编译器认知实验报告

实验目的

本实验的目的是了解工业界常用的编译器GCC和LLVM，熟悉编译器的安装和使用过程，观察编译器工作过程中生成的中间文件的格式和内容，了解编译器的优化效果，为编译器的学习和构造奠定基础。

实验内容

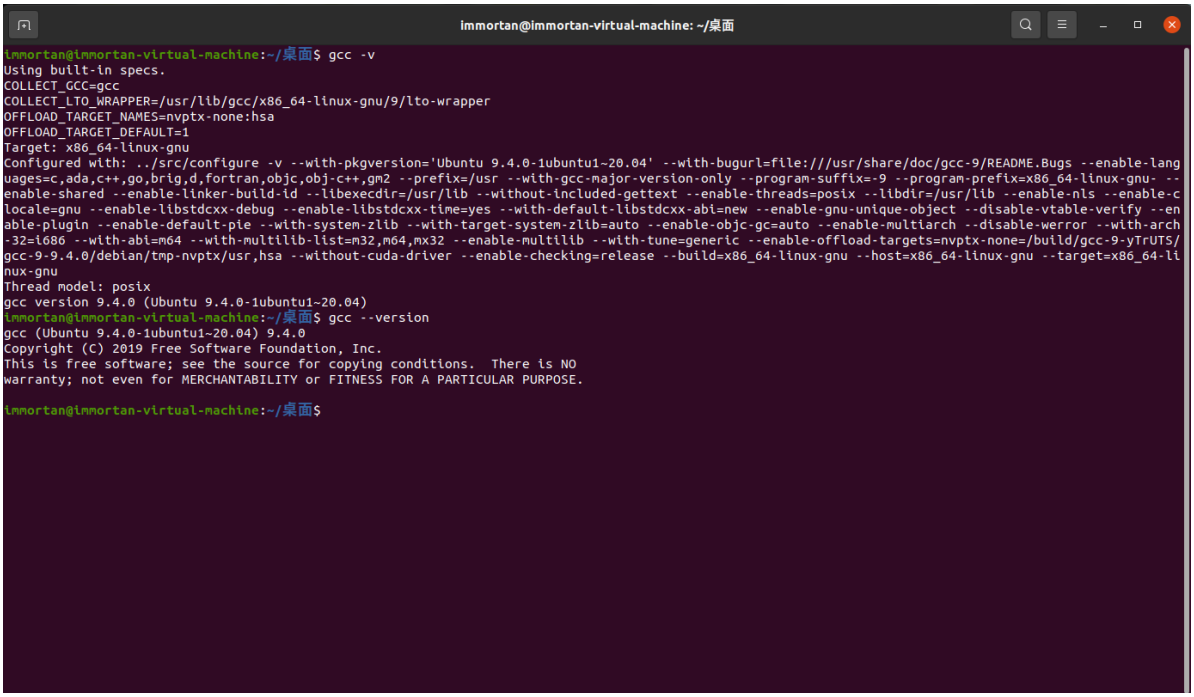
在Linux平台上安装和运行工业界常用的编译器GCC和LLVM，如果系统中没有安装，则需要首先安装编译器，安装完成后编写简单的测试程序，使用编译器编译，并观察中间输出结果。

实现的具体过程和步骤

- 对于GCC 编译器，完成编译器安装和测试程序编写后，按如下步骤完成：
  - 查看编译器的版本
  - 使用编译器编译单个文件
  - 使用编译器编译链接多个文件
  - 查看预处理结果： gcc -E main.c -o main.i
  - 查看语法分析树： gcc -fdump-tree-all main.c
  - 查看中间代码生成结果： gcc -fdump-rtl-all main.c
  - 查看生成的目标代码（汇编代码）： gcc -S main.c -o main.s
- 对于LLVM 编译器，完成编译器安装和测试程序编写后，按如下步骤完成：
  - 查看编译器的版本
  - 使用编译器编译单个文件
  - 使用编译器编译链接多个文件
  - 查看编译流程和阶段：clang -ccc-print-phases main.c -c
  - 查看词法分析结果：clang main.c -Xclang -dump-tokens
  - 查看词法分析结果2：clang main.c -Xclang -dump-raw-tokens
  - 查看语法分析结果：clang main.c -Xclang -ast-dump
  - 查看语法分析结果2：clang main.c -Xclang -ast-view
  - 查看编译优化的结果：clang main.c -S -mllvm -print-after-all
  - 查看生成的目标代码结果：clang main.c –S

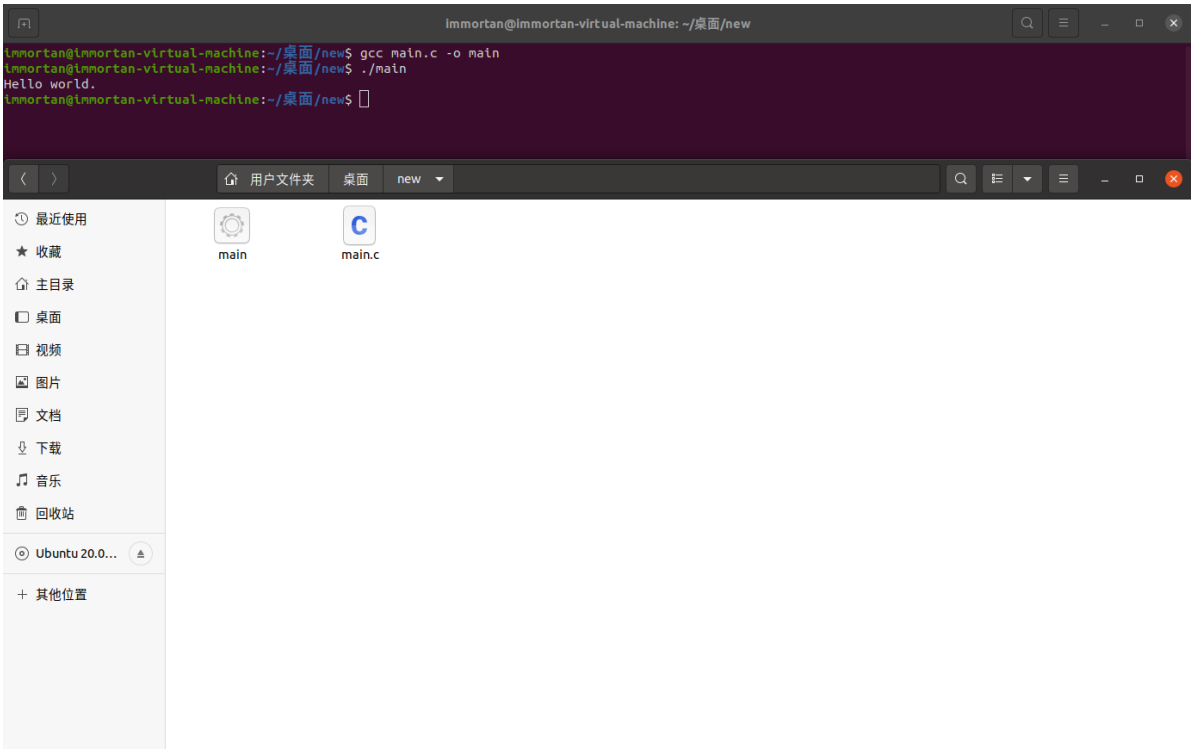
GCC运行结果分析

- 查看编译器版本



- 编译单个文件

```c
#include<stdio.h>

int main() {
    printf("Hello world.\n");
    return 0;
}
```

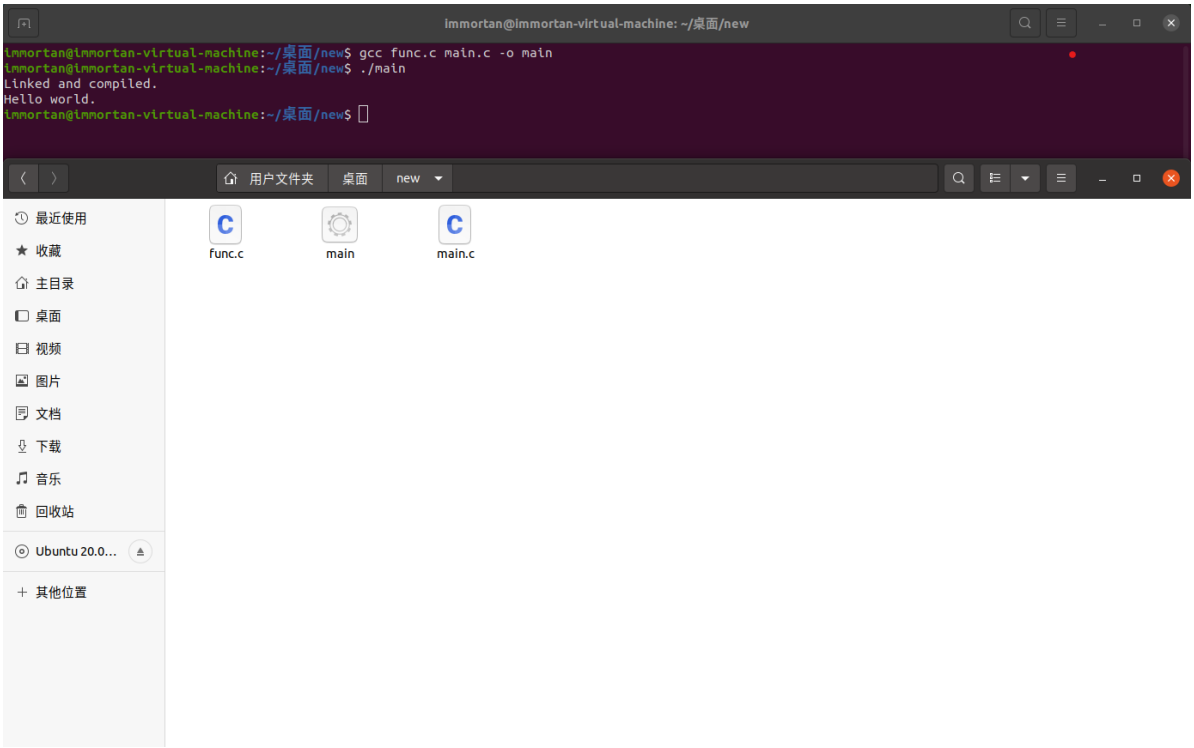


- 编译链接多个文件
    - main.c

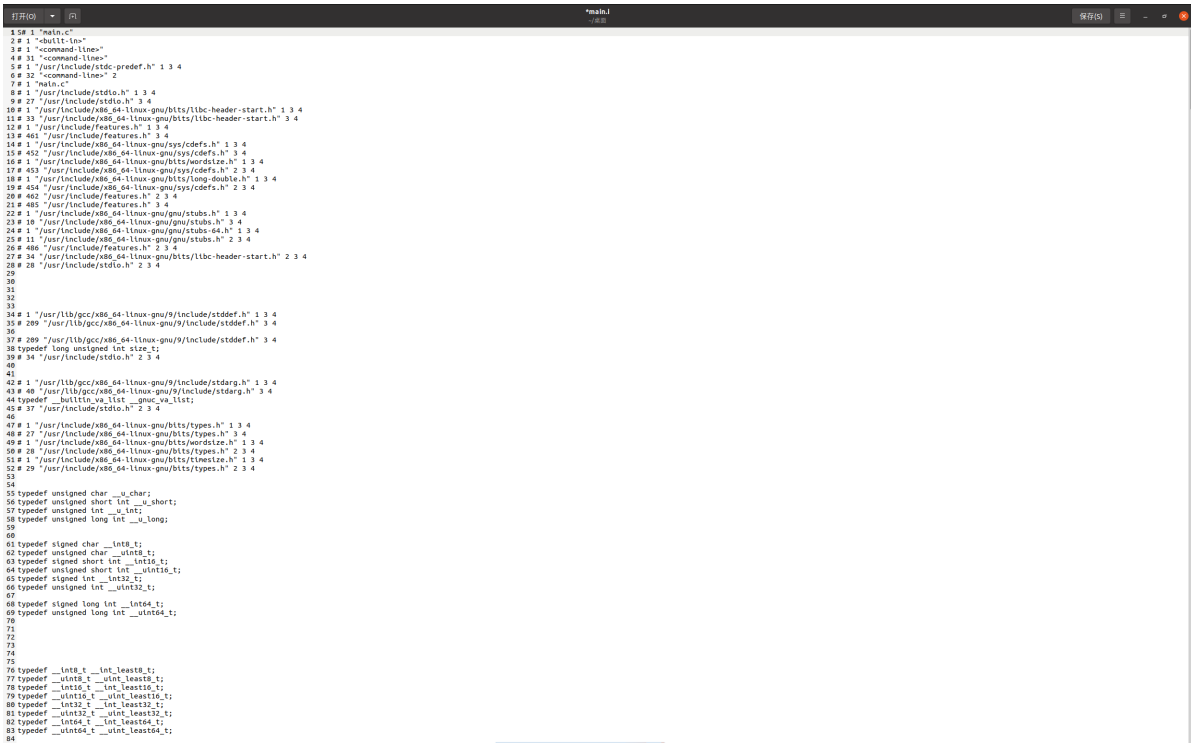
```c
#include<stdio.h>

extern void func();

int main() {
    func();
    printf("Hello world.\n");
    return 0;
}
```

- func.c

```c
#include<stdio.h>

void func() {
    printf("Linked and compiled.\n");
    return;
}
```



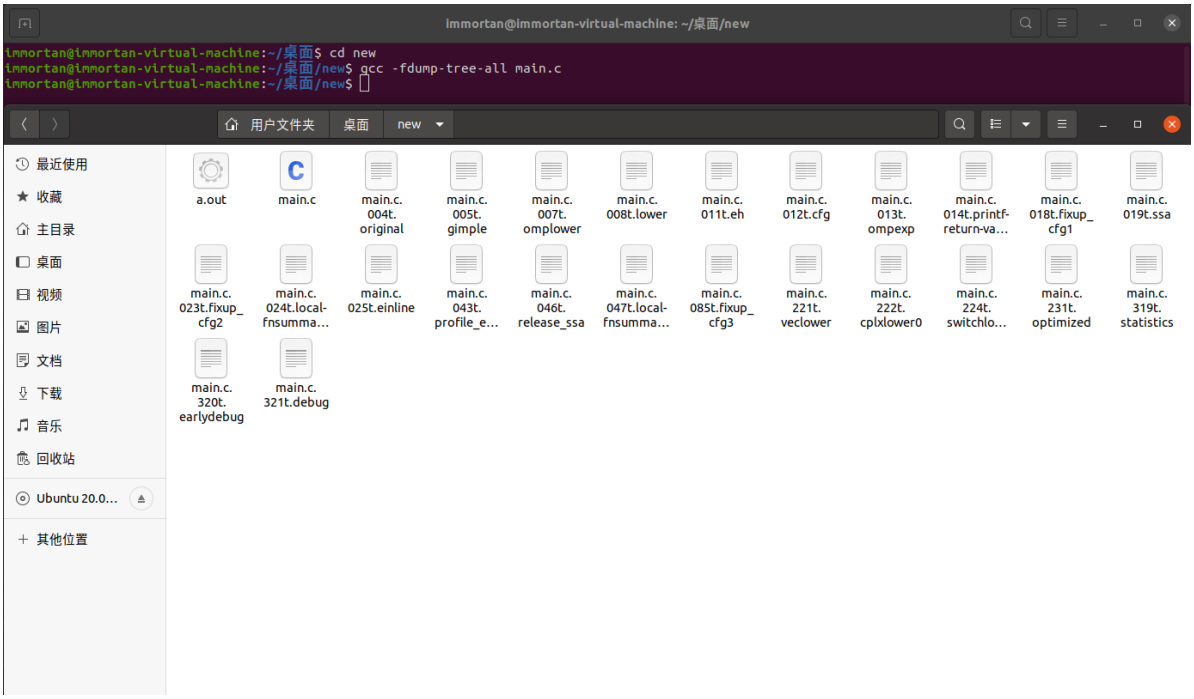
- gcc -E main.c -o main.i
  生成预处理文件，其中包含头文件的展开。



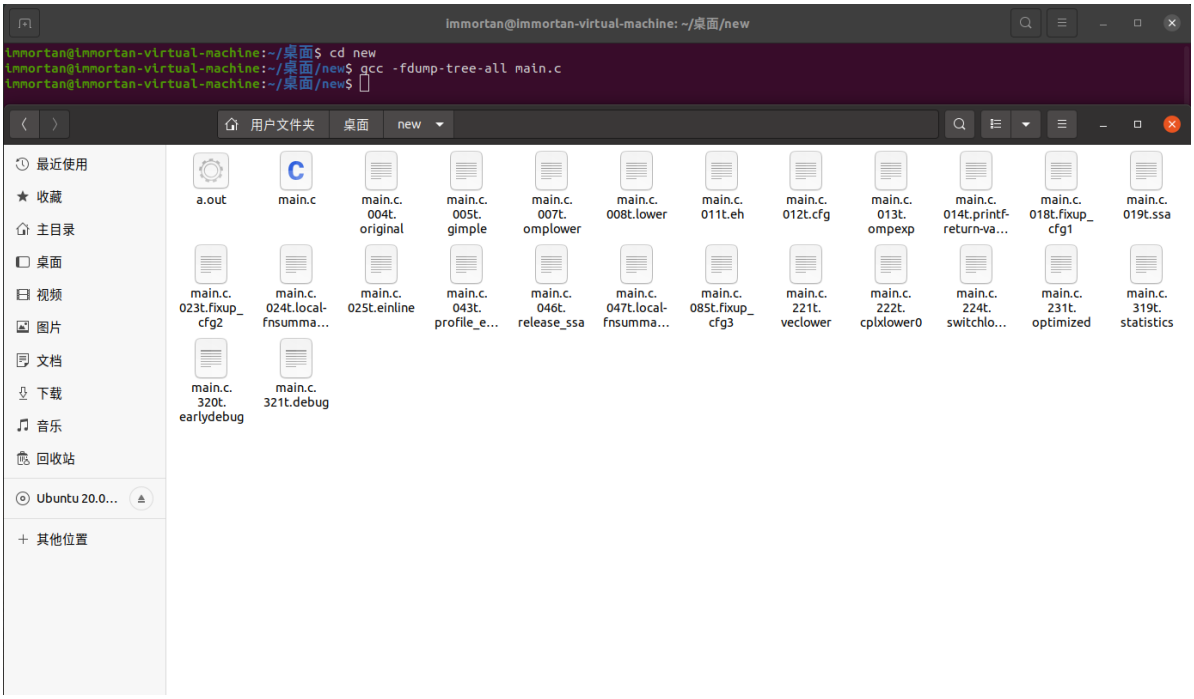

```
# 1 "main.c"
# 1 "<built-in>"
# 1 "<command-line>"
# 31 "<command-line>"
# 1 "/usr/include/stdc-predef.h" 1 3 4
# 32 "<command-line>" 2
# 1 "main.c"
# 1 "/usr/include/stdio.h" 1 3 4
# 27 "/usr/include/stdio.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 1 3 4
# 33 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 3 4
# 1 "/usr/include/features.h" 1 3 4
# 461 "/usr/include/features.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 1 3 4
# 452 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
# 453 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/long-double.h" 1 3 4
# 454 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 2 3 4
# 462 "/usr/include/features.h" 2 3 4
# 485 "/usr/include/features.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 1 3 4
# 10 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/gnu/stubs-64.h" 1 3 4
# 11 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 2 3 4
# 486 "/usr/include/features.h" 2 3 4
# 34 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 2 3 4
# 28 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 1 3 4
# 209 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 3 4
# 209 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 3 4
typedef long unsigned int size_t;
# 34 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stdarg.h" 1 3 4
# 40 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stdarg.h" 3 4
typedef __builtin_va_list __gnuc_va_list;
# 37 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types.h" 1 3 4
# 27 "/usr/include/x86_64-linux-gnu/bits/types.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
# 28 "/usr/include/x86_64-linux-gnu/bits/types.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 1 3 4
# 29 "/usr/include/x86_64-linux-gnu/bits/types.h" 2 3 4


typedef unsigned char __u_char;
typedef unsigned short int __u_short;
typedef unsigned int __u_int;
typedef unsigned long int __u_long;


typedef signed char __int8_t;
typedef unsigned char __uint8_t;
typedef signed short int __int16_t;
typedef unsigned short int __uint16_t;
typedef signed int __int32_t;
typedef unsigned int __uint32_t;

typedef signed long int __int64_t;
typedef unsigned long int __uint64_t;
typedef __int8_t __int_least8_t;
typedef __uint8_t __uint_least8_t;
typedef __int16_t __int_least16_t;
typedef __uint16_t __uint_least16_t;
typedef __int32_t __int_least32_t;
typedef __uint32_t __uint_least32_t;
typedef __int64_t __int_least64_t;
typedef __uint64_t __uint_least64_t;
```


```
typedef __uintmax_t __u_intmax_t;

typedef __intmax_t __intmax_t;
# 142 "/usr/include/x86_64-linux-gnu/bits/types.h" 3 4
typedef unsigned long int __dev_t;
typedef unsigned int __uid_t;
typedef unsigned int __gid_t;
typedef unsigned long int __ino_t;
typedef unsigned long int __ino64_t;
typedef unsigned int __mode_t;
typedef unsigned long int __nlink_t;
typedef long int __off_t;
typedef long int __off64_t;
typedef int __pid_t;
typedef struct { int __val[2]; } __fsid_t;
typedef long int __clock_t;
typedef unsigned long int __rlim_t;
typedef unsigned long int __rlim64_t;
typedef unsigned int __id_t;
typedef long int __time_t;
typedef unsigned int __useconds_t;
typedef long int __suseconds_t;

typedef int __daddr_t;
typedef int __key_t;

typedef int __clockid_t;

typedef void * __timer_t;

typedef long int __blksize_t;

typedef long int __blkcnt_t;
typedef long int __blkcnt64_t;

typedef unsigned long int __fsblkcnt_t;
typedef unsigned long int __fsblkcnt64_t;

typedef unsigned long int __fsfilcnt_t;
typedef unsigned long int __fsfilcnt64_t;

typedef long int __fsword_t;

typedef long int __ssize_t;

typedef long int __syscall_slong_t;

typedef unsigned long int __syscall_ulong_t;

typedef __off64_t __loff_t;
typedef char *__caddr_t;

typedef long int __intptr_t;

typedef unsigned int __socklen_t;
typedef int __sig_atomic_t;
# 38 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types/__fpos_t.h" 1 3 4
# 10 "/usr/include/x86_64-linux-gnu/bits/types/__fpos_t.h" 3 4
typedef struct _G_fpos_t
{
  __off_t __pos;
  __mbstate_t __state;
} __fpos_t;
# 39 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types/__fpos64_t.h" 1 3 4
# 10 "/usr/include/x86_64-linux-gnu/bits/types/__fpos64_t.h" 3 4
typedef struct _G_fpos64_t
{
  __off64_t __pos;
  __mbstate_t __state;
} __fpos64_t;
# 40 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types/__FILE.h" 1 3 4
struct _IO_FILE;
typedef struct _IO_FILE __FILE;
# 41 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types/FILE.h" 1 3 4
struct _IO_FILE;


typedef struct _IO_FILE FILE;
# 42 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types/struct_FILE.h" 1 3 4
# 35 "/usr/include/x86_64-linux-gnu/bits/types/struct_FILE.h" 3 4
struct _IO_FILE;
struct _IO_marker;
struct _IO_codecvt;
struct _IO_wide_data;
# 59 "/usr/include/x86_64-linux-gnu/bits/types/struct_FILE.h" 3 4
typedef void _IO_lock_t;


struct _IO_FILE
{
  int _flags;

  char *_IO_read_ptr;
  char *_IO_read_end;
  char *_IO_read_base;
  char *_IO_write_base;
  char *_IO_write_ptr;
  char *_IO_write_end;
  char *_IO_buf_base;
  char *_IO_buf_end;

  char *_IO_save_base;
  char *_IO_backup_base;
  char *_IO_save_end;

  struct _IO_marker *_markers;

  struct _IO_FILE *_chain;

  int _fileno;

  int _flags2;

  __off_t _old_offset;


  unsigned short _cur_column;
  signed char _vtable_offset;
  char _shortbuf[1];

  _IO_lock_t *_lock;
# 107 "/usr/include/x86_64-linux-gnu/bits/types/struct_FILE.h" 3 4
  __off64_t _offset;


  struct _IO_codecvt *_codecvt;
  struct _IO_wide_data *_wide_data;

  struct _IO_FILE *_freeres_list;

  void *_freeres_buf;

  size_t __pad5;


  int _mode;

  char _unused2[15 * sizeof (int) - 4 * sizeof (void *) - sizeof (size_t)];
};
# 43 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types/cookie_io_functions_t.h" 1 3 4
# 27 "/usr/include/x86_64-linux-gnu/bits/types/cookie_io_functions_t.h" 3 4
typedef __ssize_t cookie_read_function_t (void *__cookie, char *__buf,
                                          size_t __nbytes);


typedef __ssize_t cookie_write_function_t (void *__cookie, const char *__buf,
                                           size_t __nbytes);


typedef int cookie_seek_function_t (void *__cookie, __off64_t *__pos, int __w);


typedef int cookie_close_function_t (void *__cookie);


typedef struct _IO_cookie_io_functions_t
{
  cookie_read_function_t *read;
  cookie_write_function_t *write;
  cookie_seek_function_t *seek;
  cookie_close_function_t *close;
} cookie_io_functions_t;
# 44 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/stdio_lim.h" 1 3 4
# 45 "/usr/include/stdio.h" 2 3 4


typedef __gnuc_va_list va_list;
# 78 "/usr/include/stdio.h" 3 4
typedef __fpos_t fpos_t;
# 90 "/usr/include/stdio.h" 3 4
typedef __fpos64_t fpos64_t;
# 133 "/usr/include/stdio.h" 3 4
extern FILE *stdin;
extern FILE *stdout;
extern FILE *stderr;


extern int remove (const char *__filename) __attribute__ ((__nothrow__ , __leaf__));

extern int rename (const char *__old, const char *__new) __attribute__ ((__nothrow__ , __leaf__));


extern int renameat (int __oldfd, const char *__old, int __newfd,
                     const char *__new) __attribute__ ((__nothrow__ , __leaf__));
# 175 "/usr/include/stdio.h" 3 4
extern FILE *tmpfile (void) ;
# 189 "/usr/include/stdio.h" 3 4
extern char *tmpnam (char *__s) __attribute__ ((__nothrow__ , __leaf__)) ;


extern char *tmpnam_r (char *__s) __attribute__ ((__nothrow__ , __leaf__)) ;
# 206 "/usr/include/stdio.h" 3 4
extern char *tempnam (const char *__dir, const char *__pfx)
   __attribute__ ((__nothrow__ , __leaf__)) __attribute__ ((__malloc__)) ;
# 217 "/usr/include/stdio.h" 3 4
extern int fclose (FILE *__stream);
# 227 "/usr/include/stdio.h" 3 4
extern int fflush (FILE *__stream);
# 241 "/usr/include/stdio.h" 3 4
extern int fflush_unlocked (FILE *__stream);
# 251 "/usr/include/stdio.h" 3 4
extern int fcloseall (void);
# 261 "/usr/include/stdio.h" 3 4
extern FILE *fopen (const char *__restrict __filename,
                    const char *__restrict __modes) ;


extern FILE *freopen (const char *__restrict __filename,
                      const char *__restrict __modes,
                      FILE *__restrict __stream) ;
# 286 "/usr/include/stdio.h" 3 4
extern FILE *fopen64 (const char *__restrict __filename,
                      const char *__restrict __modes) ;
extern FILE *freopen64 (const char *__restrict __filename,
                        const char *__restrict __modes,
                        FILE *__restrict __stream) ;


extern FILE *fdopen (int __fd, const char *__modes) __attribute__ ((__nothrow__ , __leaf__)) ;


extern FILE *fopencookie (void *__restrict __magic_cookie,
                          const char *__restrict __modes,
                          cookie_io_functions_t __io_funcs) __attribute__ ((__nothrow__ , __leaf__)) ;


extern FILE *fmemopen (void *__s, size_t __len, const char *__modes)
  __attribute__ ((__nothrow__ , __leaf__)) ;


extern FILE *open_memstream (char **__bufloc, size_t *__sizeloc) __attribute__ ((__nothrow__ , __leaf__)) ;
# 325 "/usr/include/stdio.h" 3 4
extern void setbuf (FILE *__restrict __stream, char *__restrict __buf) __attribute__ ((__nothrow__ , __leaf__));


extern int setvbuf (FILE *__restrict __stream, char *__restrict __buf,
                    int __modes, size_t __n) __attribute__ ((__nothrow__ , __leaf__));


extern void setbuffer (FILE *__restrict __stream, char *__restrict __buf,
                       size_t __size) __attribute__ ((__nothrow__ , __leaf__));


extern void setlinebuf (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));
# 353 "/usr/include/stdio.h" 3 4
extern int fprintf (FILE *__restrict __stream,
                    const char *__restrict __format, ...);


extern int printf (const char *__restrict __format, ...);

extern int sprintf (char *__restrict __s,
                    const char *__restrict __format, ...) __attribute__ ((__nothrow__));


extern int vfprintf (FILE *__restrict __s, const char *__restrict __format,
                     __gnuc_va_list __arg);


extern int vprintf (const char *__restrict __format, __gnuc_va_list __arg);

extern int vsprintf (char *__restrict __s, const char *__restrict __format,
                     __gnuc_va_list __arg) __attribute__ ((__nothrow__));


extern int snprintf (char *__restrict __s, size_t __maxlen,
                     const char *__restrict __format, ...)
     __attribute__ ((__nothrow__)) __attribute__ ((__format__ (__printf__, 3, 4)));

extern int vsnprintf (char *__restrict __s, size_t __maxlen,
                      const char *__restrict __format, __gnuc_va_list __arg)
     __attribute__ ((__nothrow__)) __attribute__ ((__format__ (__printf__, 3, 0)));


extern int vdprintf (int __fd, const char *__restrict __fmt,
                     __gnuc_va_list __arg)
     __attribute__ ((__format__ (__printf__, 2, 0)));
extern int dprintf (int __fd, const char *__restrict __fmt, ...)
     __attribute__ ((__format__ (__printf__, 2, 3)));
# 417 "/usr/include/stdio.h" 3 4
extern int fscanf (FILE *__restrict __stream,
                   const char *__restrict __format, ...) ;


extern int scanf (const char *__restrict __format, ...) ;

extern int sscanf (const char *__restrict __s,
                   const char *__restrict __format, ...) __attribute__ ((__nothrow__ , __leaf__));
# 440 "/usr/include/stdio.h" 3 4
extern int vfscanf (FILE *__restrict __s, const char *__restrict __format,
                    __gnuc_va_list __arg)
     __attribute__ ((__format__ (__scanf__, 2, 0))) ;


extern int vscanf (const char *__restrict __format, __gnuc_va_list __arg)
     __attribute__ ((__format__ (__scanf__, 1, 0))) ;


extern int vsscanf (const char *__restrict __s,
                    const char *__restrict __format, __gnuc_va_list __arg)
     __attribute__ ((__nothrow__ , __leaf__)) __attribute__ ((__format__ (__scanf__, 2, 0)));
# 469 "/usr/include/stdio.h" 3 4
extern int fgetc (FILE *__stream);
extern int getc (FILE *__stream);


extern int getchar (void);
# 485 "/usr/include/stdio.h" 3 4
extern int getc_unlocked (FILE *__stream);
extern int getchar_unlocked (void);
# 496 "/usr/include/stdio.h" 3 4
extern int fgetc_unlocked (FILE *__stream);
# 508 "/usr/include/stdio.h" 3 4
extern int fputc (int __c, FILE *__stream);
extern int putc (int __c, FILE *__stream);


extern int putchar (int __c);
# 524 "/usr/include/stdio.h" 3 4
extern int fputc_unlocked (int __c, FILE *__stream);
# 534 "/usr/include/stdio.h" 3 4
extern int putc_unlocked (int __c, FILE *__stream);
extern int putchar_unlocked (int __c);
# 545 "/usr/include/stdio.h" 3 4
extern int getw (FILE *__stream);


extern int putw (int __w, FILE *__stream);
# 558 "/usr/include/stdio.h" 3 4
extern char *fgets (char *__restrict __s, int __n, FILE *__restrict __stream)
     ;
# 584 "/usr/include/stdio.h" 3 4
extern __ssize_t __getdelim (char **__restrict __lineptr,
                             size_t *__restrict __n, int __delimiter,
                             FILE *__restrict __stream) ;
extern __ssize_t getdelim (char **__restrict __lineptr,
                           size_t *__restrict __n, int __delimiter,
                           FILE *__restrict __stream) ;


extern __ssize_t getline (char **__restrict __lineptr,
                          size_t *__restrict __n,
                          FILE *__restrict __stream) ;
# 609 "/usr/include/stdio.h" 3 4
extern int fputs (const char *__restrict __s, FILE *__restrict __stream);


extern int puts (const char *__s);


extern int ungetc (int __c, FILE *__stream);


extern size_t fread (void *__restrict __ptr, size_t __size,
                     size_t __n, FILE *__restrict __stream) ;


extern size_t fwrite (const void *__restrict __ptr, size_t __size,
                      size_t __n, FILE *__restrict __s);
# 650 "/usr/include/stdio.h" 3 4
extern size_t fread_unlocked (void *__restrict __ptr, size_t __size,
                              size_t __n, FILE *__restrict __stream) ;
extern size_t fwrite_unlocked (const void *__restrict __ptr, size_t __size,
                               size_t __n, FILE *__restrict __stream);
# 668 "/usr/include/stdio.h" 3 4
extern int fseek (FILE *__stream, long int __off, int __whence);


extern long int ftell (FILE *__stream) ;


extern void rewind (FILE *__stream);
# 707 "/usr/include/stdio.h" 3 4
extern int fseeko (FILE *__stream, __off_t __off, int __whence);


extern __off_t ftello (FILE *__stream) ;
# 731 "/usr/include/stdio.h" 3 4
extern int fgetpos (FILE *__restrict __stream, fpos_t *__restrict __pos);


extern int fsetpos (FILE *__stream, const fpos_t *__pos);
# 757 "/usr/include/stdio.h" 3 4
extern void clearerr (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));

extern int feof (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;

extern int ferror (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;


extern void clearerr_unlocked (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));
extern int feof_unlocked (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;
extern int ferror_unlocked (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;
# 785 "/usr/include/stdio.h" 3 4
extern void perror (const char *__s);
# 802 "/usr/include/stdio.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/sys_errlist.h" 1 3 4
# 26 "/usr/include/x86_64-linux-gnu/bits/sys_errlist.h" 3 4
extern int sys_nerr;
extern const char *const sys_errlist[];
# 802 "/usr/include/stdio.h" 2 3 4


extern int fileno (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;


extern int fileno_unlocked (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;
# 820 "/usr/include/stdio.h" 3 4
extern FILE *popen (const char *__command, const char *__modes) ;


extern int pclose (FILE *__stream);


extern char *ctermid (char *__s) __attribute__ ((__nothrow__ , __leaf__));
# 840 "/usr/include/stdio.h" 3 4
extern void flockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));


extern int ftrylockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;


extern void funlockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));
# 858 "/usr/include/stdio.h" 3 4
extern int __uflow (FILE *);
extern int __overflow (FILE *, int);
# 873 "/usr/include/stdio.h" 3 4

# 2 "main.c" 2


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

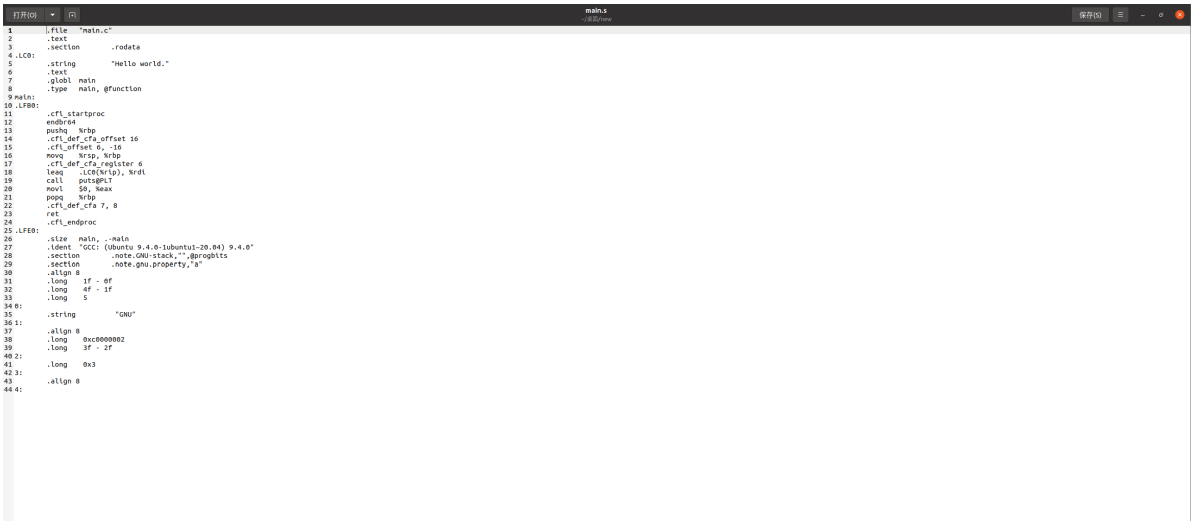
- gcc -fdump-tree-all main.c
  生成了较多文件，其中一部分为空白文件。



- gcc -fdump-rtl-all main.c
  生成中间代码。在gcc中，中间代码用指令序列表示。

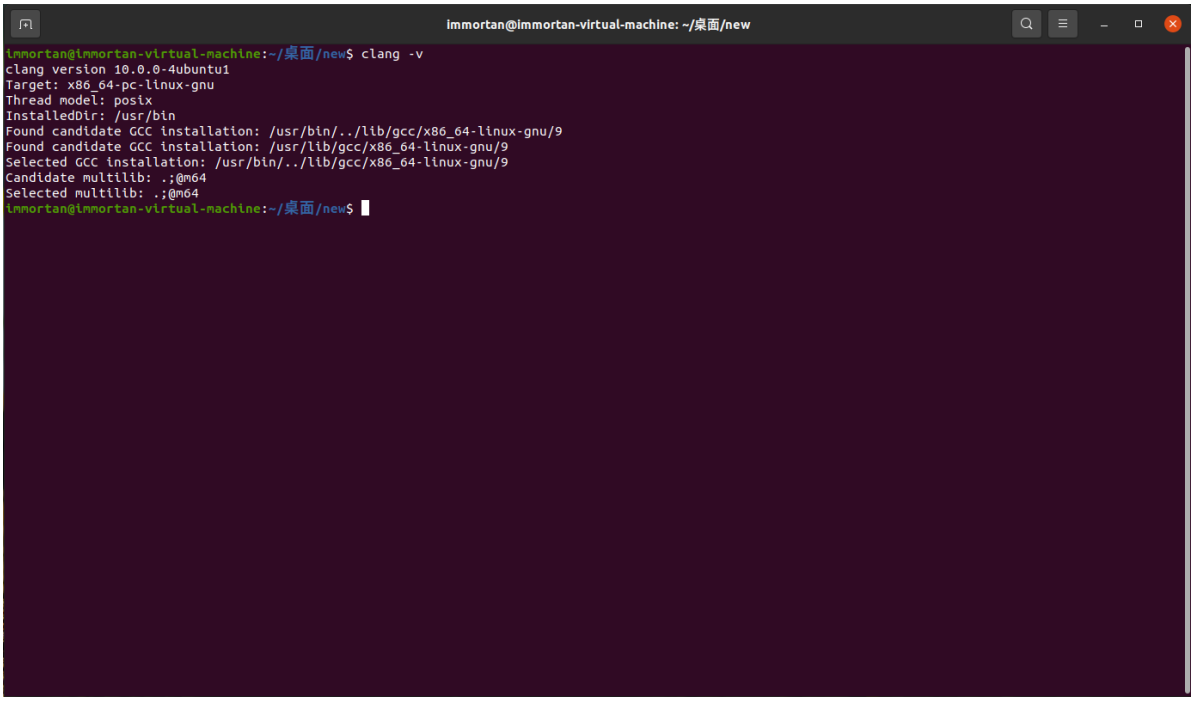


- gcc -S main.c -o main.s
  生成汇编代码



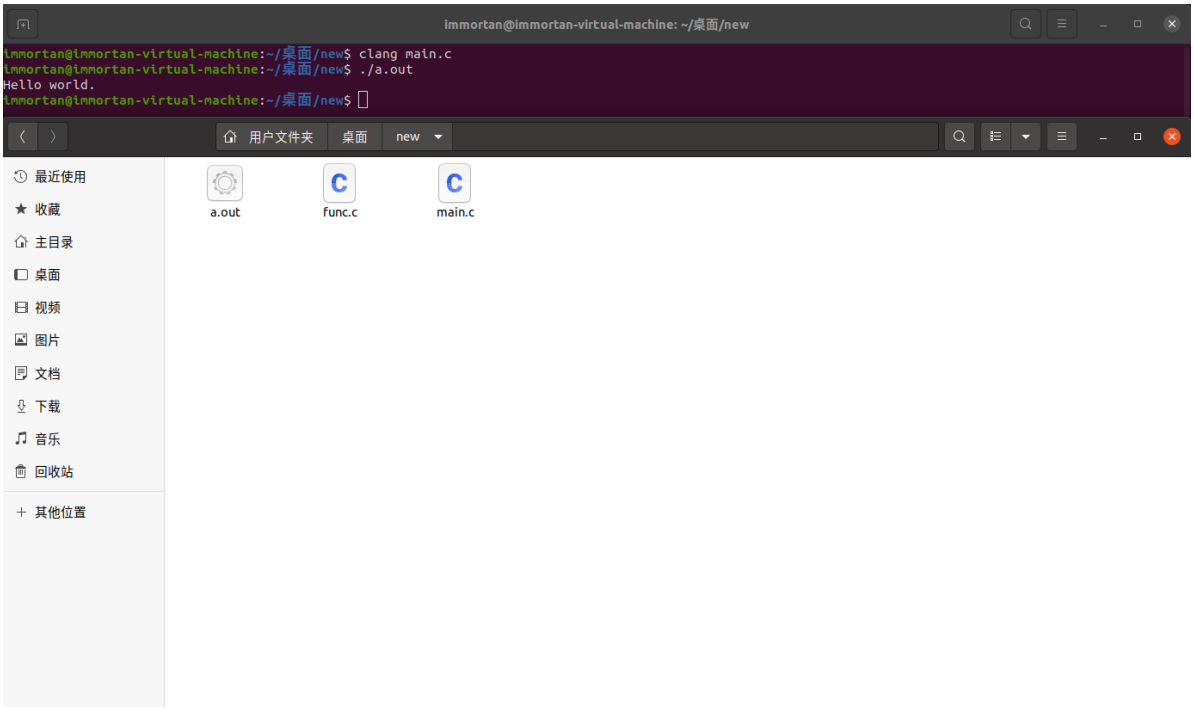
LLVM运行结果分析

- clang -v
  获取clang版本信息和配置信息。



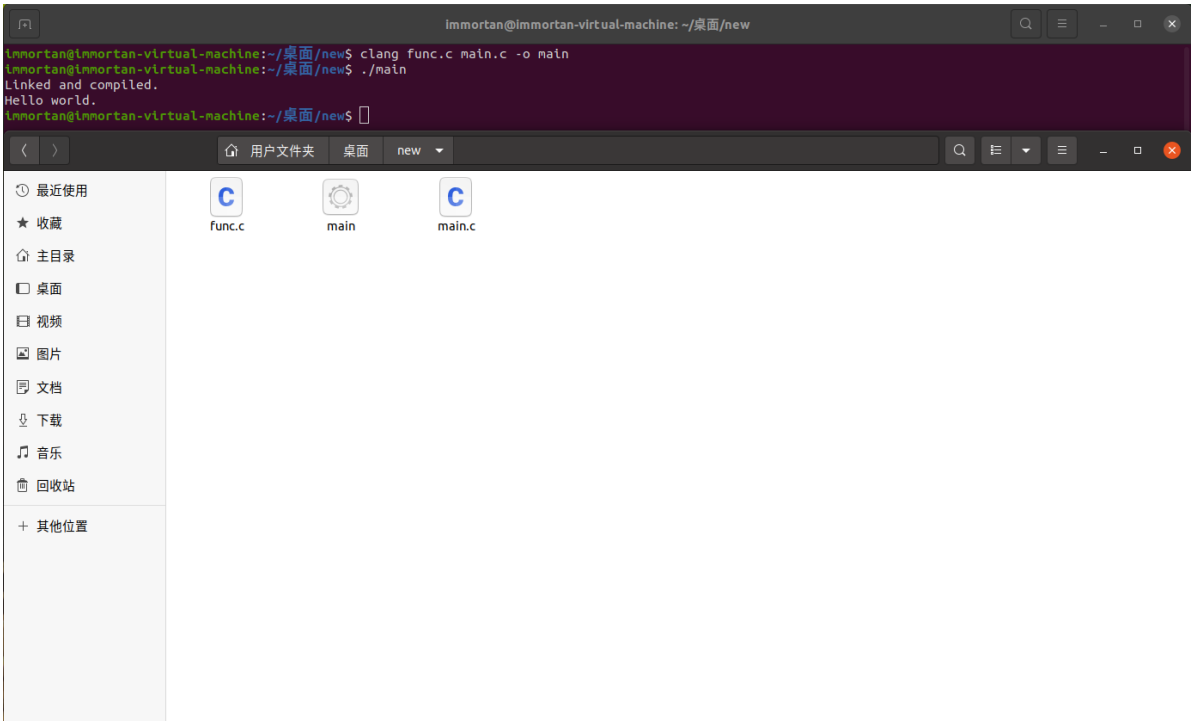
- 编译单个文件

  源码同gcc

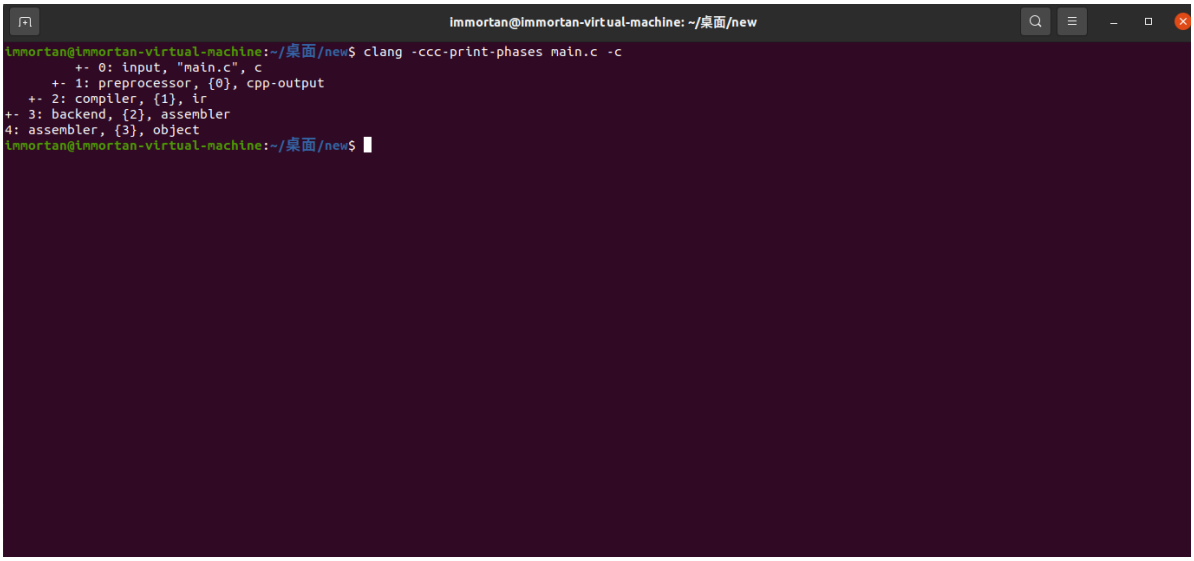


- 编译链接多个文件

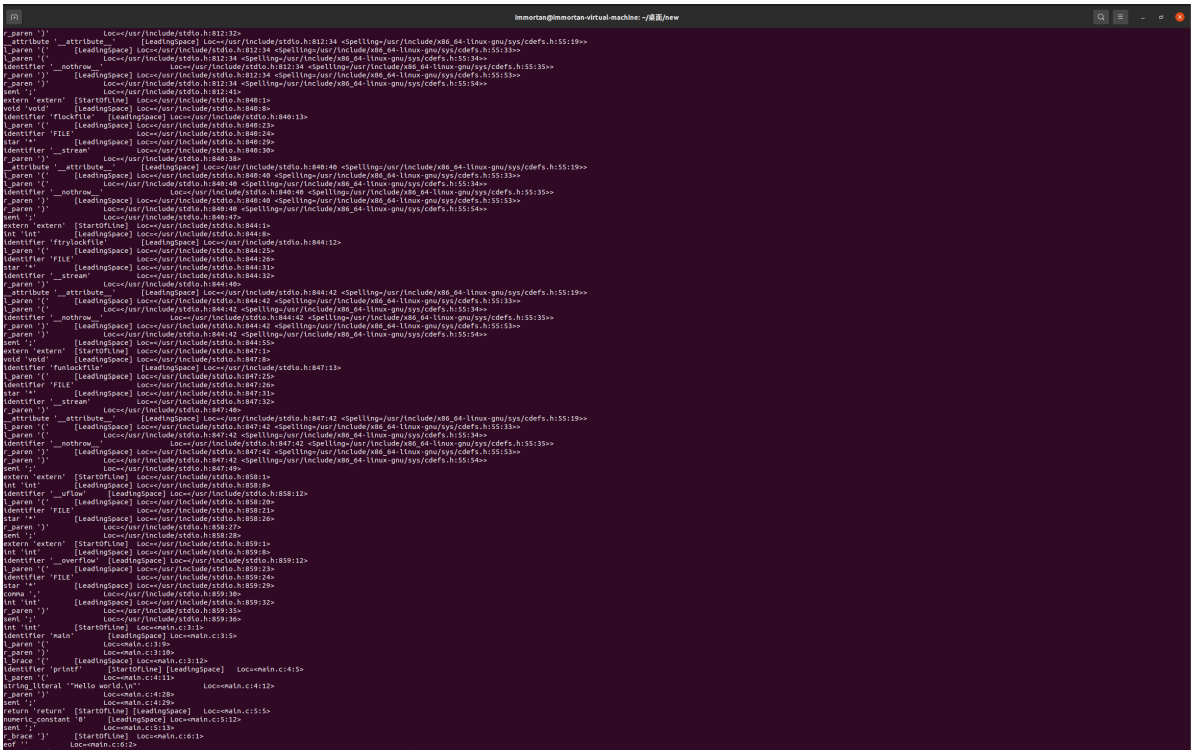
  源码同gcc



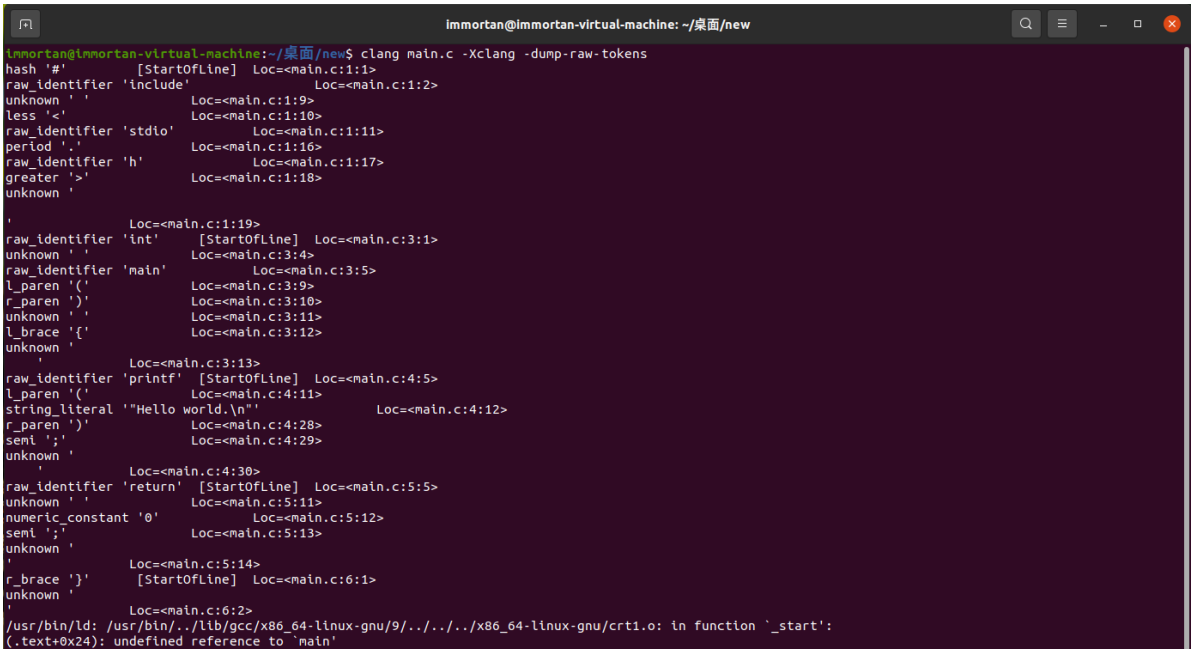
- clang -ccc-print-phases main.c -c



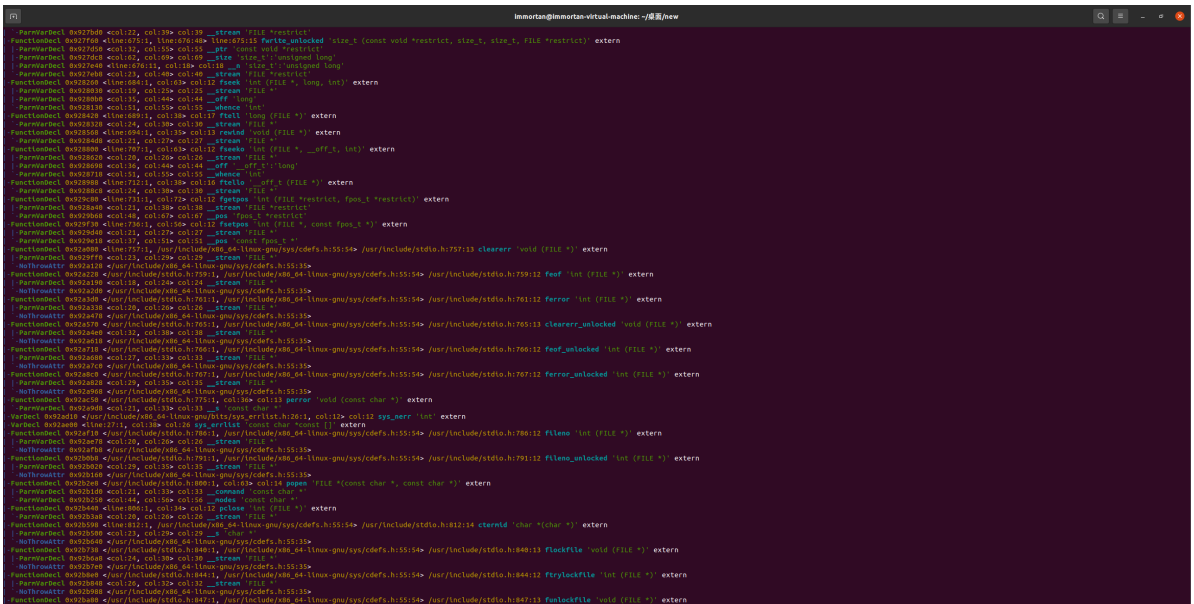
- clang main.c -Xclang -dump-tokens



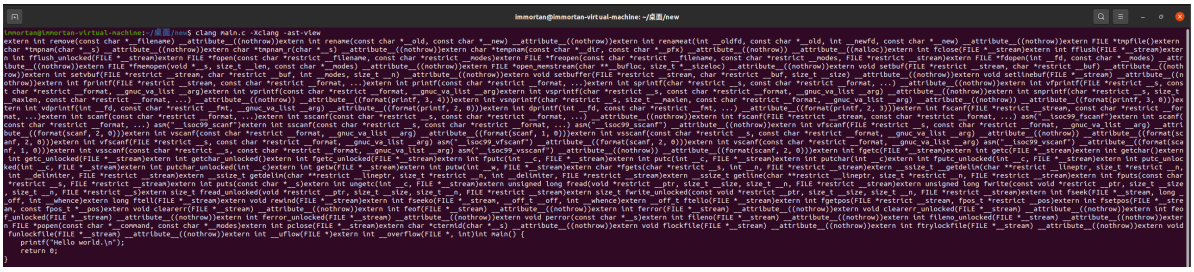
- clang main.c -Xclang -dump-raw-tokens



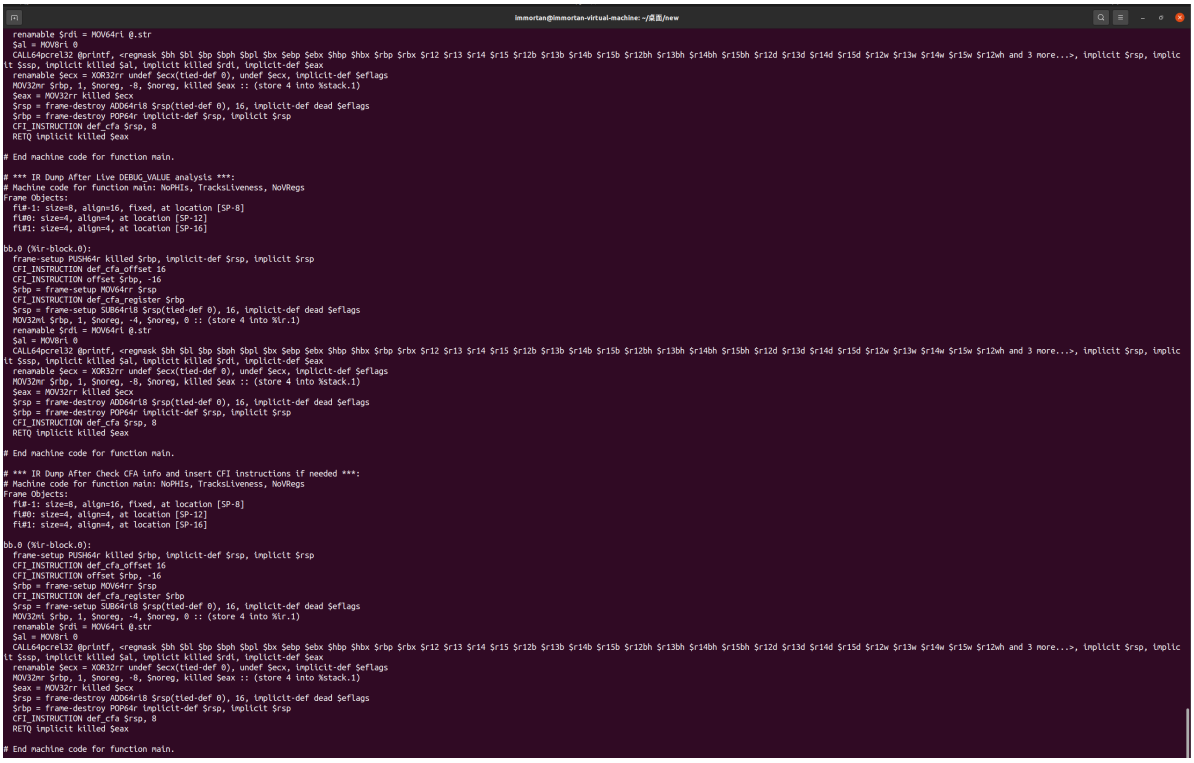
- clang main.c -Xclang -ast-dump



- clang main.c -Xclang -ast-view

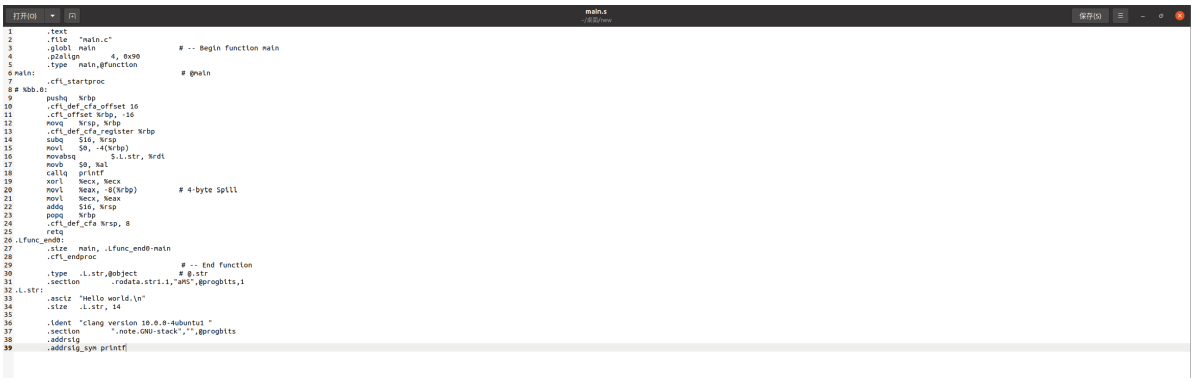


- clang main.c -S -mllvm -print-after-all




```
  renamable $rdi = MOV64ri @.str
  $al = MOV8ri 0
  CALL64pcrel32 @printf, <regmask $bh $bl $bp $bph $bpl $bx $ebp $ebx $hbp $hbx $rbp $rbx $r12 $r13 $r14 $r15 $r12b $r13b $r14b $r15b $r12bh $r13bh $r14bh $r15bh $r12d $r13d $r14d $r15d $r12w $r13w $r14w $r15w $r12wh and 3 more...>, implicit $rsp, implic
lt $ssp, implicit killed $al, implicit killed $rdi, implicit-def $eax
  renamable $ecx = XOR32rr undef $ecx(tied-def 0), undef $ecx, implicit-def $eflags
  MOV32mr $rbp, 1, $noreg, -8, $noreg, killed $eax :: (store 4 into %stack.1)
  $eax = MOV32rr killed $ecx
  $rsp = frame-destroy ADD64ri8 $rsp(tied-def 0), 16, implicit-def dead $eflags
  $rbp = frame-destroy POP64r implicit-def $rsp, implicit $rsp
  CFI_INSTRUCTION def_cfa $rsp, 8
  RETQ implicit killed $eax

# End machine code for function main.

# *** IR Dump After Live DEBUG_VALUE analysis ***:
# Machine code for function main: NoPHIs, TracksLiveness, NoVRegs
Frame Objects:
  fi#-1: size=8, align=16, fixed, at location [SP-8]
  fi#0: size=4, align=4, at location [SP-12]
  fi#1: size=4, align=4, at location [SP-16]

bb.0 (%ir-block.0):
  frame-setup PUSH64r killed $rbp, implicit-def $rsp, implicit $rsp
  CFI_INSTRUCTION def_cfa_offset 16
  CFI_INSTRUCTION offset $rbp, -16
  $rbp = frame-setup MOV64rr $rsp
  CFI_INSTRUCTION def_cfa_register $rbp
  $rsp = frame-setup SUB64ri8 $rsp(tied-def 0), 16, implicit-def dead $eflags
  MOV32mi $rbp, 1, $noreg, -4, $noreg, 0 :: (store 4 into %ir.1)
  renamable $rdi = MOV64ri @.str
  $al = MOV8ri 0
  CALL64pcrel32 @printf, <regmask $bh $bl $bp $bph $bpl $bx $ebp $ebx $hbp $hbx $rbp $rbx $r12 $r13 $r14 $r15 $r12b $r13b $r14b $r15b $r12bh $r13bh $r14bh $r15bh $r12d $r13d $r14d $r15d $r12w $r13w $r14w $r15w $r12wh and 3 more...>, implicit $rsp, implic
lt $ssp, implicit killed $al, implicit killed $rdi, implicit-def $eax
  renamable $ecx = XOR32rr undef $ecx(tied-def 0), undef $ecx, implicit-def $eflags
  MOV32mr $rbp, 1, $noreg, -8, $noreg, killed $eax :: (store 4 into %stack.1)
  $eax = MOV32rr killed $ecx
  $rsp = frame-destroy ADD64ri8 $rsp(tied-def 0), 16, implicit-def dead $eflags
  $rbp = frame-destroy POP64r implicit-def $rsp, implicit $rsp
  CFI_INSTRUCTION def_cfa $rsp, 8
  RETQ implicit killed $eax

# End machine code for function main.

# *** IR Dump After Check CFA info and insert CFI instructions if needed ***:
# Machine code for function main: NoPHIs, TracksLiveness, NoVRegs
Frame Objects:
  fi#-1: size=8, align=16, fixed, at location [SP-8]
  fi#0: size=4, align=4, at location [SP-12]
  fi#1: size=4, align=4, at location [SP-16]

bb.0 (%ir-block.0):
  frame-setup PUSH64r killed $rbp, implicit-def $rsp, implicit $rsp
  CFI_INSTRUCTION def_cfa_offset 16
  CFI_INSTRUCTION offset $rbp, -16
  $rbp = frame-setup MOV64rr $rsp
  CFI_INSTRUCTION def_cfa_register $rbp
  $rsp = frame-setup SUB64ri8 $rsp(tied-def 0), 16, implicit-def dead $eflags
  MOV32mi $rbp, 1, $noreg, -4, $noreg, 0 :: (store 4 into %ir.1)
  renamable $rdi = MOV64ri @.str
  $al = MOV8ri 0
  CALL64pcrel32 @printf, <regmask $bh $bl $bp $bph $bpl $bx $ebp $ebx $hbp $hbx $rbp $rbx $r12 $r13 $r14 $r15 $r12b $r13b $r14b $r15b $r12bh $r13bh $r14bh $r15bh $r12d $r13d $r14d $r15d $r12w $r13w $r14w $r15w $r12wh and 3 more...>, implicit $rsp, implic
lt $ssp, implicit killed $al, implicit killed $rdi, implicit-def $eax
  renamable $ecx = XOR32rr undef $ecx(tied-def 0), undef $ecx, implicit-def $eflags
  MOV32mr $rbp, 1, $noreg, -8, $noreg, killed $eax :: (store 4 into %stack.1)
  $eax = MOV32rr killed $ecx
  $rsp = frame-destroy ADD64ri8 $rsp(tied-def 0), 16, implicit-def dead $eflags
  $rbp = frame-destroy POP64r implicit-def $rsp, implicit $rsp
  CFI_INSTRUCTION def_cfa $rsp, 8
  RETQ implicit killed $eax

# End machine code for function main.
```

- clang main.c –S




```
 1        .text
 2        .file   "main.c"
 3        .globl  main                    # -- Begin function main
 4        .p2align        4, 0x90
 5        .type   main,@function
 6 main:                                  # @main
 7        .cfi_startproc
 8 # %bb.0:
 9        pushq   %rbp
10        .cfi_def_cfa_offset 16
11        .cfi_offset %rbp, -16
12        movq    %rsp, %rbp
13        .cfi_def_cfa_register %rbp
14        subq    $16, %rsp
15        movl    $0, -4(%rbp)
16        movabsq $.L.str, %rdi
17        movb    $0, %al
18        callq   printf
19        xorl    %ecx, %ecx
20        movl    %eax, -8(%rbp)          # 4-byte Spill
21        movl    %ecx, %eax
22        addq    $16, %rsp
23        popq    %rbp
24        .cfi_def_cfa %rsp, 8
25        retq
26 .Lfunc_end0:
27        .size   main, .Lfunc_end0-main
28        .cfi_endproc
29                                        # -- End function
30        .type   .L.str,@object          # @.str
31        .section        .rodata.str1.1,"aMS",@progbits,1
32 .L.str:
33        .asciz  "Hello world.\n"
34        .size   .L.str, 14
35
36        .ident  "clang version 10.0.0-4ubuntu1 "
37        .section        ".note.GNU-stack","",@progbits
38        .addrsig
39        .addrsig_sym printf
```

实验心得体会

初步了解了GCC和LLVM的编译过程，对词法分析、语法分析、中间代码和汇编有了具体的认识。