

VÕ TIẾN

Thảo luận kiến thức CNTT trường BK về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/khmt.ktmt.cse.bku>



Kỹ Thuật Lập Trình (Cơ bản và nâng cao C++)

KTLT1 - HK242

Chương 0 Chuẩn bị trước khi code

Thảo luận kiến thức CNTT trường BK
về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/khmt.ktmt.cse.bku>

Mục lục

1	Cài môi trường	2
1.1	Cài đặt g++	2
1.2	Tải Vscode	2
2	Các câu lệnh dùng trên terminal	3
2.1	Quản lý thư mục	3
2.2	Quản lý file	3
2.3	Tìm kiếm file/thư mục	3
2.4	Ứng dụng trong c++	3
3	Cách hoạt động g++ trong quá trình biên dịch c++	4
3.1	Viết mã và biên dịch	4
3.2	Thực thi chương trình	5
3.3	Câu hỏi nhiều đáp án	6

**YÊU CẦU CÁC BẠN ĐỌC PHẦN NỘI QUY VÀ
THAY ĐỔI BIỆT DANH DISCORD ĐÚNG CHUẨN**

https://docs.google.com/spreadsheets/d/1Z080oUPx_wNMksG5Hd5gTrqf1re4BBxkZ1S1QTox6gI/edit?gid=0#gid=0

<https://discord.com/channels/1334472759945990184/1334476836104835134>



1 Cài môi trường

1.1 Cài đặt g++

G++ là một trình biên dịch (compiler) cho ngôn ngữ C++ thuộc bộ công cụ GCC (GNU Compiler Collection) – một bộ biên dịch mã nguồn mở phổ biến được phát triển bởi Free Software Foundation (FSF).

1. Tải G++
2. Kiểm tra phiên bản g++
`g++ --version`
3. Viết một đoạn code cơ bản

- tạo ra file main.cpp

```
1 // VÕ TIẾN - KHÓA HỌC KTLT, CTRR, DSA, KTMT, PPL, MNLT, ...
2
3 #include <iostream>
4
5 int main() {
6     std::cout << "Hello World!";
7     return 0;
8 }
```

- Mở terminal và đưa đường dẫn đến vị trí main.cpp
- Build code với lệnh sau để tạo ra file **main.exe** hoặc **main**
`g++ -o main main.cpp`
- Run code với lệnh sau để in ra kết quả trên terminal
`./main`
hoặc
`./main.exe`

1.2 Tải Vscode

<https://code.visualstudio.com/>

Một số extensions trong vscode các bạn tự tìm hiểu nha



2 Các câu lệnh dùng trên terminal

2.1 Quản lý thư mục

Chức năng	CMD	PowerShell
Hiển thị thư mục hiện tại	cd	Get-Location
Liệt kê file/thư mục	dir	Get-ChildItem
Di chuyển thư mục	cd folder	Set-Location folder
Trở về thư mục trước	cd ..	cd ..
Về thư mục gốc	cd %HOMEPATH%	cd ~
Tạo thư mục	mkdir new_folder	New-Item -ItemType Directory -Path new_folder
Xóa thư mục	rmdir /s /q folder	Remove-Item folder -Recurse -Force

Bảng 1: Các lệnh quản lý thư mục trên Windows

2.2 Quản lý file

Chức năng	CMD	PowerShell
Tạo file mới	type nul > file.txt	New-Item file.txt
Sao chép file	copy file1.txt file2.txt	Copy-Item file1.txt file2.txt
Sao chép thư mục	xcopy folder1 folder2 /E /I	Copy-Item -Recurse folder1 folder2
Di chuyển file/thư mục	move file.txt new_folder\	Move-Item file.txt new_folder\
Đổi tên file/thư mục	rename old.txt new.txt	Rename-Item old.txt new.txt
Xóa file	del file.txt	Remove-Item file.txt
Mở file	start file.txt	Invoke-Item file.txt

Bảng 2: Các lệnh quản lý file trên Windows

2.3 Tìm kiếm file/thư mục

Chức năng	CMD	PowerShell
Tìm file trong thư mục	dir /s /b *file.txt	Get-ChildItem -Path . -Filter "*file.txt" -Recurse
Kiểm tra dung lượng file	dir file.txt	(Get-Item file.txt).length

Bảng 3: Các lệnh tìm kiếm và kiểm tra file trên Windows

2.4 Ứng dụng trong c++

1. Biên dịch và chạy nhanh: `g++ main.cpp -o main.exe && main.exe`
2. Biên dịch tất cả file C++: `g++ *.cpp -o output.exe`
3. Tạo thư mục, di chuyển file, biên dịch và chạy: `mkdir build && move main.cpp build\&& cd build && g++ main.cpp -o main.exe && main.exe`
4. Kiểm tra dung lượng file thực thi: `g++ main.cpp -o main.exe && dir main.exe`
5. Xóa file cũ trước khi biên dịch: `del main.exe && g++ main.cpp -o main.exe && main.exe`
6. Tạo file header trong C++: `echo <!DOCTYPE header.h> && notepad header.h`
7. Chạy chương trình C++ trong terminal không cần mở thư mục: `g++ path_to_your_file/main.cpp -o main.exe && main.exe`
8. Biên dịch với nhiều cờ tối ưu hóa: `g++ -O2 main.cpp -o main.exe`
9. Liệt kê tất cả thư mục con trong thư mục hiện tại: `dir /s /b`
10. Chạy chương trình và ghi đầu ra vào file log: `main.exe > log.txt`



3 Cách hoạt động g++ trong quá trình biên dịch c++

3.1 Viết mã và biên dịch

Bước 1: Viết mã nguồn

Bạn viết mã nguồn trong một tệp .cpp

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Hello, World!" << endl;
6     return 0;
7 }
```

Bước 2: Biên dịch mã nguồn

Sử dụng trình biên dịch như g++ để biến mã nguồn thành tệp thực thi:

```
g++ main.cpp -o main
```

- **Tiền xử lý:** Thay thế các chỉ thị như include| bằng mã thực tế từ thư viện.
- **Biên dịch:** Chuyển mã nguồn C++ sang mã Assembly.
- **Lắp ráp:** Chuyển mã Assembly thành mã máy (Object Code).
- **Liên kết:** Kết nối mã máy với các thư viện cần thiết, tạo tệp thực thi.

Cấu trúc biên dịch lệnh bằng g++

```
g++ [options] input_files -o output_file
```

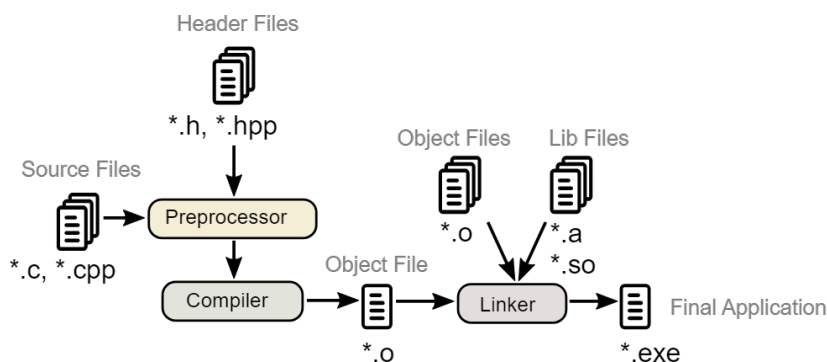
- **[options]:** Các tùy chọn cho trình biên dịch (ví dụ: tối ưu hóa, gỡ lỗi, liên kết thư viện).
- **input_files:** Các tệp mã nguồn (.cpp).
- **-o output_file:** Tên tệp thực thi sau khi biên dịch. Nếu bỏ qua, tên mặc định là a.out trên Linux hoặc a.exe trên Windows.

Bước 3: Tạo tệp thực thi

Sau khi biên dịch, tạo ra tệp thực thi (ví dụ: main trên Linux hoặc main.exe trên Windows) là một file nhị phân.

```
// file main
```

```
0100100001001000010010000100100001001000010010000100100001001000
0100100001001000010010000100100001001000010010000100100001001000
0100100001001000010010000100100001001000010010000100100001001000
0100100001001000010010000100100001001000010010000100100001001000
0100100001001000010010000100100001001000010010000100100001001000
```





3.2 Thực thi chương trình

Bước 4: Tải tệp thực thi

Khi chạy lệnh:

```
./main
```

Hệ điều hành:

- Tìm tệp thực thi main.
- Tải tệp vào bộ nhớ RAM.
- Phân bổ vùng nhớ (Code Segment, Data Segment, Stack, Heap).

Bước 5: Liên kết thư viện động

Nếu chương trình sử dụng thư viện như iostream, hệ điều hành sẽ tải các thư viện này vào bộ nhớ nếu chưa có.

Bước 6: Bắt đầu thực thi

CPU bắt đầu thực hiện chương trình tại điểm đầu (`_start`) và gọi hàm `main`.

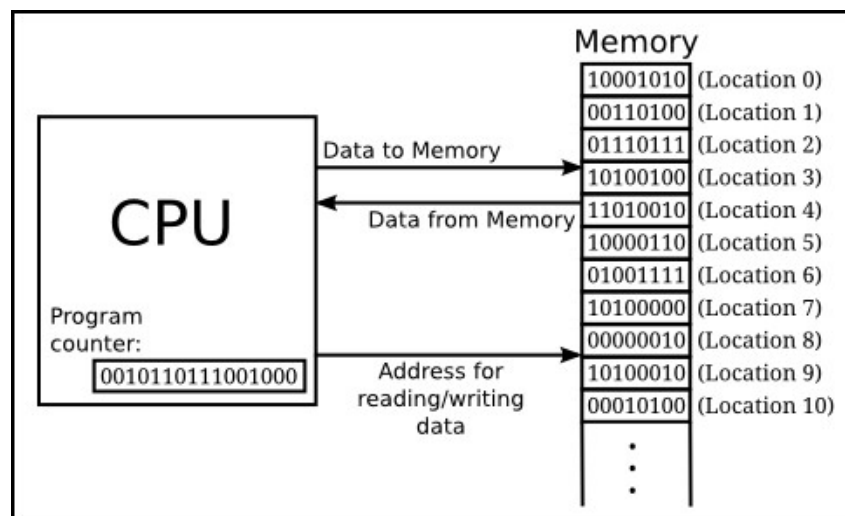
Bước 7: Thực thi hàm `main`

Chương trình thực hiện tuần tự mã lệnh trong hàm `main`: `cout: Lệnh cout << "Hello, World!" << endl; in chuỗi "Hello, World!" ra màn hình.`

Bước 8: Gọi hàm hệ thống

Bước 9: Hiển thị kết quả

Bước 10: Kết thúc chương trình





3.3 Câu hỏi nhiều đáp án

1. Biên dịch toàn bộ các tệp trong thư mục này để tạo tệp thực thi có tên `program`. Lệnh nào dưới đây đúng?

VoTien/

```
-- main.cpp
-- utils.cpp
-- utils.h
-- math.cpp
-- math.h
```

- a) `g++ main.cpp -o program`
 - b) `g++ main.cpp utils.cpp math.cpp -o program`
 - c) `g++ *.cpp -o program`
 - d) `g++ -o program`
2. Nếu bạn muốn biên dịch từng tệp trong thư mục trên thành các tệp đối tượng (`.o`), lệnh nào sau đây là đúng?
- a) `g++ -c main.cpp utils.cpp math.cpp`
 - b) `g++ -c *.cpp`
 - c) `g++ *.cpp -o program`
 - d) `g++ -o main.o utils.o math.o`
3. Sau khi tạo các tệp đối tượng (`.o`), lệnh nào dưới đây dùng để liên kết chúng thành tệp thực thi `program`?
- a) `g++ -o program main.o utils.o math.o`
 - b) `g++ -c main.o utils.o math.o -o program`
 - c) `g++ *.o -o program`
 - d) `g++ main.o utils.o math.o`
4. Nếu bạn muốn bật chế độ gỡ lỗi khi biên dịch để hỗ trợ công cụ `gdb`, lệnh nào đúng?
- a) `g++ -g *.cpp -o program`
 - b) `g++ -debug *.cpp -o program`
 - c) `g++ -o program -g *.cpp`
 - d) Cả A và C
5. Để tối ưu hóa kích thước tệp thực thi và biên dịch với chuẩn C++17, lệnh nào dưới đây đúng?
- a) `g++ -std=c++17 -Os *.cpp -o program`
 - b) `g++ -std=c++17 -O3 *.cpp -o program`
 - c) `g++ -O2 -std=c++17 *.cpp -o program`
 - d) Cả A và B

6. Biên dịch toàn bộ các tệp trong thư mục sau để tạo tệp thực thi có tên `program`. Lệnh nào dưới đây đúng?

VoTien/

```
-- src/
--   main.cpp
--   utils/
--     utils.cpp
--     utils.h
--   math/
--     math.cpp
--     math.h
-- include/
--   constants.h
-- build/
```



- a) `g++ src/main.cpp src/utils/utils.cpp src/math/math.cpp -o program`
b) `g++ src/*.cpp -o program`
c) `g++ src/main.cpp src/utils/*.cpp src/math/*.cpp -o program`
d) `g++ *.cpp -o program`
7. Nếu bạn muốn lưu các tệp đối tượng (.o) vào thư mục build/, lệnh nào dưới đây là đúng?
- a) `g++ -c src/main.cpp -o build/main.o`
b) `g++ -c src/utils/utils.cpp -o build/utils.o`
c) `g++ -c src/math/math.cpp -o build/math.o`
d) Cả A, B và C
8. Sau khi tạo các tệp .o, bạn muốn liên kết chúng thành tệp thực thi. Lệnh nào dưới đây là đúng?
- a) `g++ build/main.o build/utils.o build/math.o -o program`
b) `g++ src/*.o -o program`
c) `g++ build/*.o -o program`
d) Cả A và C
9. Thư mục include/ chứa các tệp tiêu đề dùng chung (constants.h). Để chỉ định đường dẫn thư mục này khi biên dịch, bạn cần sử dụng lệnh nào?
- a) `g++ -Iinclude src/*.cpp -o program`
b) `g++ -include include/constants.h src/*.cpp -o program`
c) `g++ -Iinclude src/main.cpp src/utils/utils.cpp src/math/math.cpp -o program`
d) Cả A và C

Câu	Đáp án	Câu	Đáp án	Câu	Đáp án	Câu	Đáp án	Câu	Đáp án
1	Đáp án 1	2	Đáp án 2	3	Đáp án 3	4	Đáp án 40	5	Đáp án 5
6	Đáp án 6	7	Đáp án 7	8	Đáp án 8	9	Đáp án 9		

Bảng 4: Trắc nghiệm