



Khoa Công Nghệ Thông Tin Trường Đại Học Cần Thơ

**Phát triển ứng dụng : biên dịch với
GNU gcc, make**



Đỗ Thanh Nghị
dtnghi@cit.ctu.edu.vn

Cần Thơ
25-07-2012

Nội dung

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

Nội dung

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

Biên dịch

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

■ Ví dụ đơn giản trong ngôn ngữ C

- Soạn thảo 2 files: **main.c** và **func.c**

```
/*** func.c ***/  
#include <stdio.h>  
void hi() {  
    printf("Hi there!!\n");  
}
```

```
/*** main.c ***/  
main() {  
    hi();  
}
```

Biên dịch

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

■ Sử dụng `gcc [options] sources`

các tùy chọn (options)

- -o: sinh ra **tập tin output**
- -c: sinh ra **tập tin đối tượng .o**
- -I: đặc tả thư mục chứa tập tin **include**
- -l: đặc tả tên thư viện
- -L: đặc tả đường dẫn đến thư viện

Biên dịch

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

-
- Sử dụng **gcc** để dịch ra file đối tượng (.o)
 - Gõ: **gcc -c main.c** (sinh ra **main.o**)
 - Gõ: **gcc -c func.c** (sinh ra **func.o**)
 - Sử dụng **gcc** để dịch ra file thực thi từ các file đối tượng
 - Gõ: **gcc -o main main.o func.o** (sinh ra ch trình **main**)

Nội dung

- Sử dụng gcc
- **Sử dụng make và Makefile**
- Thư viện liên kết tĩnh
- Thư viện liên kết động

Biên dịch

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

■ Dịch với **make** và **Makefile**

- Soạn thảo **Makefile** (nếu đặt tên khác thì khi sử dụng **make** phải dùng: **make -f tenkhac**)
- **Makefile** là 1 file đặc biệt dùng để quản lý các tập tin trong dự án
- Chứa các quy tắc biên dịch
- Sử dụng thuộc tính “modified time” để biên dịch lại
- Tạo ra 1 đồ thị phụ thuộc
- Cú pháp:

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

<TAB> lệnh

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

Biên dịch

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

■ Dịch với **make** và **Makefile**

- **Makefile** có thể sử dụng biến, ví dụ:

```
OBJ = foo.o main.o
```

```
main: $(OBJ)
```

```
gcc -o main $(OBJ)
```

- Một số biến đặc biệt:

`$(@)`: target

`$(+)`: danh sách các files phụ thuộc

`$(<)`: file đầu tiên trong danh sách

Biên dịch

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

■ Dịch với **make** và **Makefile** để dịch tự động các bước trên

- Soạn thảo **Makefile** như sau

```
CC      =      gcc
all:    main
main:   main.o  func.o
        $(CC)  -o  main  main.o  func.o
main.o: main.c
        $(CC)  -c  main.c
func.o: func.c
        $(CC)  -c  func.c
```

- Gõ lệnh **make** sẽ đọc các bước dịch trong **Makefile** để dịch và sinh ra chương trình **main**

Nội dung

- Sử dụng gcc
- Sử dụng make và Makefile
- **Thư viện liên kết tĩnh**
- Thư viện liên kết động

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

Biên dịch

■ Thư viện liên kết tĩnh

- Mặc định **gcc** có thể liên kết với các file đối tượng
- Gõ: **gcc -o main main.c func.o**
- **Thư viện tĩnh**: chứa các file đối tượng **.o** được tạo bởi công cụ **ar**
- Gõ: **ar rcs libfunc.a func.o** (tạo ra thư viện **libfunc.a**)
- Dịch main với: **gcc -o main main.c libfunc.a**

Nội dung

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- **Thư viện liên kết động**

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

Biên dịch

■ Thư viện liên kết động

- Mặc định `gcc` có thể liên kết với các file đối tượng
- Gõ: `gcc -o main main.c func.o`
- **Thư viện động**: được load khi chtrình chạy, được tạo bởi `gcc`
- Gõ: `gcc -c -fPIC func.c` (tạo ra `func.o`)
- Gõ: `gcc -shared -o libfunc.so.1.0 func.o`
`gcc -shared -Wl,-soname,libfunc.so.1.0 -o libfunc.so.1.0 func.o`
- Tạo ra `libfunc.so.1.0`
- Dịch main với: `gcc -o main main.c libfunc.so.1.0`

- Sử dụng gcc
- Sử dụng make và Makefile
- Thư viện liên kết tĩnh
- Thư viện liên kết động

Biên dịch

■ Thư viện liên kết động

- Chạy chương trình **main**: có lỗi không tìm thấy thư viện liên kết động
- Hoặc chỉ lại đường dẫn đến thư viện
- gõ: **export LD_LIBRARY_PATH=\$(pwd)**
- Hoặc sao chép **libfunc.so.1.0** đến **/usr/lib** rồi
- chạy **ldconfig** để cập nhật LD

```
gcc -shared -soname,libfoo.so.1 -o libfoo.so.1.0
```



```
{ ldconfig → /etc/ld.so.cache
```

```
{ /etc/ld.so.conf
```

```
ld.so → the linux run-time linker
```

Biên dịch

- Các công cụ khác
 - `g++`: biên dịch C++
 - `automake`
 - `autoconf`



Cám ơn !