

工程师组 第一次分享会

俞贤皓

目录

1. 聊天
2. WSL
3. C/C++基本编译知识

1. 聊天

1. 聊天

- 想大概讲一些有关“目标”的事情

1. 聊天

- 想大概讲一些有关“目标”的事情
- 工作
- 读研究生
- 待业

1. 聊天

- 想大概讲一些有关“目标”的事情
- 工作
- 读研究生
 - 保研
 - 考研（大学的高考）
 - 出国
- 待业

1. 聊天

- 想大概讲一些有关“目标”的事情
- 工作
- 读研究生
 - 保研（前六学期，总平均分年段前30）
 - 考研（大学的高考）
 - 出国
- 待业

2. WSL

2. WSL

- 为什么要用WSL？为什么要用Linux？

2. WSL

- 为什么要用WSL？为什么要用Linux？
- 好处很多，非常多！

2. WSL

- 为什么要用WSL？为什么要用Linux？
- 好处很多，非常多！
- 但我不喜欢说这种很空的理由，大家有兴趣可以上 [知乎](#)

2. WSL

- 为什么要用WSL？为什么要用Linux？
- 好处很多，非常多！
- 但我不喜欢说这种很空的理由，大家有兴趣可以上 [知乎](#)
- 其他学校的例子

2. WSL

21:11:47

对了，话说你们是什么时候开始用vscode + wsl的



我这周准备给我们学校的学弟讲wsl的这套知识，所以想引用一些其他学校的经验（



我们从大一开始就是这么用的

wsl + vscode/CLion

刚刚入学吗？还是比如学期中这样的



WOW



刚入学

2. WSL



主要是大部分工作都是在linux环境下干的

2. WSL



基本都在用



话说你们那边用wsl的人多吗



WOW



2. WSL



YXH_XianYu

23/5/8

对了老师，我有个问题，大家平常开发用windows的多
还是用linux的多？



YXH_XianYu

23/5/8

我刚刚装这个AgentVerse，仓库自带的安装是linux的



韩文娟

23/5/8

linux。。

2. WSL

2. WSL

- 如何更好地使用Linux（WSL）？

2. WSL

- 如何更好地使用Linux（WSL）？
- [Linux基本命令](#)

2. WSL

- 如何更好地使用Linux（WSL）？
- [Linux基本命令](#)
- 翻*

2. WSL

- 如何更好地使用Linux（WSL）？
- [Linux基本命令](#)
- 翻*
 - [给包管理器换源](#)（如果上一条失败的话）

2. WSL

- 如何更好地使用Linux（WSL）？

- Linux基本命令

- 翻*

- 给包管理器换源

- tldr

```
● yxh_xianyu ~> tldr gcc
gcc
Preprocess and compile C and C++

- Compile multiple source files
gcc {{path/to/source1.c path/

- Show common warnings, debug sy
gcc {{path/to/source.c}} -Wal

- Include libraries from a diff
gcc {{path/to/source.c}} -o {

- Compile source code into Assem
gcc -S {{path/to/source.c}}

- Compile source code into an o
gcc -c {{path/to/source.c}}
```

```
● yxh_xianyu ~> tldr cd
cd
Change the current working direc

- Go to the specified directory
cd {{path/to/directory}}

- Go up to the parent of the cu
cd ..

- Go to the home directory of t
cd

- Go to the home directory of t
cd ~{{username}}

- Go to the previously chosen d
cd -

- Go to the root directory:
cd /
```

2. WSL

- 如何更好地使用Linux

- [Linux基本命令](#)

- 翻*

- [给包管理器换源](#)

- tldr

- tree

```
yxh_xianyu ~/c-cpp-basic-compilation-tutorial (main)> ls
01 02 03 04 05 06 README.md
```

```
yxh_xianyu ~/c-cpp-basic-compilation-tutorial (main)> tree -L 2
.
├── 01
│   └── main.c
├── 02
│   ├── add.c
│   └── main.c
├── 03
│   ├── add.c
│   ├── add.h
│   └── main.c
├── 04
│   └── main.c
└── 05
    └── A.c
```

2. WSL

- 如何更好地使用Linux（WSL）？
- [Linux基本命令](#)
- 翻*
 - [给包管理器换源](#)
- tldr
- tree
- git&github
 - 之后可能会讲

2. WSL

- 如何更好地使用Linux（WSL）？

- [Linux基本命令](#)

- 翻*

- [给包管理器换源](#)

- tldr

- tree

- git&github

- 之后可能会讲

- shell（fish、zsh等）

```
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
yhx_xianyu ~> 
```

```
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
yhx_xianyu ~> cd c-cpp-basic-compilation-tutorial/
```

2. WSL

- 如何更好地使用Linux（WSL）？
- [Linux基本命令](#)
- 翻*
 - [给包管理器换源](#)
- tldr
- tree
- git&github
 - 之后可能会讲
 - etc...
- shell（fish、zsh等）
- （或许可以尝试直接装一台真正的Linux机器）

3. C/C++基本编译知识

3. C/C++基本编译知识 - 引入

```
通... music-10-10.cpp(.text+0xd0): undefined reference to `circle(int, int, int)'  
通... music-10-10.cpp(.text+0x116): undefined reference to `setfillcolor(unsigned lon  
通... music-10-10.cpp(.text+0x131): undefined reference to `rectangle(int, int, int, in  
通... music-10-10.cpp(.text+0x15a): undefined reference to `initgraph(int, int, int)'  
通... [Error] ld returned 1 exit status  
通... recipe for target '项目1.exe' failed
```

这玩意不是在graphics.h下面吗

```
music-10-10.cpp(.text+0xd0): undefined reference to `circle(int, int, int)'  
music-10-10.cpp(.text+0x116): undefined reference to `setfillcolor(unsigned lon  
music-10-10.cpp(.text+0x131): undefined reference to `rectangle(int, int, int, int  
music-10-10.cpp(.text+0x15a): undefined reference to `initgraph(int, int, int)'  
[Error] ld returned 1 exit status  
recipe for target '项目1.exe' failed
```

3. C/C++基本编译知识 - 引入



```
cannot find -lgraphics64  
[Error] ld returned 1 exit status
```

3. C/C++基本编译知识 - 引入

文件名	行	列	
global.h	6		multiple definition of `size'
global.h	6		first defined here
global.h	9		multiple definition of `paddress'; st
global.h	10		multiple definition of `count_nodes';
global.h	6		multiple definition of `size'; stu_se
global.h	9		multiple definition of `paddress'; st
global.h	10		multiple definition of `count_nodes';
global.h	6		multiple definition of `size'; stu_se
global.h	9		multiple definition of `paddress'; st
global.h	10		multiple definition of `count_nodes';
collect2.e...			[错误] ld returned 1 exit status

消息 编译器 (12) 工具输出 调试 查找 TODO

3. C/C++基本编译知识-01

3. C/C++基本编译知识 - 01

```
01 > C main.c > ...
1   #include <stdio.h>
2
3   int add(int a, int b) {
4       return a + b;
5   }
6
7   int main() {
8       int a, b;
9       scanf("%d%d", &a, &b);
10
11       int c = add(a, b);
12       printf("%d\n", c);
13
14       return 0;
15   }
```

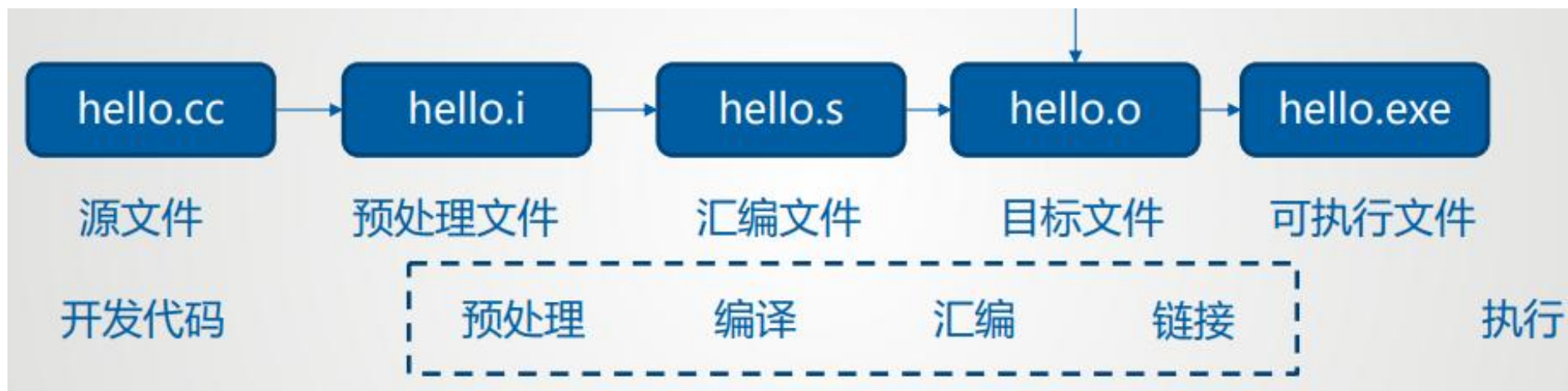
```
• yxh_xianyu ~/c/01 (main)> gcc main.c -o a.out
• yxh_xianyu ~/c/01 (main)> ls
  a.out  main.c
• yxh_xianyu ~/c/01 (main)> ./a.out
1 2
3
```

- 源代码是如何变成可执行文件的？

3. C/C++基本编译知识 - 01



3. C/C++基本编译知识 - 01



- ① 预处理

- `gcc -E main.c -o main.i`

- ② 编译

- `gcc -S main.i -o main.s`

- ③ 汇编

- `gcc -c main.s -o main.o`

- ④ 链接

- `gcc main.o -o main`

3. C/C++基本编译知识-02

```
02 > C main.c > ...
1  #include<stdio.h>
2
3
4  int add(int a, int b);
5
6
7
8  int main() {
9      int a, b;
10     scanf("%d%d", &a, &b);
11
12     int c = add(a, b);
13     printf("%d\n", c);
14
15     return 0;
16 }
```

```
02 > C add.c > ...
1
2  int add(int a, int b) {
3      return a + b;
4  }
```

- yxh_xianyu ~/c/02 (main)> gcc main.c add.c -o main
- yxh_xianyu ~/c/02 (main)> ls
add.c main main.c
- yxh_xianyu ~/c/02 (main)> ./main
1 2
3

3. C/C++基本编译知识 - 02



- 在链接阶段，链接器会把多个目标文件进行链接
- 最后，得到可执行文件

3. C/C++基本编译知识 - 02



- 在链接阶段，链接器会把多个目标文件进行链接
- 最后，得到可执行文件
- 问题：scanf 和 printf 这两个函数的定义在哪里？

3. C/C++基本编译知识 - 02



- 在链接阶段，链接器会把多个目标文件进行链接
- 最后，得到可执行文件
- 问题：scanf 和 printf 这两个函数的定义在哪里？
- 使用 `ldd ./main`

3. C/C++基本编译知识 - 02

- 多说一句
- 在编译时，加入类似 `-lgraphics` `-L/path/to/library` 的编译选项
- 就是为了让链接器帮你多链接一个静态或动态链接库

```
-static-libgcc -static-libgcc -lgraphics64 -luuid -lmsimg32 -lgdi32 -  
limm32 -lole32 -loleaut32 -lgdiplus -lwinmm -mwindows -leasyx
```

3. C/C++基本编译知识 - 03

- 头文件呢？

3. C/C++基本编译知识 - 03

- 头文件呢？



3. C/C++基本编译知识 - 03

```
03 > C main.c > ...
1  #include <stdio.h>
2
3
4  #include "add.h"
5
6
7
8  int main() {
9      int a, b;
10     scanf("%d%d", &a, &b);
11
12     int c = add(a, b);
13     printf("%d\n", c);
14
15     return 0;
16 }
```

```
add.c > ...
#include "add.h"

int add(int a, int b) {
    return a + b;
}
```

```
add.h > ...
#ifndef _ADD_H_
#define _ADD_H_

int add(int a, int b);

#endif // _ADD_H_
```

- yxh_xianyu ~/c/03 (main)> gcc main.c add.c -o main
- yxh_xianyu ~/c/03 (main)> ls
add.c add.h main main.c
- yxh_xianyu ~/c/03 (main)> ./main
1 2
3

3. C/C++基本编译知识 - 实例

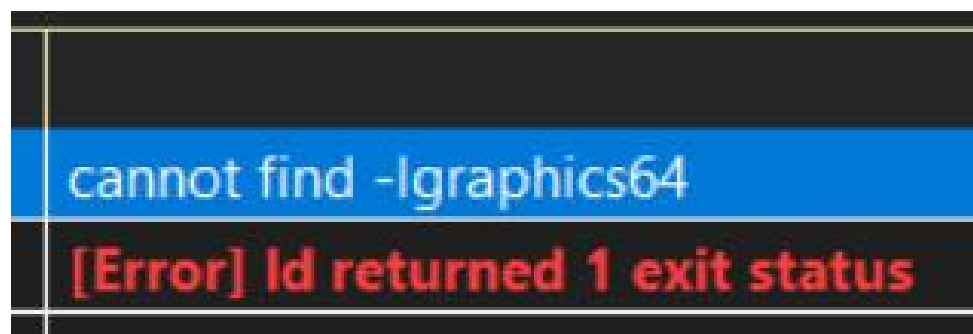
3. C/C++基本编译知识 - 实例



```
music-10-10.cpp:(.text+0xd0): undefined reference to `setcolor(int, int)'\nmusic-10-10.cpp:(.text+0x116): undefined reference to `setfillcolor(unsigned lon\nmusic-10-10.cpp:(.text+0x131): undefined reference to `rectangle(int, int, int, int\nmusic-10-10.cpp:(.text+0x15a): undefined reference to `initgraph(int, int, int)'\n[Error] ld returned 1 exit status\nrecipe for target '项目1.exe' failed
```

- 只 引用了头文件
- 但在 **链接** 阶段，没有找到对应的函数定义
- 解决：需要添加编译命令，添加链接库
- （如 -lgraphics）

3. C/C++基本编译知识 - 实例



- 引用了头文件，也添加了参数
- 但是链接器找不到对应的库文件
- 解决：路径问题。可能是库文件放错文件夹了，也有可能是少了 `-Lpath` 之类的参数

3. C/C++基本编译知识 - 实例

文件名	行	列	
global.h	6		multiple definition of `size'
global.h	6		first defined here
global.h	9		multiple definition of `paddress'; st
global.h	10		multiple definition of `count_nodes';
global.h	6		multiple definition of `size'; stu_se
global.h	9		multiple definition of `paddress'; st
global.h	10		multiple definition of `count_nodes';
global.h	6		multiple definition of `size'; stu_se
global.h	9		multiple definition of `paddress'; st
global.h	10		multiple definition of `count_nodes';
collect2.e...			[错误] ld returned 1 exit status

- multiple definition
- 重定义

3. C/C++基本编译知识 - 04

```
04 > C main.c > ...
1  #include <stdio.h>
2
3  int add(int a, int b) {
4      return a + b;
5  }
6
7  int add(int a, int b) {
8      return a + b;
9  }
10
11 int main() {
12     int a, b;
13     scanf("%d%d", &a, &b);
14
15     int c = add(a, b);
16     printf("%d\n", c);
17
18     return 0;
19 }
```

- C/C++, 多次声明, 一次定义

```
● yxh_xianyu ~/c/04 (main)> code main.c
⊗ yxh_xianyu ~/c/04 (main)> gcc main.c -o a.out
main.c:7:5: error: redefinition of 'add'
    7 | int add(int a, int b) {
      |     ^~~~
main.c:3:5: note: previous definition of 'add' with type 'int(int, int)'
    3 | int add(int a, int b) {
      |     ^~~~
```

3. C/C++基本编译知识 - 04

```
04 > C main.c > ...
1  #include <stdio.h>
2
3  int add(int a, int b) {
4      return a + b;
5  }
6
7  int add(int a, int b) {
8      return a + b;
9  }
10
11 int main() {
12     int a, b;
13     scanf("%d%d", &a, &b);
14
15     int c = add(a, b);
16     printf("%d\n", c);
17
18     return 0;
19 }
```

- C/C++, 多次声明, 一次定义
- 重定义, 只会在编译的时候发生吗?

```
● yxh_xianyu ~/c/04 (main)> code main.c
⊗ yxh_xianyu ~/c/04 (main)> gcc main.c -o a.out
main.c:7:5: error: redefinition of 'add'
    7 | int add(int a, int b) {
      |     ^~~~
main.c:3:5: note: previous definition of 'add' with type 'int(int, int)'
    3 | int add(int a, int b) {
      |     ^~~~
```


3. C/C++基本编译知识 - 05

```
05 > C main.c > ...  
1  #include "A.h"  
1  #include "B.h"  
2  
3  int main() {  
4      return 0;  
5  }
```

```
05 > C A.h > add(int, int)  
1  int add(int a, int b);
```

```
05 > C B.h > add(int, int)  
1  int add(int a, int b);
```

```
05 > C A.c > ...  
1  #include "A.h"  
1  
2  int add(int a, int b) {  
3      return a + b;  
4  }
```

```
05 > C B.c > ...  
1  #include "B.h"  
1  
2  int add(int a, int b) {  
3      return a + b;  
4  }
```

```
yxh_xianyu ~/c/05 (main)> gcc main.c A.c B.c -o a.out  
/usr/bin/ld: /tmp/ccIfwWu0.o: in function `add':  
B.c:(.text+0x0): multiple definition of `add' /tmp/ccdcCc0q.o:A.c:(.text+0x0): first defined here  
collect2: error: ld returned 1 exit status
```

3. C/C++基本编译知识 - 05

文件名	行	列	
global.h	6		multiple definition of `size'
global.h	6		first defined here
global.h	9		multiple definition of `paddress'; st
global.h	10		multiple definition of `count_nodes';
global.h	6		multiple definition of `size'; stu_se
global.h	9		multiple definition of `paddress'; st
global.h	10		multiple definition of `count_nodes';
global.h	6		multiple definition of `size'; stu_se
global.h	9		multiple definition of `paddress'; st
global.h	10		multiple definition of `count_nodes';
collect2.e...			[错误] ld returned 1 exit status

- 再仔细一看
- 重定义的不是函数，是变量！

3. C/C++基本编译知识 - 06

- （今天最后一个知识点）
- 函数声明我会写
- 但是，变量的声明怎么写？

3. C/C++基本编译知识 - 06

- 变量的声明怎么写?
- 使用 `extern` 关键字
 - `extern int globalVar; // 声明一个变量`
 - `int globalVar; // 定义一个变量（无初值）`
 - `int globalVar = 0; // 定义一个变量（初值=0）`

```
● yxh_xianyu ~/c/06 (main)> gcc main.c globalVar.c add1.c add2.c get.c -o main
● yxh_xianyu ~/c/06 (main)> ./main
0
```

结束啦！

- 这次分享会的所有材料都在 [这个Github仓库](#) 中
- 如果你从今天的分享会中能学到一些知识，那么我会很高兴
- 如果你学有余力的话，可以参考如下视频，进行进一步的学习
 - [Makefile入门](#) 20分钟：写gcc命令很麻烦？使用Makefile帮你生成
 - [CMake入门](#) 6分钟：写Makefile很麻烦？使用CMake帮你生成
 - [小彭老师的C/C++编译基本知识](#) 80分钟：更深入的介绍，如果你觉得今天的知识太简单，那么你可以试试这个视频（这位up特别特别特别厉害，建议关注）

谢谢大家
