**Buổi 3: Lập trình C trên Linux**

* Bộ biên dịch GCC
  + *Kiểm tra trình biên dịch GCC*

Mở Terminal và gõ gcc, nếu thấy thông báo như sau

“ gcc: fatal error: no input files” thì GCC đã được cài đặt rồi. Nếu nhận được thông báo lỗi như sau “No command ‘gcc’ found” thì GCC chưa được cài đặt.

Để cài đặt GCC thì từ terminal gõ:

$ sudo apt-get install build-essential

* + *Hướng dẫn biên dịch*

Để dịch code của C thì dùng trình biên dịch gcc, còn code C++ thì dùng trình biên dịch g++

Các tùy chọn quan trọng của gcc và g++:

-o: sinh ra tập tin output

-c: sinh ra tập tin đối tượng

-S: sinh ra tập tin assembly

-I: đặc tả thư mục chứa tập tin include

-l: đặc tả thư viện (.lib hoặc .a)

-L: đặc tả đường dẫn đến thư viện

Ví dụ biên dịch file code helloworld.c

$ gcc –S helloworld.c (sinh ra file helloworld.s: file assembly)

$ gcc –c helloworld.c (sinh ra file helloworld.o: file object)

$ gcc –o helloworld helloworld.c (sinh ra file thực thi helloworld)

* Tạo Makefile
  + *Cú pháp của Makefile*

<target>: <danh sách các file phụ thuộc>

<TAB> lệnh

Chú ý phải dùng phím <TAB> chứ không phải khoảng trắng

Ví dụ: tạo Makefile cho hai file main.c và func.c có nội dụng lần lượt như sau

File main.c

#include <stdio.h>

main(){

fun();

}

File func.c

void fun(){

printf ("Hello World! \n");

}

Makefile

# Makefile cho HelloWord

CC=gcc

all: main

main: main.o fun.o

$(CC) -o main main.o fun.o

main.o: main.c

$(CC) -c main.c

fun.o: fun.c

$(CC) -c fun.c

clean:

rm -f \*.o main

* + *Biên dịch với Makefile*

Sau khi đã tạo Makefile thì chúng ta có thể biên dịch bằng lệnh make như sau

$ make

Để xóa các file object và file thực thi để biên dịch lại

$ make clean

* + *Thực hành:*

Cho các file code trong thư mục bai3 (main.c, fun1.c, fun2.c, fun3.c, fun1.h, fun2.h, fun3.h). Tạo Makefile để biên dịch các file code với nội dụng như sau:

CC = gcc

all: main

main: main.o fun1.o fun2.o fun3.o

$(CC) -o main main.o fun1.o fun2.o fun3.o

main.o: main.c

$(CC) -c main.c

fun1.o: fun1.c

$(CC) -c fun1.c

fun2.o: fun2.c

$(CC) -c fun2.c

fun3.o: fun1.c

$(CC) -c fun3.c

clean:

rm -f \*.o main

Vẫn là các file code như ở bai3 trong thư mục bai4 nhưng yêu cầu tạo Makefile với các biến và các ký hiệu đại diện nên Makefile cho folder bai4 có nội dung như sau:

CFLAGS = -g -Wall

CC = gcc

OBJ = fun1.o fun2.o fun3.o

SRCS = fun1.c fun2.c fun3.c main.c

INC = fun1.h fun2.h fun3.h

all: main

# The variable $@ has the value of the target.

# In this case $@ = psort

main: main.o ${OBJ}

${CC} ${CFLAGS} ${INC} -o $@ main.o ${OBJ}

.c.o:

${CC} ${CFLAGS} ${INC} -c $<

depend:

makedepend ${SRCS}

clean:

rm -f \*.o main \*~

Cho các file code trong thư mục bai5 (main.c, fun1.c, fun2.c, fun3.c, fun1.h, fun2.h, fun3.h). Nhưng các file code đã được bổ sung dòng “#include <zlib.h>”, một dạng như thư viện nên sau khi bổ sung chúng ta có thể gọi các hàm của thư viện này. Do đó chúng ta phải bổ sung đường dẫn chỉ đến file zlib.h và file thư viện .so (zlib.so) vào Makefile như sau:

CC = gcc

OBJ = fun1.o fun2.o fun3.o

SRCS = fun1.c fun2.c fun3.c main.c

INC = -I/home/user/apps/zlib/include

LIB = -lz -L/home/user/apps/zlib/lib

all: main

# The variable $@ has the value of the target.

# In this case $@ = psort

main: main.o ${OBJ}

${CC} ${LIB} ${INC} -o $@ main.o ${OBJ} ${LIB}

.c.o:

${CC} ${LIB} ${INC} -c $<

depend:

makedepend ${SRCS}

clean:

rm -f \*.o main \*~