

目录

第7章家庭作业-链接	1.1
7.6	1.2
7.7	1.3
7.8	1.4
7.9	1.5
7.10	1.6
7.11	1.7
7.12	1.8
7.13	1.9

李一鸣

1160300625

7.6

```

/* 7.6
 * swap.c
 */

extern int buf[];

int* bufp0 = &buf[0];
static int* bufp1;

static void incr() {
    static int count=0;
    count++;
}

void swap() {
    int temp;

    incr();
    bufp1 = &buf[1];
    temp = *bufp0;
    *bufp0 = *bufp1;
    *bufp1 = temp;
}

```

符号	.symtab条目?	符号类型	在哪个模块中定义	节
buf	是	external	m. o	.data
bufp0	是	global	swap. o	.data
bufp1	是	local	swap. o	.bss
incr	是	global	swap. o	.text
count	是	local	swap. o	.bss
swap	是	global	swap. o	.text
temp	否	×	×	×

```
(cd code/7.6; make && make sym)
```

输出如下: gcc -c swap.c objdump -t swap.o

```

swap.o:      file format elf64-x86-64

SYMBOL TABLE:
0000000000000000 l    df  *ABS*  0000000000000000 swap.c
0000000000000000 l    d  .text  0000000000000000 .text
0000000000000000 l    d  .data  0000000000000000 .data
0000000000000000 l    d  .bss  0000000000000000 .bss
0000000000000000 l    O  .bss  0000000000000008 bufp1
0000000000000000 l    F  .text  0000000000000016 incr
0000000000000008 l    O  .bss  0000000000000004 count.1835
0000000000000000 l    d  .note.GNU-stack  0000000000000000 .note.GNU-stack
0000000000000000 l    d  .eh_frame  0000000000000000 .eh_frame
0000000000000000 l    d  .comment  0000000000000000 .comment
0000000000000000 g    O  .data  0000000000000008 bufp0
0000000000000000             *UND*  0000000000000000 buf

```

```
0000000000000016 g      F .text  0000000000000004a swap
```

7.7

y = 15212, 16进制值为: 0x3B6C
 x = 15213, 16进制值为: 0x3B6D

修改bar5.c而不改变变量名字，应该将x修改为0x00003B6D00003B6C

下面将0x 0000 0000 0000 0000 0011 1011 0110 1101 0000 0000 0000 0011 1011 0110 1100 分段 |s(63):1|exp(62-52):11|frac(51-0):52| |-|-|-| 0|000 0000 0000|0000 0011 1011 0110 1101 0000 0000 0000 0011 1011 011 0 1100|

Bias = $2^{(11-1)} - 1 = 1023$

非规格化数:

E = 1 - Bias = -1022

M = f = 0.014508247378866

表示的浮点数为: 0.014508247378866 * 2^{-1022} 接近于0，因此这种做法行不通。

考虑使用强制类型转换:

```
/* bar5.c */
double x;

void f(){
    int *p = (int*)&x;
    *p = 15213;
    *(p+1) = 15212;
}
```

```
(cd code/7.7; make orig; ./foobar5-orig)
```

输出为:

```
gcc -Wall -Og -o foobar5-orig foo5.c bar5-original.c
/usr/bin/ld: Warning: alignment 4 of symbol `x' in /tmp/ccGZaVng.o is smaller than 8 in /tmp/ccvNRlW
M.o
x = 0x0 y = 0x80000000

(cd code/7.7; make; ./foobar5)
```

输出为:

```
gcc -Wall -Og -o foobar5 foo5.c bar5.c
/usr/bin/ld: Warning: alignment 4 of symbol `x' in /tmp/cc5CleTm.o is smaller than 8 in /tmp/ccS16us
c.o
x = 0x3b6d y = 0x3b6c
```

7.8

A.

(a)

(cd ./code/7.8; make A)

输出为:

```
gcc -Wall -Og -o A A-m1.c A-m2.c
A-m2.c:2:12: warning: 'main' is usually a function [-Wmain]
 static int main=1;
           ^
A-m2.c:2:12: warning: 'main' defined but not used [-Wunused-variable]
A-m2.c: In function 'p2':
A-m2.c:5:1: warning: control reaches end of non-void function [-Wreturn-type]
}
^
```

(a)

REF(main. 1) → DEF(main. 1)

REF(main. 2) → DEF(main. 2)

(b)

(cd ./code/7.8; make B)

输出为:

```
gcc -Wall -Og -o B B-m1.c B-m2.c
B-m1.c:3:6: warning: return type of 'main' is not 'int' [-Wmain]
 void main()
           ^
B-m2.c: In function 'p2':
B-m2.c:5:1: warning: control reaches end of non-void function [-Wreturn-type]
}
^
```

REF(x. 1) → DEF(x. 1) 和 DEF(x. 2) 中任意选择一个

DEF(x. 2) → DEF(x. 1) 和 DEF(x. 2) 中任意选择一个

(c)

(cd ./code/7.8; make C)

输出为:

```
gcc -Wall -Og -o C C-m1.c C-m2.c
C-m1.c:3:6: warning: return type of 'main' is not 'int' [-Wmain]
 void main()
           ^
C-m2.c: In function 'p2':
C-m2.c:4:1: warning: control reaches end of non-void function [-Wreturn-type]
}
^
```

```
/tmp/ccWqVgch.o:(.data+0x0): multiple definition of `x'  
/tmp/ccZB5dUV.o:(.data+0x0): first defined here  
/usr/bin/ld: Warning: size of symbol `x' changed from 4 in /tmp/ccZB5dUV.o to 8 in /tmp/ccWqVgch.o  
collect2: error: ld returned 1 exit status  
makefile:14: recipe for target 'C' failed  
make: *** [C] Error 1
```

可见x有多处强符号定义，出现链接错误。

7.9

```
/* foo6.c */
void p2(void);

int main()
{
    p2();
    return 0;
}
```

```
#include <stdio.h>

char main;

void p2()
{
    printf("0x%llx\n", main);
}
```

```
(cd ./code/7.9; make; ./foobar6)
```

输出为:

```
gcc -Wall -Og -o foobar6 foo6.c bar6.c
bar6.c:3:6: warning: 'main' is usually a function [-Wmain]
    char main;
           ^
0x48
```

在foo6.c中，main是强符号；在bar6.c中，main是弱符号，因此bar6.c中对main的使用链接器会选择foo6.c中的强符号，最终打印的是foo6.c中main函数的地址。

```
(cd ./code/7.9; make sym)
```

输出为:

```
0000000000400546 <main>:
400546: 48 83 ec 08          sub    $0x8,%rsp
40054a: e8 0a 00 00 00       callq  400559 <p2>
40054f: b8 00 00 00 00       mov    $0x0,%eax
400554: 48 83 c4 08          add    $0x8,%rsp
400558: c3                   retq
```

0x48是main函数的第一条指令的第一个字节。也就是说bar6.c中char main的地址是foo6.c中main函数的地址。

7. 10

A.

```
gcc p.o libx.a
```

之所以不用

```
gcc p.o libx.a p.o
```

是因为p.o是目标文件，而不是静态库文件，其中所有定义都会保留，不用再次重复出现

B.

```
gcc p.o libx.a liby.a libx.a
```

C.

```
gcc p.o libx.a liby.a libx.a libz.a
```

7.11

在数据段的程序头部表有如下信息：

PHT信息	值	解释
flags	rw-	有读写权限
off	0xdf8	数据从目标文件中偏移0xdf8开始的. data开始初始化
filesz	0x228	共初始化. data节中的0x228个字节
vaddr	0x600df8	数据开始于内存地址0x600df8处
memsz	0x230	总的内存大小为0x230字节
align	2**21	要求 $(vaddr-off)\%align==0$

总的内存大小为0x230字节，而只从. data节初始化0x228个字节，还有8个字节对应于运行时将被初始化为0的. bss数据。

7.12

r.type = R_X86_64_PC32 PC相对寻址

A.

ADDR(s) = ADDR(.text) = 0x4004e0

ADDR(r.symbol) = ADDR(swap) = 0x4004f8

引用的运行时地址:

refaddr = ADDR(s) + r.offset = 0x4004e0 + 0xa = 0x4004ea

更新该引用，使得它在运行时指向swap程序:

*refptr = (unsigned)(ADDR(r.symbol) + r.append - refaddr) = (unsigned)(0x4004f8 + (-4) - 0x4004ea) = (unsigned)0xa

验算:

0x4004e0 + 0xa + 4 + 0xa == 0x4004f8

B.

同上

*refptr = 0x400500 + (-4) - 0x4004d0 - 0xa = 0x22

7.13

A.

```
whereis lib.c
lib: /usr/local/lib /usr/src/linux-headers-4.10.0-37-generic/lib /usr/src/linux-headers-4.10.0-37/li
b /usr/src/linux-headers-4.10.0-28-generic/lib /usr/src/linux-headers-4.10.0-28/lib

whereis libm.a
libm:
```

发现没有相应的文件，因此用 `apt-file` 查询

```
sudo apt-get install apt-file
sudo apt-file update
apt-file search libm.a
apt-file search libm.so
```

也可以到paste.ubunut.com查询。

这是从Ubuntu仓库查询到的结果：



Datei	Pakete
/usr/share/anjuta/templates/library/src/lib.c	anjuta-common
/usr/share/doc/gprbuild/examples/first_steps/util_src/lib.c	gprbuild
/usr/share/doc/gprbuild/examples/libraries/lib_src/lib.c	gprbuild
/usr/share/doc/gprbuild/examples/subsystems/util_src/lib.c	gprbuild

libm.a

/usr/lib/x86_64-linux-gnu/diet/lib-x86_64/libm.a	dietlibc-dev [amd64]
/usr/lib/x86_64-linux-gnu/libm.a	libc6-dev [amd64]
/usr/lib/x86_64-linux-musl/libm.a	musl-dev [amd64]
/usr/lib32/libm.a	libc6-dev-i386 [amd64], libc6-dev-s390 [s390x]
/usr/lib64/libm.a	libc6-dev-amd64 [i386]
/usr/libx32/libm.a	libc6-dev-x32 [i386, amd64]
/usr/m68k-linux-gnu/lib/libm.a	libc6-dev-m68k-cross
/usr/mips-linux-gnu/lib/libm.a	libc6-dev-mips-cross
/usr/mips-linux-gnu/lib32/libm.a	libc6-dev-mipsn32-mips-cross
/usr/mips-linux-gnu/lib64/libm.a	libc6-dev-mips64-mips-cross

用 `apt-get` 安装好相应的软件包后：

```
whereis libm.a
libm: /usr/lib64/libm.a /usr/lib64/libm.so

ar t /usr/lib64/libm.a
```

输出为：

```
s_lib_version.o
s_matherr.o
s_signgam.o
fclrexcpt.o
```

```
fgetexcptflg.o
fraiseexcpt.o
fsetexcptflg.o
fetestexcept.o
fegetround.o
fesetround.o
fegetenv.o
feholdexcpt.o
fesetenv.o
feupdateenv.o
t_exp.o
fedisblxcpt.o
feenablxcpt.o
fegetexcept.o
powl_helper.o
e_acos.o
e_acosh.o
e_asin.o
e_atan2.o
e_atanh.o
e_cosh.o
e_exp.o
e_fmod.o
e_hypot.o
e_j0.o
e_j1.o
e_jn.o
e_lgamma_r.o
e_log.o
e_log10.o
e_pow.o
e_rem_pio2.o
e_remainder.o
e_scalb.o
e_sinh.o
e_sqrt.o
e_gamma_r.o
e_ilogb.o
k_cos.o
k_rem_pio2.o
k_sin.o
k_tan.o
s_asinh.o
s_atan.o
s_cbrt.o
s_ceil.o
s_cos.o
s_erf.o
s_expm1.o
s fabs.o
s_floor.o
s_log1p.o
w_log1p.o
s_logb.o
s_nextafter.o
s_nexttoward.o
s_rint.o
s_scalbln.o
w_scalbln.o
s_significand.o
```

```
s_sin.o
s_tan.o
s_tanh.o
w_acos.o
w_acosh.o
w_asin.o
w_atan2.o
w_atanh.o
w_cosh.o
w_exp.o
w_exp2.o
w_exp10.o
w_fmod.o
w_tgamma.o
w_hypot.o
w_j0.o
w_j1.o
w_jn.o
w_lgamma.o
w_lgamma_r.o
w_log.o
w_log10.o
w_pow.o
w_remainder.o
w_scalb.o
w_sinh.o
w_sqrt.o
w_ilogb.o
s_fpclassify.o
s_fmax.o
s_fmin.o
s_fdim.o
s_nan.o
s_trunc.o
s_remquo.o
e_log2.o
e_exp2.o
s_round.o
s_nearbyint.o
s_sincos.o
conj.o
cimag.o
creal.o
cabs.o
carg.o
s_cexp.o
s_csinh.o
s_ccosh.o
s_clog.o
s_catatan.o
s_casin.o
s_ccos.o
s_csin.o
s_ctan.o
s_ctanh.o
s_cacos.o
s_casinh.o
s_cacosh.o
s_catanh.o
s_csqrt.o
```

```
s_cpow.o
s_cproj.o
s_clog10.o
s_fma.o
s_lrint.o
s_llrint.o
s_lround.o
s_llround.o
e_exp10.o
w_log2.o
s_issignaling.o
m_isinf.o
m_isnan.o
m_finite.o
m_copysign.o
m_modf.o
m_scalbn.o
m_frexp.o
m_ldexp.o
m_signbit.o
x2y2m1.o
k_casinh.o
gamma_product.o
k_standard.o
lgamma_neg.o
lgamma_product.o
w_lgamma_compat.o
e_acosf.o
e_acoshf.o
e_asinf.o
e_atan2f.o
e_atanhf.o
e_coshf.o
e_expf.o
e_fmodf.o
e_hypotf.o
e_j0f.o
e_j1f.o
e_jnf.o
e_lgammaf_r.o
e_logf.o
e_log10f.o
e_powf.o
e_rem_pio2f.o
e_remainderf.o
e_scalbf.o
e_sinhf.o
e_sqrtf.o
e_gammaf_r.o
e_ilogbf.o
k_cosf.o
k_rem_pio2f.o
k_sinf.o
k_tanf.o
s_asinhf.o
s_atanf.o
s_cbrtf.o
s_ceilf.o
s_cosf.o
s_erff.o
```

```
s_expm1f.o
s_fabsf.o
s_floorf.o
s_log1pf.o
w_log1pf.o
s_logbf.o
s_nextafterf.o
s_nexttowardf.o
s_rintf.o
s_scalblnf.o
w_scalblnf.o
s_significandf.o
s_sinf.o
s_tanf.o
s_tanhf.o
w_acosf.o
w_acoshf.o
w_asinf.o
w_atan2f.o
w_atanhf.o
w_coshf.o
w_expf.o
w_exp2f.o
w_exp10f.o
w_fmodf.o
w_tgammaf.o
w_hypotf.o
w_j0f.o
w_j1f.o
w_jnf.o
w_lgammaf.o
w_lgammaf_r.o
w_logf.o
w_log10f.o
w_powf.o
w_remainderf.o
w_scalbf.o
w_sinhf.o
w_sqrtf.o
w_ilogbf.o
s_fpclassifyf.o
s_fmaxf.o
s_fminf.o
s_fdimf.o
s_nanf.o
s_truncf.o
s_remquo.f
e_log2f.o
e_exp2f.o
s_roundf.o
s_nearbyintf.o
s_sincosf.o
conjf.o
cimagf.o
crealf.o
cabsf.o
cargf.o
s_cexpf.o
s_csinhf.o
s_ccoshf.o
```

```
s_clogf.o
s_catanf.o
s_casinff.o
s_ccosf.o
s_csinff.o
s_ctanf.o
s_ctanhf.o
s_cacosf.o
s_casinhf.o
s_cacoshf.o
s_catanhf.o
s_csqrtf.o
s_cpowf.o
s_cprojf.o
s_clog10f.o
s_fmaf.o
s_lrintf.o
s_llrintf.o
s_lroundf.o
s_llroundf.o
e_exp10f.o
w_log2f.o
s_issignalingf.o
m_isinff.o
m_isnanf.o
m_finitef.o
m_copysignf.o
m_modff.o
m_scalbnf.o
m_frexpf.o
m_ldexpf.o
m_signbitf.o
x2y2m1f.o
k_casinhf.o
gamma_productf.o
k_standardf.o
lgamma_negf.o
lgamma_productf.o
w_lgamma_compatf.o
e_acosl.o
e_acoshl.o
e_asinl.o
e_atan2l.o
e_atanh1.o
e_coshl.o
e_expl.o
e_fmodl.o
e_hypotl.o
e_j0l.o
e_j1l.o
e_jnl.o
e_lgammal_r.o
e_logl.o
e_log10l.o
e_powl.o
e_rem_pio2l.o
e_remainderl.o
e_scalbl.o
e_sinh1.o
e_sqrtl.o
```

```
e_gammal_r.o
e_ilogbl.o
k_cosl.o
k_rem_pio2l.o
k_sinl.o
k_tanl.o
s_asinhl.o
s_atanl.o
s_cbrtl.o
s_ceill.o
s_cosl.o
s_erfl.o
s_expm1l.o
s fabsl.o
s_floorl.o
s_log1pl.o
w_log1pl.o
s_logbl.o
s_nextafterl.o
s_nexttowardl.o
s_rintl.o
s_scalblnl.o
w_scalblnl.o
s_significandl.o
s_sinl.o
s_tanl.o
s_tanh1.o
w_acosl.o
w_acoshl.o
w_asinl.o
w_atan2l.o
w_atanh1.o
w_coshl.o
w_expl.o
w_exp2l.o
w_exp10l.o
w_fmodl.o
w_tgammal.o
w_hypotl.o
w_j0l.o
w_j1l.o
w_jnl.o
w_lgammal.o
w_lgammal_r.o
w_logl.o
w_log10l.o
w_powl.o
w_remainderl.o
w_scalbl.o
w_sinh1.o
w_sqrtl.o
w_ilogbl.o
s_fpclassifyl.o
s_fmaxl.o
s_fminl.o
s_fdiml.o
s_nanl.o
s_truncl.o
s_remquo1.o
e_log2l.o
```

```
e_exp2l.o
s_roundl.o
s_nearbyintl.o
s_sincosl.o
conj1.o
cimagl.o
creall.o
cabsl.o
cargl.o
s_cexpl.o
s_csinhl.o
s_ccoshl.o
s_clogl.o
s_catanyl.o
s_casinl.o
s_ccosl.o
s_csinl.o
s_ctanl.o
s_ctanh1.o
s_cacosl.o
s_casinh1.o
s_cacosh1.o
s_catanh1.o
s_csqrtl.o
s_cpowl.o
s_cprojl.o
s_clog10l.o
s_fmal.o
s_lrintl.o
s_llrintl.o
s_lroundl.o
s_llroundl.o
e_exp10l.o
w_log2l.o
s_issignalingl.o
m_isinfl.o
m_isnanl.o
m_finitel.o
m_copysignl.o
m_modfl.o
m_scalbnl.o
m_frexpl.o
m_ldexpl.o
m_signbitl.o
x2y2m1l.o
k_casinhl.o
gamma_productl.o
k_standardl.o
lgamma_negl.o
lgamma_productl.o
w_lgamma_compatl.o
t_sincosl.o
k_sincosl.o
branred.o
doasin.o
dosincos.o
halfulp.o
mpa.o
mpatan2.o
mpatan.o
```

```

mpexp.o
mplog.o
mpsqrt.o
mptan.o
sincos32.o
slowexp.o
slowpow.o
sincostab.o
s_floor-c.o
s_ceil-c.o
s_floorf-c.o
s_ceilf-c.o
s_rint-c.o
s_rintf-c.o
s_nearbyint-c.o
s_nearbyintf-c.o
e_exp-fma4.o
e_log-fma4.o
e_pow-fma4.o
s_atan-fma4.o
e_asin-fma4.o
e_atan2-fma4.o
s_sin-fma4.o
s_tan-fma4.o
mplog-fma4.o
mpa-fma4.o
slowexp-fma4.o
slowpow-fma4.o
sincos32-fma4.o
doasin-fma4.o
dosincos-fma4.o
halfulp-fma4.o
mpexp-fma4.o
mpatan2-fma4.o
mpatan-fma4.o
mpsqrt-fma4.o
mptan-fma4.o
e_exp-avx.o
e_log-avx.o
s_atan-avx.o
e_atan2-avx.o
s_sin-avx.o
s_tan-avx.o
mplog-avx.o
mpa-avx.o
slowexp-avx.o
mpexp-avx.o

```

B.

```
(cd ./code/7.13; make; make da)
```

输出为:

```

gcc -o Og-test -Og test.c
gcc -o Og-g-test -Og -g test.c
objdump -d Og-test > diff1
objdump -d Og-g-test > diff2
diff diff1 diff2;[ $? -eq 1 ]

```

```
2c2
< Og-test:      file format elf64-x86-64
---
> Og-g-test:    file format elf64-x86-64
```

没有区别

C.

```
ldd Og-test

linux-vdso.so.1 => (0x00007ffd24ffc000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f2c6b996000)
/lib64/ld-linux-x86-64.so.2 (0x000056119ce6d000)
```