

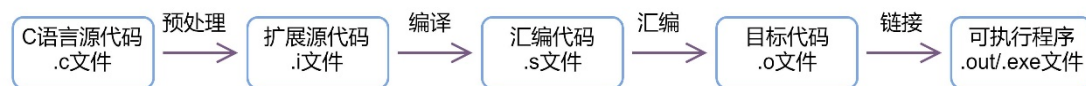
##写在前面

本人之前没有过任何 C 语言的编程经验，所以这次任务从自学开始一路摸爬滚打走来，再加上看到 Dian 招新任务已经是 9 月 16 号左右了，时间比较紧，如果有做得不够好的地方还请各位前辈见谅~

##关于 gcc 编译过程

实验流程

首先，通过简单的 google 我们知道，C 语言的编译过程大致如下：



我们先对最简单的 Hello World 文件进行编译

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("hello world!\n");
6      return 0;
7  }
8
```

进入命令行用 gcc -save-temps hello.c -o compilation，把所有的编译产生的文件保存下来 gcc 的具体编译指令学习到

gcc Hello.c -o Hello.i，将.c 转换为.i

看 hello.i

```
# 0 "hello.c"
# 0 "<built-in>"
# 0 "<command-line>"
# 1 "hello.c"
# 1 "C:/Program Files/mingw64/x86_64-mingw32/include/stdio.h" 1 3
# 9 "C:/Program Files/mingw64/x86_64-mingw32/include/stdio.h" 3
# 1 "C:/Program Files/mingw64/x86_64-mingw32/include/corecrt_stdio_config.h" 1 3
# 10 "C:/Program Files/mingw64/x86_64-mingw32/include/corecrt_stdio_config.h" 3
# 1 "C:/Program Files/mingw64/x86_64-mingw32/include/corecrt.h" 1 3
# 10 "C:/Program Files/mingw64/x86_64-mingw32/include/corecrt.h" 3
# 1 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw.h" 1 3
# 10 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw.h" 3
# 1 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw_mac.h" 1 3
# 98 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw_mac.h" 3
# 107 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw_mac.h" 3
# 306 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw_mac.h" 3
# 384 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw_mac.h" 3
# 11 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw.h" 2 3
# 1 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw_secapi.h" 1 3
# 12 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw.h" 2 3
# 282 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw.h" 3
# 1 "C:/Program Files/mingw64/x86_64-mingw32/include/vadefs.h" 1 3
# 9 "C:/Program Files/mingw64/x86_64-mingw32/include/vadefs.h" 3
# 1 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw.h" 1 3
# 661 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw.h" 3
# 1 "C:/Program Files/mingw64/x86_64-mingw32/include/sdks/_mingw_ddk.h" 1 3
# 662 "C:/Program Files/mingw64/x86_64-mingw32/include/_mingw.h" 2 3
# 10 "C:/Program Files/mingw64/x86_64-mingw32/include/vadefs.h" 2 3
#pragma pack(push, _CRT_PACKING)
# 24 "C:/Program Files/mingw64/x86_64-mingw32/include/vadefs.h" 3
# 24 "C:/Program Files/mingw64/x86_64-mingw32/include/vadefs.h" 3
typedef __builtin_va_list __gnuc_va_list;
```

观察到这一步首先把头文件内容引入了，貌似还替换了宏常量好奇去 c 盘找到对应的文件

stdio.h 打开

```
#ifndef _INC_STDIO
#define _INC_STDIO

#include <corecrt_stdio_config.h>

#pragma pack(push, _CRT_PACKING)

#pragma push_macro("snprintf")
#undef snprintf
#pragma push_macro("vsnprintf")
#undef vsnprintf
#pragma push_macro("snwprintf")
#undef snwprintf
#pragma push_macro("vsnwprintf")
#undef vsnwprintf

#ifdef __cplusplus
extern "C" {
#endif

#define BUFSIZ 512
#define _NFILE _NSTREAM_
#define _NSTREAM 512
#define _IOB_ENTRIES 20
#define EOF (-1)

#ifndef _FILE_DEFINED
struct _iobuf {
#ifdef _UCRT
    void *_Placeholder;
#else
    char *_ptr;
    int _cnt;
    char *_base;
    int _flag;
    int _file;
    int _charbuf;
    int _bufsiz;
    char *_tmpfname;
#endif
};
typedef struct _iobuf FILE;
#define _FILE_DEFINED
#endif
```

通过搜索了解到`#ifndef` `#define` `#endif` 实现了宏定义并且防止了重复定义，貌似可以防止重复编译，好像还 `include` 了另一个头文件

```
# 3 "hello.c"
int main()
{
    printf("hello world!\n");
    return 0;
}
```

Hello.i 最下面依然出现了原来代码内容，加入注释后编译发现注释被删。
再看 hello.s

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("hello world!\n");
6      return 0;
7  }
8
```

对代码处理使其语法错误，然后预处理成.i，使用 `gcc -E hello.c` 让文件内容直接输出

```
# 3 "hello.c"
int main()
{
    printf("hello world!\n")
    return 0;
}
```

错误依然被输出，然后进行编译，gcc -S hello.c

```
D:\Dian>gcc -S hello.c
hello.c: In function 'main':
hello.c:5:29: error: expected ';' before 'return'
   5 |         printf("hello world!\n")
     |                                     ^
     |                                     ;
   6 |     return 0;
     |     ~~~~~
```

报错，说明在编译时才会检查语法，弹出报错消息

```
.file "hello.c"
.text
.def __main; .scl 2; .type 32; .endef
.section .rdata,"dr"
.LC0:
.ascii "hello world!\0"
.text
.globl main
.def main; .scl 2; .type 32; .endef
.seh_proc main
main:
    pushq %rbp
    .seh_pushreg %rbp
    movq %rsp, %rbp
    .seh_setframe %rbp, 0
    subq $32, %rsp
    .seh_stackalloc 32
    .seh_endprologu
    call __main
    leaq .LC0(%rip), %rax
    movq %rax, %rcx
    call puts
    movl $0, %eax
    addq $32, %rsp
    popq %rbp
    ret
    .seh_endproc
.ident "GCC: (x86_64-posix-seh-rev1, Built by MinGW-Builds project) 13.1.0"
.def puts; .scl 2; .type 32; .endef
```

当然，汇编代码我是一点不懂/头秃

接着看 hello.o

