

河南工业大学 Linux 基础与应用 实验报告

专业班级: 软件 1305 班 学号: 201316920311 姓名: 田劲锋 指导老师: 赵晨阳 评分: _____

实验题目: Linux 下 C 语言编程

实验目的: (1) 理解 GCC; (2) 理解程序维护工具 make。

实验内容:

1. 简单的编写一个 C 语言程序, 使用 GCC 编译该程序, 要求按照以下步骤进行: 预处理、编译、汇编、连接, 并查看中间结果。
2. 编写两个 C 语言程序, 其中一个程序的编译需要另一个程序的编译完成, 试写出一个 makefile 文件, 并测试。

实验步骤:

本次实验在 Mac OS X Yosemite 上运行。

脚本的运行结果如图??。

```
1. $ ls
    aplusb.c
    $ gcc -E aplusb.c -o aplusb.i
    $ ls
    aplusb.c aplusb.i
    $ gcc -S aplusb.c -o aplusb.s
    $ ls
    aplusb.c aplusb.i aplusb.s
    $ gcc -c aplusb.s -o aplusb.o
    $ ls
    aplusb.c aplusb.i aplusb.o aplusb.s
    $ gcc aplusb.o -o aplusb
    $ ls
    aplusb  aplusb.c aplusb.i aplusb.o aplusb.s
    $ ./aplusb
    12 34
    46
```

Listing 1: aplusb.c

```
1  #include <stdio.h>
2
3  int main(int argc, char* argv[])
4  {
5      int a, b;
6      scanf("%d%d", &a, &b);
7      int c = a + b;
8      printf("%d\n", c);
9      return 0;
10 }
```

2.

```

1 # 1 "apusb.c"
2 # 1 "<built-in>" 1
3 # 1 "<built-in>" 3
4 # 321 "<built-in>" 3
5 # 1 "<command line>" 1
6 # 1 "<built-in>" 2
7 # 1 "apusb.c" 2
8 # 1 "/usr/include/stdio.h" 1 3 4
9 # 64 "/usr/include/stdio.h" 3 4
10 # 1 "/usr/include/sys/cdefs.h" 1 3 4
11 # 506 "/usr/include/sys/cdefs.h" 3 4
12 # 1 "/usr/include/sys/_symbol_aliasing.h" 1 3 4
13 # 507 "/usr/include/sys/cdefs.h" 2 3 4
14 # 572 "/usr/include/sys/cdefs.h" 3 4
15 # 1 "/usr/include/sys/_posix_availability.h" 1 3 4
16 # 573 "/usr/include/sys/cdefs.h" 2 3 4
17 # 65 "/usr/include/stdio.h" 2 3 4
18 # 1 "/usr/include/Availability.h" 1 3 4
19 # 153 "/usr/include/Availability.h" 3 4
20 # 1 "/usr/include/AvailabilityInternal.h" 1 3 4
21 # 154 "/usr/include/Availability.h" 2 3 4
22 # 66 "/usr/include/stdio.h" 2 3 4
23
24 # 1 "/usr/include/_types.h" 1 3 4
25 # 27 "/usr/include/_types.h" 3 4
26 # 1 "/usr/include/sys/_types.h" 1 3 4
27 # 33 "/usr/include/sys/_types.h" 3 4
28 # 1 "/usr/include/machine/_types.h" 1 3 4
29 # 32 "/usr/include/machine/_types.h" 3 4
30 # 1 "/usr/include/i386/_types.h" 1 3 4
31 # 37 "/usr/include/i386/_types.h" 3 4
32 typedef signed char __int8_t;
33
34
35
36 typedef unsigned char __uint8_t;
37 typedef short __int16_t;
38 typedef unsigned short __uint16_t;
39 typedef int __int32_t;
40 typedef unsigned int __uint32_t;
41 typedef long long __int64_t;
42 typedef unsigned long long __uint64_t;
43
44 typedef long __darwin_intptr_t;
45 typedef unsigned int __darwin_natural_t;
46 # 70 "/usr/include/i386/_types.h" 3 4
47 typedef int __darwin_ct_rune_t;
48

```

```

49
50
51
52
53 typedef union {
54     char __mbstate8[128];
55     long long __mbstateL;
56 } __mbstate_t;
57
58 typedef __mbstate_t __darwin_mbstate_t;
59
60
61 typedef long int __darwin_ptrdiff_t;
62
63
64
65
66
67
68
69 typedef long unsigned int __darwin_size_t;
70
71
72
73
74
75 typedef __builtin_va_list __darwin_va_list;
76
77
78
79
80
81 typedef int __darwin_wchar_t;
82
83
84
85
86 typedef __darwin_wchar_t __darwin_rune_t;
87
88
89 typedef int __darwin_wint_t;
90
91
92
93
94 typedef unsigned long __darwin_clock_t;
95 typedef __uint32_t __darwin_socklen_t;
96 typedef long __darwin_ssize_t;
97 typedef long __darwin_time_t;

```

```

98 # 33 "/usr/include/machine/_types.h" 2 3 4
99 # 34 "/usr/include/sys/_types.h" 2 3 4
100 # 55 "/usr/include/sys/_types.h" 3 4
101 typedef __int64_t __darwin_blkcnt_t;
102 typedef __int32_t __darwin_blksize_t;
103 typedef __int32_t __darwin_dev_t;
104 typedef unsigned int __darwin_fsblkcnt_t;
105 typedef unsigned int __darwin_fsfilcnt_t;
106 typedef __uint32_t __darwin_gid_t;
107 typedef __uint32_t __darwin_id_t;
108 typedef __uint64_t __darwin_ino64_t;
109
110 typedef __darwin_ino64_t __darwin_ino_t;
111
112
113
114 typedef __darwin_natural_t __darwin_mach_port_name_t;
115 typedef __darwin_mach_port_name_t __darwin_mach_port_t;
116 typedef __uint16_t __darwin_mode_t;
117 typedef __int64_t __darwin_off_t;
118 typedef __int32_t __darwin_pid_t;
119 typedef __uint32_t __darwin_sigset_t;
120 typedef __int32_t __darwin_suseconds_t;
121 typedef __uint32_t __darwin_uid_t;
122 typedef __uint32_t __darwin_useconds_t;
123 typedef unsigned char __darwin_uuid_t[16];
124 typedef char __darwin_uuid_string_t[37];
125
126
127 # 1 "/usr/include/sys/_pthread/_pthread_types.h" 1 3 4
128 # 57 "/usr/include/sys/_pthread/_pthread_types.h" 3 4
129 struct __darwin_pthread_handler_rec {
130     void (*__routine)(void *);
131     void *__arg;
132     struct __darwin_pthread_handler_rec *__next;
133 };
134
135 struct _opaque_pthread_attr_t {
136     long __sig;
137     char __opaque[56];
138 };
139
140 struct _opaque_pthread_cond_t {
141     long __sig;
142     char __opaque[40];
143 };
144
145 struct _opaque_pthread_condattr_t {
146     long __sig;

```

```

147  char  __opaque[8];
148  };
149
150  struct _opaque_pthread_mutex_t {
151      long  __sig;
152      char  __opaque[56];
153  };
154
155  struct _opaque_pthread_mutexattr_t {
156      long  __sig;
157      char  __opaque[8];
158  };
159
160  struct _opaque_pthread_once_t {
161      long  __sig;
162      char  __opaque[8];
163  };
164
165  struct _opaque_pthread_rwlock_t {
166      long  __sig;
167      char  __opaque[192];
168  };
169
170  struct _opaque_pthread_rwlockattr_t {
171      long  __sig;
172      char  __opaque[16];
173  };
174
175  struct _opaque_pthread_t {
176      long  __sig;
177      struct __darwin_pthread_handler_rec *__cleanup_stack;
178      char  __opaque[8176];
179  };
180
181  typedef struct _opaque_pthread_attr_t __darwin_pthread_attr_t;
182  typedef struct _opaque_pthread_cond_t __darwin_pthread_cond_t;
183  typedef struct _opaque_pthread_condattr_t __darwin_pthread_condattr_t;
184  typedef unsigned long __darwin_pthread_key_t;
185  typedef struct _opaque_pthread_mutex_t __darwin_pthread_mutex_t;
186  typedef struct _opaque_pthread_mutexattr_t __darwin_pthread_mutexattr_t;
187  typedef struct _opaque_pthread_once_t __darwin_pthread_once_t;
188  typedef struct _opaque_pthread_rwlock_t __darwin_pthread_rwlock_t;
189  typedef struct _opaque_pthread_rwlockattr_t __darwin_pthread_rwlockattr_t;
190  typedef struct _opaque_pthread_t *__darwin_pthread_t;
191  # 81 "/usr/include/sys/_types.h" 2 3 4
192  # 28 "/usr/include/_types.h" 2 3 4
193  # 39 "/usr/include/_types.h" 3 4
194  typedef int __darwin_nl_item;
195  typedef int __darwin_wctrans_t;

```

```

196
197 typedef __uint32_t __darwin_wctype_t;
198 # 68 "/usr/include/stdio.h" 2 3 4
199
200
201
202 # 1 "/usr/include/sys/_types/_va_list.h" 1 3 4
203 # 31 "/usr/include/sys/_types/_va_list.h" 3 4
204 typedef __darwin_va_list va_list;
205 # 72 "/usr/include/stdio.h" 2 3 4
206 # 1 "/usr/include/sys/_types/_size_t.h" 1 3 4
207 # 30 "/usr/include/sys/_types/_size_t.h" 3 4
208 typedef __darwin_size_t size_t;
209 # 73 "/usr/include/stdio.h" 2 3 4
210 # 1 "/usr/include/sys/_types/_null.h" 1 3 4
211 # 74 "/usr/include/stdio.h" 2 3 4
212
213 # 1 "/usr/include/sys/stdio.h" 1 3 4
214 # 39 "/usr/include/sys/stdio.h" 3 4
215 int renameat(int, const char *, int, const char *) __attribute__((avai
216 # 76 "/usr/include/stdio.h" 2 3 4
217
218 typedef __darwin_off_t fpos_t;
219 # 88 "/usr/include/stdio.h" 3 4
220 struct __sbuf {
221     unsigned char *_base;
222     int _size;
223 };
224
225
226 struct __sFILEX;
227 # 122 "/usr/include/stdio.h" 3 4
228 typedef struct __sFILE {
229     unsigned char *_p;
230     int _r;
231     int _w;
232     short _flags;
233     short _file;
234     struct __sbuf _bf;
235     int _lbfsz;
236
237
238     void *_cookie;
239     int (*_close)(void *);
240     int (*_read)(void *, char *, int);
241     fpos_t (*_seek)(void *, fpos_t, int);
242     int (*_write)(void *, const char *, int);
243
244

```

```

245  struct __sbuf _ub;
246  struct __sFILEX *_extra;
247  int _ur;
248
249
250  unsigned char _ubuf[3];
251  unsigned char _nbuf[1];
252
253
254  struct __sbuf _lb;
255
256
257  int _blksize;
258  fpos_t _offset;
259 } FILE;
260
261
262 extern FILE *__stdinp;
263 extern FILE *__stdoutp;
264 extern FILE *__stderrp;
265 # 231 "/usr/include/stdio.h" 3 4
266 void clearerr(FILE *);
267 int fclose(FILE *);
268 int feof(FILE *);
269 int ferror(FILE *);
270 int fflush(FILE *);
271 int fgetc(FILE *);
272 int fgetpos(FILE * restrict, fpos_t *);
273 char *fgets(char * restrict, int, FILE *);
274
275
276
277 FILE *fopen(const char * restrict, const char * restrict) __asm("__fopen");
278
279 int fprintf(FILE * restrict, const char * restrict, ...) __attribute__((format(printf, 2, 3)));
280 int fputc(int, FILE *);
281 int fputs(const char * restrict, FILE * restrict) __asm("__fputs");
282 size_t fread(void * restrict, size_t, size_t, FILE * restrict);
283 FILE *freopen(const char * restrict, const char * restrict,
284               FILE * restrict) __asm("__freopen");
285 int fscanf(FILE * restrict, const char * restrict, ...) __attribute__((format(printf, 2, 3)));
286 int fseek(FILE *, long, int);
287 int fsetpos(FILE *, const fpos_t *);
288 long ftell(FILE *);
289 size_t fwrite(const void * restrict, size_t, size_t, FILE * restrict);
290 int getc(FILE *);
291 int getchar(void);
292 char *gets(char *);
293 void perror(const char *);

```

```

294 int printf(const char * restrict, ...) __attribute__((__format__ (__pr
295 int putc(int, FILE *);
296 int putchar(int);
297 int puts(const char *);
298 int remove(const char *);
299 int rename (const char *, const char *);
300 void rewind(FILE *);
301 int scanf(const char * restrict, ...) __attribute__((__format__ (__sca
302 void setbuf(FILE * restrict, char * restrict);
303 int setvbuf(FILE * restrict, char * restrict, int, size_t);
304 int sprintf(char * restrict, const char * restrict, ...) __attribute__
305 int sscanf(const char * restrict, const char * restrict, ...) __attrib
306 FILE *tmpfile(void);
307
308
309 __attribute__((deprecated("This function is provided for compatibility
Due to security concerns inherent in the design of tmpnam(3), it is hi
310
311 char *tmpnam(char *);
312 int ungetc(int, FILE *);
313 int vfprintf(FILE * restrict, const char * restrict, va_list) __attrib
314 int vprintf(const char * restrict, va_list) __attribute__((__format__
315 int vsprintf(char * restrict, const char * restrict, va_list) __attrib
316 # 296 "/usr/include/stdio.h" 3 4
317 char *ctermid(char *);
318
319
320
321
322
323 FILE *fdopen(int, const char *) __asm("_" "fdopen" );
324
325 int fileno(FILE *);
326 # 315 "/usr/include/stdio.h" 3 4
327 int pclose(FILE *);
328
329
330
331 FILE *popen(const char *, const char *) __asm("_" "popen" );
332 # 337 "/usr/include/stdio.h" 3 4
333 int __srget(FILE *);
334 int __svfscanf(FILE *, const char *, va_list) __attribute__((__format__
335 int __swbuf(int, FILE *);
336 # 348 "/usr/include/stdio.h" 3 4
337 inline __attribute__((__always_inline__)) int __sputc(int _c, FILE *_
338 if (--_p->_w >= 0 || (_p->_w >= _p->_lbfsz && (char)_c != '\n'))
339 return (*_p->_p++ = _c);
340 else
341 return (__swbuf(_c, _p));

```



```

342 }
343 # 374 "/usr/include/stdio.h" 3 4
344 void flockfile(FILE *);
345 int ftrylockfile(FILE *);
346 void funlockfile(FILE *);
347 int getc_unlocked(FILE *);
348 int getchar_unlocked(void);
349 int putc_unlocked(int, FILE *);
350 int putchar_unlocked(int);
351
352
353
354 int getw(FILE *);
355 int putw(int, FILE *);
356
357
358
359 __attribute__((deprecated("This function is provided for compatibility
    Due to security concerns inherent in the design of tempnam(3), it is h
360
361 char *tempnam(const char *, const char *) __asm("_" "tempnam" );
362 # 411 "/usr/include/stdio.h" 3 4
363 # 1 "/usr/include/sys/_types/_off_t.h" 1 3 4
364 # 30 "/usr/include/sys/_types/_off_t.h" 3 4
365 typedef __darwin_off_t off_t;
366 # 412 "/usr/include/stdio.h" 2 3 4
367
368
369 int fseeko(FILE *, off_t, int);
370 off_t ftello(FILE *);
371
372
373
374
375
376 int snprintf(char * restrict, size_t, const char * restrict, ...) __at
377 int vfscanf(FILE * restrict, const char * restrict, va_list) __attribu
378 int vscanf(const char * restrict, va_list) __attribute__((__format__ (
379 int vsnprintf(char * restrict, size_t, const char * restrict, va_list)
380 int vsscanf(const char * restrict, const char * restrict, va_list) __a
381 # 436 "/usr/include/stdio.h" 3 4
382 # 1 "/usr/include/sys/_types/_ssize_t.h" 1 3 4
383 # 30 "/usr/include/sys/_types/_ssize_t.h" 3 4
384 typedef __darwin_ssize_t ssize_t;
385 # 437 "/usr/include/stdio.h" 2 3 4
386
387
388 int dprintf(int, const char * restrict, ...) __attribute__((__format__
389 int vdprintf(int, const char * restrict, va_list) __attribute__((__for

```

```

390 ssize_t getdelim(char ** restrict, size_t * restrict, int, FILE * rest
391 ssize_t getline(char ** restrict, size_t * restrict, FILE * restrict)
392 # 452 "/usr/include/stdio.h" 3 4
393 extern const int sys_nerr;
394 extern const char *const sys_errlist[];
395
396 int asprintf(char ** restrict, const char * restrict, ...) __attribute
397 char *ctermid_r(char *);
398 char *fgetln(FILE *, size_t *);
399 const char *fmtcheck(const char *, const char *);
400 int fpurge(FILE *);
401 void setbuffer(FILE *, char *, int);
402 int setlinebuf(FILE *);
403 int vasprintf(char ** restrict, const char * restrict, va_list) __attr
404 FILE *zopen(const char *, const char *, int);
405
406
407
408
409
410 FILE *funopen(const void *,
411               int (*)(void *, char *, int),
412               int (*)(void *, const char *, int),
413               fpos_t (*)(void *, fpos_t, int),
414               int (*)(void *));
415 # 492 "/usr/include/stdio.h" 3 4
416 # 1 "/usr/include/secure/_stdio.h" 1 3 4
417 # 31 "/usr/include/secure/_stdio.h" 3 4
418 # 1 "/usr/include/secure/_common.h" 1 3 4
419 # 32 "/usr/include/secure/_stdio.h" 2 3 4
420 # 42 "/usr/include/secure/_stdio.h" 3 4
421 extern int __sprintf_chk (char * restrict, int, size_t,
422                          const char * restrict, ...);
423 # 52 "/usr/include/secure/_stdio.h" 3 4
424 extern int __snprintf_chk (char * restrict, size_t, int, size_t,
425                          const char * restrict, ...);
426
427
428
429
430
431
432
433 extern int __vsprintf_chk (char * restrict, int, size_t,
434                          const char * restrict, va_list);
435
436
437
438

```

```

439
440
441
442 extern int __vsnprintf_chk (char * restrict, size_t, int, size_t,
443     const char * restrict, va_list);
444 # 493 "/usr/include/stdio.h" 2 3 4
445 # 2 "apusb.c" 2
446
447 int main(int argc, char* argv[])
448 {
449     int a, b;
450     scanf("%d%d", &a, &b);
451     int c = a + b;
452     printf("%d\n", c);
453     return 0;
454 }

```

Listing 3: apusb.s

```

3.
1     .section      __TEXT,__text,regular,pure_instructions
2     .macosx_version_min 10, 10
3     .globl      _main
4     .align      4, 0x90
5 _main:                                           ## @main
6     .cfi_startproc
7 ## BB#0:
8     pushq      %rbp
9 Ltmp0:
10    .cfi_def_cfa_offset 16
11 Ltmp1:
12    .cfi_offset %rbp, -16
13    movq      %rsp, %rbp
14 Ltmp2:
15    .cfi_def_cfa_register %rbp
16    subq      $48, %rsp
17    leaq      L_.str(%rip), %rax
18    leaq      -20(%rbp), %rcx
19    leaq      -24(%rbp), %rdx
20    movl      $0, -4(%rbp)
21    movl      %edi, -8(%rbp)
22    movq      %rsi, -16(%rbp)
23    movq      %rax, %rdi
24    movq      %rcx, %rsi
25    movb      $0, %al
26    callq     _scanf
27    leaq      L_.str1(%rip), %rdi
28    movl      -20(%rbp), %r8d
29    addl      -24(%rbp), %r8d
30    movl      %r8d, -28(%rbp)
31    movl      -28(%rbp), %esi

```

```

32      movl    %eax, -32(%rbp)          ## 4-byte Spill
33      movb    $0, %al
34      callq   _printf
35      xorl    %esi, %esi
36      movl    %eax, -36(%rbp)          ## 4-byte Spill
37      movl    %esi, %eax
38      addq    $48, %rsp
39      popq    %rbp
40      retq
41      .cfi_endproc
42
43      .section      __TEXT,__cstring,cstring_literals
44 L_.str:                                     ## @.str
45      .asciz     "%d%d"
46
47 L_.str1:                                    ## @.str1
48      .asciz     "%d\n"
49
50
51 .subsections_via_symbols

```

4.

实验体会：