## Tham số dòng lệnh (Command Line Arguments)

**Tham số dòng lệnh (Command Line Arguments)** trong ngôn ngữ lập trình C là một cách để truyền dữ liệu vào chương trình thông qua dòng lệnh khi chạy chương trình. Đây là một phần quan trọng trong việc xây dựng các chương trình có thể tương tác với người dùng hoặc hệ điều hành một cách linh hoạt.

### Tổng quan về tham số dòng lệnh

Khi chạy một chương trình C từ dòng lệnh (Command Line), bạn có thể truyền thêm **các tham số** cùng với tên chương trình. Những tham số này được gọi là **tham số dòng lệnh**.

Trong C, tham số dòng lệnh được xử lý thông qua hai tham số đặc biệt của hàm main()

|  |
| --- |
| int main(int argc, char \*argv[]); |

Hoặc

|  |
| --- |
| int main(int argc, char \*\*argv); |

**Ý nghĩa của các tham số:**

1. argc (**Argument Count**):
   * Là số lượng tham số dòng lệnh được truyền vào, bao gồm cả tên của chương trình.
   * argc luôn lớn hơn hoặc bằng 1.
     + Nếu không có tham số nào được truyền (ngoài tên chương trình), argc = 1.
2. argv (**Argument Vector**):
   * Là một mảng các chuỗi ký tự (mảng con trỏ char \*), chứa các tham số dòng lệnh.
   * argv[0] chứa tên chương trình (hoặc đường dẫn đầy đủ của chương trình).
   * argv[1], argv[2], ..., argv[argc-1] chứa các tham số dòng lệnh khác mà người dùng truyền vào.

### Cách sử dụng tham số dòng lệnh

Để sử dụng tham số dòng lệnh, bạn cần thực hiện các bước sau:

1. Định nghĩa hàm main() với hai tham số argc và argv.
2. Truy cập các tham số dòng lệnh thông qua mảng argv.
3. Xử lý tham số dựa trên giá trị của argc.

**VD:** In các tham số dòng lệnh

|  |
| --- |
| #include <stdio.h>  int main(int *argc*, char \**argv*[]) {      printf("Số lượng tham số (argc): %d\n", *argc*);      printf("Danh sách các tham số (argv):\n");      for (int i = 0; i < *argc*; i++) {          printf("argv[%d]: %s\n", i, *argv*[i]);      }      return 0;  } |

Giả sử bạn biên dịch chương trình thành file thực thi program và chạy với lệnh sau:

|  |
| --- |
| ./program hello world |

**Kết quả:**

|  |
| --- |
| Số lượng tham số (argc): 3  Danh sách các tham số (argv):  argv[0]: ./program  argv[1]: hello  argv[2]: world |

### Phân tích sâu về argc và argv

#### Giá trị của argc

* argc là số lượng các tham số truyền vào chương trình.
* **Lưu ý:**
  + argc luôn >= 1 vì argv[0] luôn chứa tên chương trình.

| **Lệnh chạy chương trình** | **argc** | **argv** |
| --- | --- | --- |
| ./program | 1 | argv[0] = ./program |
| ./program arg1 | 2 | argv[0] = ./program, argv[1] = arg1 |
| ./program arg1 arg2 arg3 | 4 | argv[0] = ./program, argv[1] = arg1, ... |

#### Mảng argv

* argv là mảng chứa các chuỗi, mỗi chuỗi là một tham số dòng lệnh.
* **Truy cập các phần tử trong argv:**
  + argv[0]: Tên chương trình.
  + argv[1] đến argv[argc-1]: Các tham số dòng lệnh.
* argv[i] luôn là một chuỗi ký tự (char \*).

**Ví dụ:**

|  |
| --- |
|  |

#include <stdio.h>

int main(int argc, char \*argv[]) {

if (argc > 1) {

printf("Tham số đầu tiên (argv[1]): %s\n", argv[1]);

} else {

printf("Không có tham số nào ngoài tên chương trình.\n");

}

return 0;

}

**Chạy chương trình:**

|  |
| --- |
| ./program example |

**Kết quả:**

Tham số đầu tiên (argv[1]): example

### Chuyển đổi tham số dòng lệnh sang kiểu dữ liệu khác

Các tham số dòng lệnh mặc định là chuỗi ký tự (char \*). Để xử lý số nguyên hoặc số thực, bạn cần chuyển đổi chúng.

#### Chuyển đổi chuỗi thành số nguyên

Sử dụng hàm atoi() (có sẵn trong thư viện <stdlib.h>):

|  |
| --- |
|  |

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char \*argv[]) {

if (argc != 2) {

printf("Sử dụng: %s <số nguyên>\n", argv[0]);

return 1;

}

int num = atoi(argv[1]); // Chuyển chuỗi argv[1] thành số nguyên

printf("Số bạn nhập: %d\n", num);

return 0;

}

**Chạy chương trình:**

|  |
| --- |
| ./program 42 |

**Kết quả:**

|  |
| --- |
| Số bạn nhập: 42 |

#### Chuyển đổi chuỗi thành số thực

Sử dụng hàm atof():

|  |
| --- |
|  |

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char \*argv[]) {

if (argc != 2) {

printf("Sử dụng: %s <số thực>\n", argv[0]);

return 1;

}

double num = atof(argv[1]); // Chuyển chuỗi argv[1] thành số thực

printf("Số thực bạn nhập: %.2f\n", num);

return 0;

}

**Chạy chương trình:**

|  |
| --- |
| ./program 3.14 |

**Kết quả:**

|  |
| --- |
| Số thực bạn nhập: 3.14 |

### Một số ví dụ ứng dụng tham số dòng lệnh

**VD:** Cộng hai số nguyên truyền qua dòng lệnh

|  |
| --- |
|  |

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char \*argv[]) {

if (argc != 3) {

printf("Sử dụng: %s <số nguyên 1> <số nguyên 2>\n", argv[0]);

return 1;

}

int num1 = atoi(argv[1]);

int num2 = atoi(argv[2]);

printf("Tổng: %d\n", num1 + num2);

return 0;

}

**Chạy chương trình:**

./program 10 20

**Kết quả:**

Tổng: 30

#### In tất cả các tham số dòng lệnh

|  |
| --- |
|  |

#include <stdio.h>

int main(int argc, char \*argv[]) {

printf("Bạn đã nhập %d tham số:\n", argc);

for (int i = 0; i < argc; i++) {

printf("argv[%d]: %s\n", i, argv[i]);

}

return 0;

}

**Chạy chương trình:**

./program this is a test

**Kết quả:**

|  |
| --- |
| Bạn đã nhập 5 tham số:  argv[0]: ./program  argv[1]: this  argv[2]: is  argv[3]: a  argv[4]: test |

**VD:** Kiểm tra một chuỗi là số nguyên hợp lệ

|  |
| --- |
|  |

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

// Hàm kiểm tra xem chuỗi có phải là số nguyên hay không

int isInteger(char \*str) {

for (int i = 0; str[i] != '\0'; i++) {

if (!isdigit(str[i])) {

return 0; // Không phải số

}

}

return 1; // Là số nguyên

}

int main(int argc, char \*argv[]) {

if (argc != 2) {

printf("Sử dụng: %s <chuỗi cần kiểm tra>\n", argv[0]);

return 1;

}

if (isInteger(argv[1])) {

printf("Chuỗi \"%s\" là một số nguyên hợp lệ.\n", argv[1]);

} else {

printf("Chuỗi \"%s\" không phải là số nguyên.\n", argv[1]);

}

return 0;

}

**Chạy chương trình:**

|  |
| --- |
| ./program 12345  ./program abc123 |

**Kết quả:**

|  |
| --- |
| Chuỗi "12345" là một số nguyên hợp lệ.  Chuỗi "abc123" không phải là số nguyên. |

### Tổng kết

* **argc**: Số lượng tham số dòng lệnh, bao gồm cả tên chương trình.
* **argv**: Mảng các chuỗi, chứa tên chương trình và các tham số dòng lệnh.
* Tham số dòng lệnh được sử dụng để truyền dữ liệu vào chương trình khi chạy từ dòng lệnh.
* Có thể chuyển đổi các tham số dòng lệnh từ chuỗi sang số nguyên (atoi) hoặc số thực (atof).
* Ứng dụng: Giao tiếp với người dùng, viết các chương trình linh hoạt, xử lý dữ liệu từ dòng lệnh.

Nếu bạn cần giải thích thêm hoặc ví dụ cụ thể hơn, hãy cho mình biết nhé! 😊