

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN
-----o0o-----

Tạ Tuấn Anh

Bài giảng điện tử môn học

NGÔN NGỮ LẬP TRÌNH C

Tóm tắt nội dung môn học

C là một ngôn ngữ lập trình cấu trúc bậc cao được các nhà lập trình chuyên nghiệp sử dụng phổ biến để phát triển các phần mềm hệ thống (hệ điều hành, chương trình dịch, cơ sở dữ liệu, ...). Lý do ngôn ngữ C được ưu chuộng chính là tính mềm dẻo và ngắn gọn của nó. Một chương trình được viết ở ngôn ngữ C có tính khả chuyển cao. Nó có thể được dịch và chạy trong nhiều loại máy tính (PC, Sun, Mainframe,...) cũng như trên nhiều nền hệ điều hành (DOS, UNIX,...). Ngoài ra C cho phép viết chương trình bám sát cách tổ chức bộ nhớ chương trình khi chạy. Do vậy một chương trình được dịch từ C luôn có kích thước nhỏ gọn hơn một chương trình cùng loại được dịch từ các ngôn ngữ bậc cao khác như PASCAL. Nhưng cũng chính vì lý do này mà việc nắm bắt và thành thạo ngôn ngữ C sẽ khó khăn hơn nhiều so với ngôn ngữ khác. Môn học này giới thiệu cho các học viên các kiến thức căn bản cũng như nâng cao về ngôn ngữ lập trình C. Bên cạnh các kiến thức về cú pháp cũng như kỹ năng viết chương trình C, học viên còn nắm bắt được các vấn đề liên quan đến tổ chức bộ nhớ của một chương trình.

Kiến thức yêu cầu

Để tiếp thu tốt kiến thức môn học này, yêu cầu học viên trước khi học đã tìm hiểu các khái niệm cơ bản trong Tin học, có kỹ năng căn bản viết một chương trình có cấu trúc bằng một ngôn ngữ bậc cao như PASCAL. Ngoài ra một số kiến thức về cấu trúc dữ liệu và giải thuật (danh sách móc nối, cây tìm kiếm, ...) có thể giúp học viên sử dụng C để viết các chương trình ứng dụng.

Tổng thời lượng: 45 tiết

MỤC LỤC

TÀI LIỆU THAM KHẢO

Nhập môn Lập trình Ngôn ngữ C

Trần Việt Linh, Lê Đăng Hưng, Lê Đức Trung, Nguyễn Thanh Thủy

Nhà Xuất bản Khoa học Kỹ thuật, 2000

Ngôn ngữ Lập trình C

Quách Tuấn Ngọc

Nhà Xuất bản Giáo dục, 1998

Language C

Brian Kernighan, Denis Ritchie

Prentice Hall, 1988

Programmer en langage C

Claude Delannoy

Eyrolles, 1998

CHƯƠNG 1 - NHẬP MÔN LẬP TRÌNH C

Mục đích của chương này là giới thiệu tổng quan về ngôn ngữ C bao gồm các kiến thức về lịch sử, đặc điểm và vai trò của nó. Học viên được làm quen với các chương trình viết bằng C cũng như cách dịch chúng để chạy.

Yêu cầu: Có một phiên bản cài đặt của trình biên dịch Turbo C hay một trình biên dịch khác để chạy thử chương trình.

Thời lượng: 5 tiết

Mục 1.1 - Tổng quan về ngôn ngữ C

Mục này cho phép học viên làm quen với một chương trình viết dưới ngôn ngữ C và tìm hiểu lịch sử của nó. Các thành phần cơ bản của một chương trình C được giới thiệu để học viên có một cái nhìn tổng quan về ngôn ngữ lập trình này.

Yêu cầu: Đã có khái niệm về lập trình và ngôn ngữ lập trình.

Thời lượng: 3 tiết

Bài 1 - Lịch sử hình thành và phát triển

Tóm tắt nội dung:

Ngôn ngữ lập trình C ra đời vào đầu thập kỉ 70 với mục đích dùng để viết hệ điều hành UNIX. C được phát triển rất mạnh sau đó và được chuẩn hoá với tên gọi ANSI C. Ngôn ngữ này được các nhà lập trình chuyên nghiệp rất ưa chuộng để phát triển các phần mềm hệ thống. Một mở rộng của C là C++ ra đời vào đầu thập kỉ 80. Nó là một ngôn ngữ lập trình hướng đối tượng được phát triển trên nền của C.

Thời lượng: 1 tiết

Ngôn ngữ C do Brian W.Kernighan và Denis M. Ritchie phát triển vào đầu những năm 70 tại phòng thí nghiệm BELL (Hoa Kỳ) với mục đích ban đầu để phát triển hệ điều hành UNIX. Bối cảnh ra đời xuất phát từ nhu cầu cần phải có một ngôn ngữ lập trình hệ thống thay thế cho hợp ngữ (ASSEMBLY) rất nặng nề trong lập trình. Hơn nữa một chương trình viết bằng hợp ngữ không có tính khả chuyển vì chúng gắn chặt với bộ lệnh của vi xử lí.

Tiền thân của C phải kể đến các ngôn ngữ BCPL do Martin Richard nghiên cứu. Tiếp đến là ngôn ngữ B do Ken Thompson xây dựng năm 1970 dùng để viết hệ điều hành UNIX cho dòng máy tính PDP-7. C là ngôn ngữ được kế thừa từ B và hoàn thiện để có được các tính năng mạnh của một ngôn ngữ lập trình hệ thống có khả năng ứng dụng rộng rãi như ngày nay. Đó là các tính năng:

- *Lập trình bậc cao*: Giống như PASCAL, chương trình C sử dụng tập các câu lệnh điều khiển như rẽ nhánh, lặp ở mức độ trừu tượng của lưu đồ giải thuật. Điều này cho phép viết các giải thuật bằng ngôn ngữ C khá dễ dàng.
- *Lập trình cấu trúc*: Một chương trình C có thể được phân chia, cấu trúc thành các modul nhỏ. Điều này giúp phát triển chương trình một cách hệ thống hơn và dễ bảo trì.
- *Lập trình hệ thống*: Không giống như PASCAL, ngôn ngữ C không dùng nhiều kiểu dữ liệu trừu tượng. C cho phép các thao tác với bộ nhớ chương trình rất uyển chuyển. Một người lập trình trên ngôn ngữ C có thể tự do tổ chức và lưu trữ dữ liệu trên bộ nhớ theo ý mình. Tính năng này là vô cùng quan trọng khi cần phát triển các chương trình hệ thống liên quan nhiều đến bộ nhớ máy tính. Ngoài ra ngôn ngữ C hỗ trợ phần lớn các phép xử lý mà một hợp ngữ có thể làm.
- *Tính khả chuyển*: Một chương trình viết trên ngôn ngữ C có thể được dịch ra mã chương trình trên nhiều dòng máy và hệ điều hành khác nhau bởi khả năng tương thích của các câu lệnh trong chương trình. Ngoài ra C còn có bộ tiền xử lý tạo ra khả năng biên dịch theo điều kiện để việc dịch chương trình thích ứng cho từng hệ thống khác nhau.
- *Tính nhỏ gọn*: Một chương trình viết trên C sau khi dịch có độ tối ưu về mã lệnh hơn bất cứ một ngôn ngữ bậc cao nào khác. Chính vì vậy các chương trình được dịch từ ngôn ngữ C thường có kích thước nhỏ gọn.

Với tất cả các tính năng trên, ngôn ngữ C là một ngôn ngữ cực kì hiệu quả và có sức diễn cảm trong lập trình. Nó đã trở thành ngôn ngữ lập trình mà các nhà lập trình chuyên nghiệp ưa chuộng trong nhiều lĩnh vực. Trong lĩnh vực lập trình hệ thống, có tới 90% chương trình được viết bằng ngôn ngữ C. Ngoài ra nó còn được dùng để viết chương trình trong các lĩnh vực hiện đại khác của Tin học về xử lý tín hiệu, số liệu, văn bản, ...

Ngôn ngữ C đã được viện tiêu chuẩn quốc gia Mỹ (ANSI) chuẩn hoá và công bố vào năm 1988 với tên gọi là ANSI C. Bên cạnh ngôn ngữ C người lập trình còn biết đến một ngôn ngữ lập trình có tên tương tự là C++. Đây là một ngôn ngữ lập trình được mở rộng từ C để thêm khả năng lập trình hướng đối tượng. Vì C++ là ngôn ngữ bao trùm lên C nên để học tốt C++ yêu cầu người lập trình trước hết phải nắm vững C.

Bài 2 - Bắt đầu lập trình C

Tóm tắt nội dung:

Học viên sẽ được làm quen với hai chương trình đơn giản. Chúng minh họa các thành phần cơ bản có trong một chương trình C.

Thời lượng: 1 tiết

1. Chương trình Hello!

Cũng như với nhiều ngôn ngữ lập trình khác, chúng ta bắt đầu tìm hiểu ngôn ngữ qua một chương trình đơn giản chỉ in ra một thông báo "Hello!" cho người sử dụng. Chương trình này được viết trong ngôn ngữ C như dưới đây.

```
#include <stdio.h>
void main()
```

```
{
    printf("Hello!");
}
```

Dòng đầu tiên của chương trình được gọi là một khai báo sử dụng tệp tiêu đề. Trong bất kỳ một chương trình nào ở ngôn ngữ C đều cần có những khai báo tệp tiêu đề bởi vì chúng cho phép gọi các hàm có sẵn trong thư viện hoặc được viết bởi một người lập trình khác. Trong chương trình này chúng ta đã khai báo sử dụng tệp tiêu đề <stdio.h> để có thể sử dụng các hàm vào ra dữ liệu chuẩn của hệ thống (printf(), scanf(),...). Phần tiếp theo của chương trình là một hàm chính chứa các câu lệnh được thực hiện khi chương trình chạy. Một chương trình C thì luôn có một hàm chính có tên là main. Dòng thông báo "Hello!" được in ra màn hình được thực hiện nhờ câu lệnh gọi hàm thư viện printf() trong chương trình chính.

2. Chương trình thứ hai

Để có thể thấy rõ hơn cấu trúc của một chương trình trong C, chúng ta tiếp tục tìm hiểu ngôn ngữ thông qua một ví dụ thứ hai. Ở ví dụ này chúng ta sẽ tạo một chương trình cho phép in diện tích của một hình tròn tương ứng với một bán kính được người sử dụng nhập vào.

```
#include <stdio.h>
/* khai báo một hằng số PI */
#define PI 3.14
/* khai báo một hàm tạo công thức tính diện tích */
float dientich(float r)
{
    return PI*r*r;
}
/* hàm chính của chương trình */
void main()
{
    float r;
    printf("Nhập bán kính r =");
    scanf("%f", &r);
    printf("Diện tích hình tròn là %f", dientich(r));
}
```

Chương trình này đã thể hiện một phần cấu trúc của một chương trình C sẽ được mô tả chi tiết trong bài 3. Dưới đây là thuyết minh cho chương trình:

- *Chú thích chương trình:* Các dòng văn bản được đặt nằm trong bộ dấu /* ... */ đều là chú thích cho một chương trình C.
- *Khai báo hằng số:* Dòng thứ hai trong chương trình có tác dụng khai báo một hằng số PI có giá trị là 3.14.
- *Hàm:* Một chương trình C có thể được cấu trúc bằng nhiều hàm. Trong chương trình trên, chúng ta đã khai báo một hàm để tính diện tích của một hình tròn với bán kính truyền vào như là một tham số thực (float). Kết quả của nó rõ ràng phải là một số thực. Hàm này được gọi trong hàm chính của chương trình để in giá trị diện tích hình tròn ra ngoài màn hình.
- *Hàm chính:* Một chương trình hoạt động thế nào được thể hiện bởi các câu lệnh trong hàm chính. Trong hàm chính của chương trình trên chúng ta đã khai báo một biến thực để chứa giá trị của bán kính hình tròn do người sử dụng nhập vào từ bàn phím. Lệnh gọi hàm scanf() cho phép chương trình lấy dữ liệu bán kính do người sử dụng nhập vào. Dòng lệnh

cuối cùng của hàm chính dùng để in giá trị diện tích của hình tròn có bán kính tương ứng ra màn hình. Giá trị diện tích được tính thông qua hàm dientich() đã được khai báo.

Bài 3 - Các thành phần của chương trình C

Tóm tắt nội dung:

Cũng như các ngôn ngữ lập trình khác, một chương trình C được viết từ các từ khoá, biến, câu lệnh, hàm,... Trong bài này chúng ta xem chi tiết các thành phần có trong một chương trình C để biết được ý nghĩa và vai trò của chúng.

Thời lượng: 1 tiết

1. Từ vựng

Một chương trình C được xây dựng trên một bộ từ vựng bao gồm tập kí tự có phân biệt chữ hoa và thường. Chính vì vậy khi viết chương trình C chúng ta không được phép sử dụng tùy tiện chữ hoa hay chữ thường. Ví dụ hàm chính của chương trình bắt buộc phải viết là main mà không được viết là MAIN hay Main. Trong một chương trình C chúng ta gặp hai loại bộ từ vựng quan trọng là từ khoá và tên của người sử dụng.

Từ khoá: Từ khoá là một bộ từ vựng được định nghĩa từ trước đối với một ngôn ngữ và chỉ để sử dụng vào các mục đích đã được xác định trong chương trình. Một số từ khoá hay dùng là các từ khoá kiểu dữ liệu (char, int, long, float, double,...), các lệnh điều khiển (if, switch, for, do, while,...), v.v.

Bảng các từ khoá trong ngôn ngữ C

asm	break	case	cdecl
char	const	continue	default
do	double	else	enum
extern	far	float	for
goto	huge	if	int
interrupt	long	near	pascal
register	return	short	signed
sizeof	static	struct	switch
typedef	union	unsigned	void
volatile	while		

Tên: Trong chương trình người sử dụng cần tên để đặt cho biến, hàm, kiểu dữ liệu mới. Tên của người sử dụng đòi hỏi phải là một chuỗi kí tự là các chữ cái, chữ số và gạch nối '_'. Tên phải bắt đầu bằng một chữ cái hoặc gạch nối.

2. Cấu trúc cơ bản của chương trình

Trong bài 2 chúng ta đã thấy được cấu trúc cơ bản của chương trình C. Phần này sẽ trình bày khái quát về các thành phần trong cấu trúc của chương trình.

a. Khai báo tiêu đề

Một tệp tiêu đề trong chương trình C chứa mô tả các hàm có thể được gọi trong chương trình (xem mục 6.2 - Tệp tiêu đề). Khi cần gọi một hàm nào trong chương trình ta phải khai báo tệp tiêu đề có mô tả tương ứng cho hàm. Chính vì vậy các hàm thư viện luôn được chỉ ra cùng với các tệp tiêu đề tương ứng khi được trình bày.

b. Khai báo hằng, cấu trúc dữ liệu

Các hằng số trong chương trình C thường được định nghĩa bằng chỉ thị `#define` (xem bài 35 - Một số chỉ thị tiền xử lí). Ngoài các kiểu dữ liệu cơ bản, một chương trình C có thể có các cấu trúc dữ liệu phức do người sử dụng định nghĩa. Các cấu trúc này thường được khai báo ở phần đầu chương trình (xem mục 3.4 - Cấu trúc).

c. Khai báo hàm và/hoặc nguyên mẫu hàm

Chương trình C được cấu trúc bởi các hàm (xem mục 2.3 - Hàm). Một hàm thể hiện một modul con trong một chương trình. Nó có thể nhận tham số thực hiện và trả về kết quả cho nơi gọi hàm. Thân của một hàm được chia làm hai phần: khai báo biến (xem mục 2.1 - Biến, hằng số và biểu thức) và lệnh thực hiện (xem mục 2.2 - Các cấu trúc lệnh điều khiển). Các biến luôn được khai báo ở phần đầu thân hàm sau đó mới đến các câu lệnh. Mỗi câu lệnh trong chương trình C phải được kết thúc bằng một dấu chấm phẩy ‘;’. Chúng ta có thể tạo một khối lệnh bằng cách đặt chúng trong cặp dấu {}. Thân hàm cũng phải được đặt trong cặp dấu này.

d. Khai báo biến

Trong một chương trình C có thể khai báo biến dạng tổng thể hay cục bộ (biến của một hàm). Một biến tổng thể được khai báo bên ngoài hàm và nó có thể sử dụng trong tất cả các hàm của chương trình. Ngược lại biến cục bộ chỉ có thể sử dụng trong hàm nơi nó được khai báo.

Mục 1.2 - Biên dịch chương trình

Mục này đưa ra cho người sử dụng một kiến thức chuyên sâu về vấn đề biên dịch một chương trình. Các trình IDE (Integrated Development Environment) được biết đến như là một môi trường tích hợp trợ giúp việc phát triển chương trình bao gồm các công việc soạn thảo, biên dịch và gỡ rối. Học viên sẽ được tìm hiểu một trình IDE được dùng phổ biến trong môi trường DOS là Turbo C.

Yêu cầu: Có trình cài đặt Turbo C hoặc Borland C++.

Thời lượng: 2 tiết

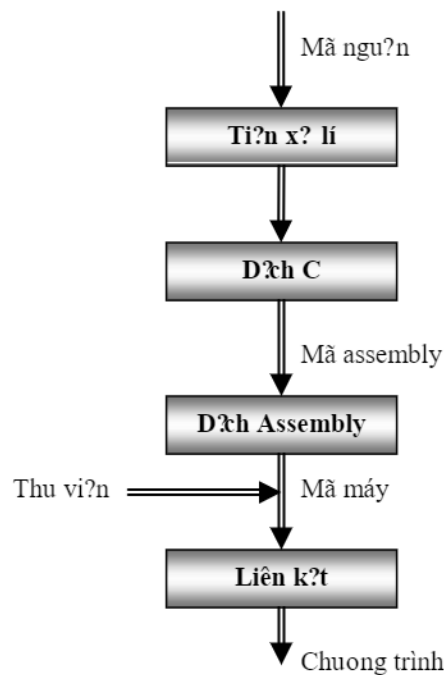
Bài 4 - Quá trình biên dịch

Tóm tắt nội dung:

Biên dịch một chương trình là quá trình dịch một chương trình viết dưới dạng văn bản như C sang chương trình mã máy như tệp .exe trong DOS. Quá trình này có thể phải qua nhiều pha khác nhau và có thể xảy ra lỗi tại mỗi pha.

Thời lượng: 1 tiết

Ngôn ngữ lập trình C là một ngôn ngữ dạng biên dịch. Chương trình được viết bằng C muốn chạy được trên máy tính phải trải qua một quá trình biên dịch để chuyển đổi từ dạng mã nguồn sang chương trình dạng mã thực hiện. Toàn bộ quá trình biên dịch ở trong ngôn ngữ được minh họa trong hình 1. Quá trình này được chia làm hai giai đoạn chính: giai đoạn dịch và giai đoạn liên kết.



Hình 1: Quá trình biên dịch một chương trình C

1. Giai đoạn dịch (compiling)

Mục đích của giai đoạn này là chuyển đổi tất cả mã chương trình được viết dưới ngôn ngữ C sang dạng mã máy. Ba công đoạn được thực hiện lần lượt là: tiền xử lý mã nguồn, dịch C sang Assembly và dịch Assembly sang mã máy.

- *Công đoạn tiền xử lý* sẽ nhận mã nguồn của chương trình và thực hiện xóa bỏ tất cả chú thích của chương trình. Ngoài ra tất cả các chỉ thị tiền xử lý (bắt đầu bằng #) cũng được xử lý ngay trong công đoạn này. Ví dụ chỉ thị `#include` cho phép ghép thêm mã chương trình của một tệp tiêu đề vào mã nguồn cần dịch. Các hằng số được định nghĩa bằng `#define` sẽ được thay thế bằng giá trị cụ thể tại mỗi nơi sử dụng trong chương trình.

- *Công đoạn dịch C* cho phép phân tích cú pháp của mã nguồn C và sau đó chuyển chúng sang dạng mã Assembly là một ngôn ngữ bậc thấp gần với tập lệnh của bộ vi xử lý.
- *Công đoạn dịch Assembly* cho phép dịch chương trình sang mã máy. Sau công đoạn này một tệp mã máy (.o) thường được sinh ra trong hệ thống.

Như vậy sau giai đoạn dịch các lỗi chương trình có thể được phát hiện và thông báo cho người sử dụng là các lỗi về cú pháp chương trình. Nó cũng có thể phát hiện lỗi không tồn tại (hoặc có mà đặt đường dẫn tìm kiếm sai để không tìm thấy) tệp tiêu đề trong công đoạn tiền xử lý.

2. Giai đoạn liên kết (linking)

Trong giai đoạn này mã máy của một chương trình dịch từ nhiều nguồn khác nhau được liên kết lại với nhau để tạo thành chương trình đích duy nhất. Mã máy của các hàm thư viện gọi trong chương trình cũng được đưa vào chương trình cuối trong giai đoạn này. Chính vì vậy mà các lỗi liên quan đến việc gọi hàm hay sử dụng biến tổng thể mà không tồn tại sẽ bị phát hiện. Kể cả lỗi viết chương trình chính không có hàm main() cũng được phát hiện trong khi liên kết. Để liên kết với các hàm thư viện cần phải đặt đúng đường dẫn tìm kiếm thư viện đến các tệp chứa mã máy thư viện (.obj, .lib,...)

Bài 5 - Turbo C

Tóm tắt nội dung:

Turbo C là một trình IDE cho ngôn ngữ C chạy trên nền DOS. Một số phiên bản hay dùng bao gồm Turbo C 1.0, Turbo C++ 3.0. Ngoài ra ta có thể dùng các phiên bản chạy trên nền hệ điều hành Windows có tên là Borland C++.

Thời lượng: 1 tiết

1. Môi trường trợ giúp phát triển (IDE)

Để trợ giúp cho việc phát triển ứng dụng trên một ngôn ngữ một cách thuận tiện, các nhà sản xuất cung cấp các tiện ích cho phép người sử dụng thao tác biên soạn chương trình, dịch hay gỡ rối bằng các thao tác đơn giản qua phím nóng. Như vậy nhờ có IDE (Integrated Development Environment) mà lập trình viên không phải thực hiện các pha biên dịch bằng các câu lệnh phức tạp mà chỉ cần đơn giản là nhấn một phím nóng. Quá trình sửa lỗi cũng rất thuận tiện do IDE định vị giúp người lập trình vị trí lỗi của chương trình ngay trên màn hình soạn thảo. Một IDE thường có ba thành phần cơ bản là:

- *Công cụ soạn thảo* giúp lập trình viên soạn thảo chương trình trên ngôn ngữ được hỗ trợ. Trình soạn thảo này có thể thực hiện đổi màu các từ khóa trong chương trình để giúp người lập trình quan sát chương trình một cách trực quan hơn.
- *Công cụ biên dịch và chạy chương trình.* Bằng một thao tác đơn giản lập trình viên có thể yêu cầu biên dịch toàn bộ chương trình xong rồi chạy. Nếu chương trình có lỗi thì công cụ này sẽ thông báo loại lỗi cùng với vị trí của lỗi trong chương trình đến người sử dụng. Chính vì vậy việc tiến hành sửa lỗi chương trình trong IDE là rất đơn giản.

- *Công cụ gỡ rối* giúp lập trình viên có thể chạy lần bước một chương trình theo từng dòng lệnh. Trong quá trình lần bước lập trình viên có thể quan sát sự thay đổi giá trị các biến trong chương trình qua đó mà phát hiện ra lỗi.

2. Turbo C

Có rất nhiều trình IDE khác nhau cho ngôn ngữ C nhưng được phổ dụng nhất trong môi trường DOS hoặc Windows là các phiên bản của Turbo C. Chúng ta có thể kể ra một số phiên bản hay sử dụng là Turbo C chỉ hỗ trợ ngôn ngữ C và chạy trên nền DOS, Turbo C++ 3.0 hỗ trợ C và C++ chạy trên DOS, các phiên bản từ 3.1 trở lên được đổi tên là Borland C++ và chạy trên nền Windows.

Khi dịch một chương trình trên Turbo C cần chú ý thiết lập đúng đường dẫn tìm kiếm tệp tiêu đề và thư viện. Ví dụ với phiên bản Turbo C++ để xác lập các đường dẫn thư mục này ta cần dùng menu và vào lựa chọn Options|Directories. Khi đó một hộp hội thoại sẽ xuất hiện để cho chúng ta đặt đường dẫn thư mục tìm kiếm tệp tiêu đề (Include) và thư mục tìm kiếm thư viện (Library). Thường thì các tệp tiêu đề được lưu trong thư mục con INCLUDE của thư mục cài đặt Turbo. Còn các tệp thư viện được lưu trong thư mục con LIB.



Hình 2. Màn hình IDE của Turbo C++ 3.0

Chúng ta thường dùng phím nóng của IDE để thực hiện nhanh một số thao tác. Sau đây là một số phím nóng thông dụng:

Phím nóng	Ý nghĩa
F3	Mở tệp nguồn
F2	Ghi tệp
F9	Biên dịch chương trình và tạo tập tin EXE
Ctrl+F9	Biên dịch và chạy chương trình
F4, F7, F8	Chạy lần vết, gỡ rối