yy.c -o lex



```
M /c/Users/user/Desktop/testFlex
user@DESKTOP-430QKCT MINGW64 ~
$ cd /c/Users/user/Desktop/testFlex

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ ls
test.1 test.txt

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ flex test.1

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ ls
lex.yy.c test.1 test.txt

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ gcc lex.yy.c -o lex

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ ls
lex.exe lex.yy.c test.1 test.txt

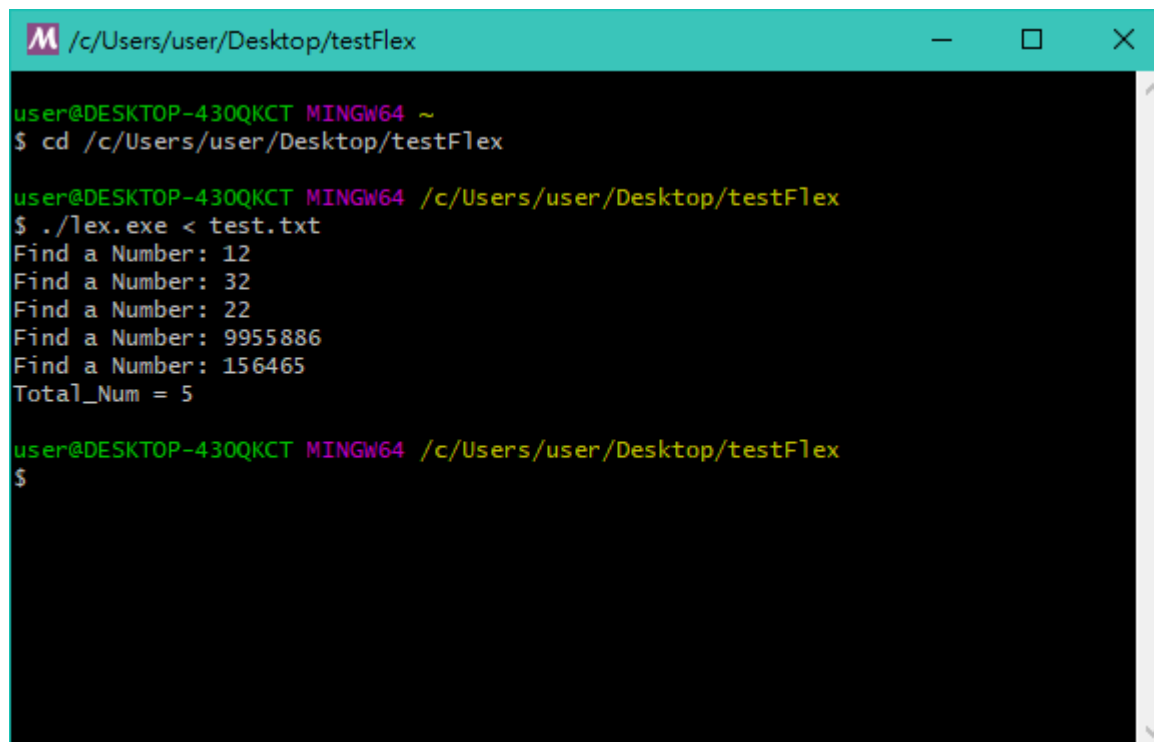
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$
```


實作方式

- 建立一測試檔test.txt：

```
12+32+22  
9955886  
156465
```

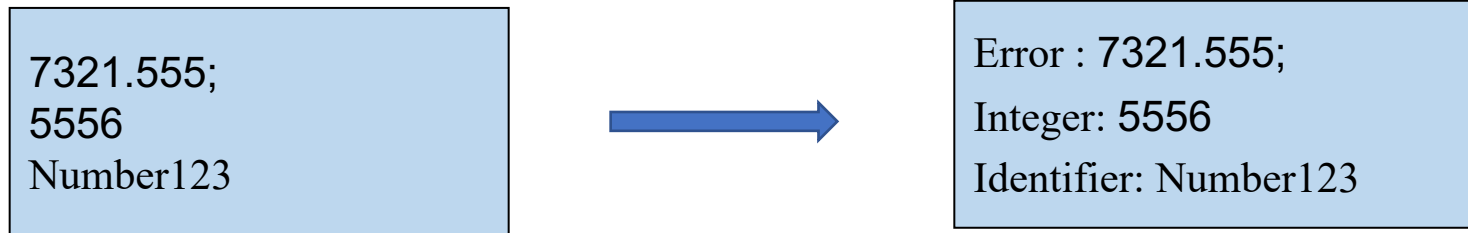
- 輸入./lex.exe < test.txt分析：



```
M /c/Users/user/Desktop/testFlex  
user@DESKTOP-430QKCT MINGW64 ~  
$ cd /c/Users/user/Desktop/testFlex  
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex  
$ ./lex.exe < test.txt  
Find a Number: 12  
Find a Number: 32  
Find a Number: 22  
Find a Number: 9955886  
Find a Number: 156465  
Total_Num = 5  
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex  
$
```

Homework

- 撰寫一個lex檔，並完成下列功能：
 - 輸入為input.txt檔，內有浮點數(float)、整數(int)、變數(id)，**並用換行隔開**，例如：



※批改時會使用較難的範例，請自己完整測試

- 利用lex辨別出input.txt中浮點數、整數、變數，每次辨識完後輸出float、int、id或error。**並在最後輸出每個型態的數量統計**

Homework

- 變數(id) : 第一個字必須為字母，後面可以接字母、底線以及數字，變數長度不限。
- 整數(int) : 正、負整數，且包含0。
- 浮點數(float) : 至少一位整數部分與至少一位小數部分（正、負數皆可）。
- 一行一個token（以換行隔開），並且判斷這一行的token是哪一種型態，所以以上token若是包含 分號 或者 空格 都會判定為error。

Homework

輸出的內容需要包含：

1.型態：內容

Ex: Integer : 23456

Float : 12.2334

Identifier(ID) : A23_abc

Error : 74;

2.各個型態的數量有幾個

Ex: Total_ Integer : 6

Total_ Float : 4

Total_ Identifier(ID) : 3

Total_ Error : 10

Homework

- 繳交項目：報告、程式碼（.l檔）
- 報告格式
 - 1.封面：作業題目、學號、姓名
 - 2.作法：規則建立說明
 - 3.程式碼
 - 4.執行結果：測試資料及結果
 - 5.討論&心得：遭遇困難及解決過程
 - 繳交報告：上傳至ilearn，繳交pdf檔
11/3(日) 晚上 23:55之前
- 有任何問題請用E-mail或ilearn聯絡：
 - 莊惠淇：M1310081@o365.fcu.edu.tw
 - 劉秉霖：M1305865@o365.fcu.edu.tw

延伸作業

- 試著用C/C++撰寫scanner程式，
 - 輸入及輸出如作業相同，輸入檔案後，透過scanner可以判斷是數字、浮點數、變數或者錯誤，並印出各Token有幾個。
- 在報告中額外說明，與作業一起繳交，便可額外加分(3分)。