Example I - 3A Nested functions

Young W. Lim

2019-02-21 Thr

Outline

1. Based on

2. example 3 : nested functions
   - source codes
   - Makefile

3. example 3 (nest) effects of compile and link options
   - relocatable object `func1.o`
   - `func1` in the executable object with no ld option
   - `func1` in the executable object with `-static` ld option
   - `func1` in the executable object with `-no-pie` ld option
   - `func1` in the shared object with no ld option

Based on

1. https://stac47.github.io/c/relocation/elf/tutorial/2018/03/01/understanding-relocation-elf.html

Compling 32-bit program on 64-bit gcc

- gcc -v
- gcc -m32 t.c
- sudo apt-get install gcc-multilib
- sudo apt-get install g++-multilib
- gcc-multilib
- g++-multilib
- gcc -m32
- objdump -m i386

func1.c, func2.c

```
/*::::: func1.c :::::::::::::::::::::::::*/
extern int g;
int func2(int a);

int func1(int a, int b) {
  int c = b + func2(a);
  g += c;
  return b + g;
}

/*::::: func2.c :::::::::::::::::::::::::*/
int func2(int a) {
  return a+1;
}
```

main.c

```
/*::::: main.c  ::::::::::::::::::::::::::*/
#include <stdio.h>
int g = 42;
int func1(int a, int b);
int func2(int a);

int main() {
  int a=11, b=22, c;
  c = func1(a,b);
  printf("[%d, %d] : %d\n", a, b, c);
}
```

Makefile (1)

```
CF0 =
CF1 = -fPIC
CF2 = -fno-pic
CF3 = -fno-plt

LF0 =
LF1 = -static
LF2 = -no-pie

all : static dynamic cases so

cases : case0 case1 case2 case3 case4 case5 case6 case7 case8 case9 casea caseb

so : libfunc.so libfunc_pic.so libfunc_nopic.so libfunc_noplt.so

clean :
        rm *.o *.a *.so *.out
```

Makefile (2)

```
#-------------------------------------------------------------
static : func1.c func2.c main.c
        gcc -m32 -Wall -g -c func1.c
        gcc -m32 -Wall -g -c func2.c
        ar rcs libfunc.a func1.o func2.o

        gcc -m32 -Wall -g -c main.c
        gcc -m32 -static -o nest.out main.o ./libfunc.a


dynamic : func1.c func2.c main.c
        gcc -fPIC -m32 -Wall -g -c func1.c -o func1_pic.o
        gcc -fPIC -m32 -Wall -g -c func2.c -o func2_pic.o
        gcc -shared -m32 -o libfunc.so func1_pic.o func2_pic.o

        gcc -m32 -Wall -g -c main.c
        gcc -m32 -o nest_dyn.out main.o ./libfunc.so
```

Makefile (3)

```
#-------------------------------------------------------------
libfunc.so : func1.c func2.c
	gcc $(CF0) -m32 -Wall -g -c func1.c
	gcc $(CF0) -m32 -Wall -g -c func2.c
	gcc -shared -m32 -o libfunc.so func1.o func2.o

libfunc_pic.so : func1.c func2.c
	gcc $(CF1) -m32 -Wall -g -c func1.c -o func1_pic.o
	gcc $(CF1) -m32 -Wall -g -c func2.c -o func2_pic.o
	gcc -shared -m32 -o libfunc_pic.so func1_pic.o func2_pic.o

libfunc_nopic.so : func1.c func2.c
	gcc $(CF2) -m32 -Wall -g -c func1.c -o func1_nopic.o
	gcc $(CF2) -m32 -Wall -g -c func2.c -o func2_nopic.o
	gcc -shared -m32 -o libfunc_nopic.so func1_nopic.o func2_nopic.o

libfunc_noplt.so : func1.c func2.c
	gcc $(CF3) -m32 -Wall -g -c func1.c -o func1_noplt.o
	gcc $(CF3) -m32 -Wall -g -c func2.c -o func2_noplt.o
	gcc -shared -m32 -o libfunc_noplt.so func1_noplt.o func2_noplt.o
```

Makefile (4)

```
#-------------------------------------------------------------
libfunc.a : func1.c func2.c
	gcc $(CF0) -m32 -Wall -g -c func1.c
	gcc $(CF0) -m32 -Wall -g -c func2.c
	ar rcs libfunc.a func1.o func2.o

libfunc_pic.a : func1.c func2.c
	gcc $(CF1) -m32 -Wall -g -c func1.c -o func1_pic.o
	gcc $(CF1) -m32 -Wall -g -c func2.c -o func2_pic.o
	ar rcs libfunc_pic.a func1_pic.o func2_pic.o

libfunc_nopic.a : func1.c func2.c
	gcc $(CF2) -m32 -Wall -g -c func1.c -o func1_nopic.o
	gcc $(CF2) -m32 -Wall -g -c func2.c -o func2_nopic.o
	ar rcs libfunc_nopic.a func1_nopic.o func2_nopic.o

libfunc_noplt.a : func1.c func2.c
	gcc $(CF3) -m32 -Wall -g -c func1.c -o func1_noplt.o
	gcc $(CF3) -m32 -Wall -g -c func2.c -o func2_noplt.o
	ar rcs libfunc_noplt.a func1_noplt.o func2_noplt.o
```

Makefile (5)

```
#------------------------------------------------------------
case0 : main.c libfunc.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF0)-o nest_0.out main.o ./libfunc.a

case1 : main.c libfunc_pic.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF0)-o nest_1_pic.out main.o ./libfunc_pic.a

case2 : main.c libfunc_nopic.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF0)-o nest_2_nopic.out main.o ./libfunc_nopic.a

case3 : main.c libfunc_noplt.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF0) -o nest_3_noplt.out main.o ./libfunc_noplt.a
```

Makefile (6)

```
#-------------------------------------------------------------
case4 : main.c libfunc.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF1) -o nest_4_static.out main.o ./libfunc.a

case5 : main.c libfunc_pic.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF1) -o nest_5_pic_static.out main.o ./libfunc_pic.a

case6 : main.c libfunc_nopic.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF1) -o nest_6_nopic_static.out main.o ./libfunc_nopic.a

case7 : main.c libfunc_noplt.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF1) -o nest_7_noplt_static.out main.o ./libfunc_noplt.a
```

Makefile (7)

```
#-------------------------------------------------------------
case8 : main.c libfunc.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF2) -o nest_8_nopie.out main.o ./libfunc.a

case9 : main.c libfunc_pic.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF2) -o nest_9_pic_nopie.out main.o ./libfunc_pic.a

casea : main.c libfunc_nopic.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF2) -o nest_a_nopic_nopie.out main.o ./libfunc_nopic.a

caseb : main.c libfunc_noplt.a
        gcc -m32 -Wall -g -c main.c
        gcc -m32 $(LF2) -o nest_b_noplt_nopie.out main.o ./libfunc_noplt.a
```

objview.bash

```
#!/bin/bash

for i in $( ls func1*.o ); do
  objdump -drS $i > $i.txt
done

for i in $( ls *.out *.so ); do
  objdump -drS $i | sed  -n  '/<func1>:/, /^$/p' > $i.txt
done
```

analyzing commands

- $ readelf --segments swap_dyn.out
  $ objdump -d -s swap_dyn.out
  $ objdump -d -j .plt.got swap_dyn.out
  $ objdump -d -j .plt.got swap_dyn.out
  $ gdb ... disas, x/a 0x...., c
  $ cat /proc/<pid>/map

func1.o (1)

objdump -drS func1.o

```
00000000 <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   53                      push   %ebx
   4:   83 ec 14                sub    $0x14,%esp
   7:   e8 fc ff ff ff          call   8 <func1+0x8>
                        8: R_386_PC32   __x86.get_pc_thunk.bx
   c:   81 c3 02 00 00 00       add    $0x2,%ebx
                        e: R_386_GOTPC  _GLOBAL_OFFSET_TABLE_
```

func1.o (2)

objdump -drS func1.o

```
int c = b + func2(a);
12:   83 ec 0c                sub    $0xc,%esp
15:   ff 75 08                pushl  0x8(%ebp)
18:   e8 fc ff ff ff          call   19 <func1+0x19>
                      19: R_386_PLT32 func2
1d:   83 c4 10                add    $0x10,%esp
20:   89 c2                   mov    %eax,%edx
22:   8b 45 0c                mov    0xc(%ebp),%eax
25:   01 d0                   add    %edx,%eax
27:   89 45 f4                mov    %eax,-0xc(%ebp)
```

func1.o (3)

objdump -drS func1.o

```
  g += c;
  2a:   8b 83 00 00 00 00       mov    0x0(%ebx),%eax
                        2c: R_386_GOT32X        g
  30:   8b 10                   mov    (%eax),%edx
  32:   8b 45 f4                mov    -0xc(%ebp),%eax
  35:   01 c2                   add    %eax,%edx
  37:   8b 83 00 00 00 00       mov    0x0(%ebx),%eax
                        39: R_386_GOT32X        g
  3d:   89 10                   mov    %edx,(%eax)
  return b + g;
  3f:   8b 83 00 00 00 00       mov    0x0(%ebx),%eax
                        41: R_386_GOT32X        g
  45:   8b 10                   mov    (%eax),%edx
  47:   8b 45 0c                mov    0xc(%ebp),%eax
  4a:   01 d0                   add    %edx,%eax
}
  4c:   8b 5d fc                mov    -0x4(%ebp),%ebx
  4f:   c9                      leave
  50:   c3                      ret
```

func1_pic.o with -fPIC (1)

objdump -drS func1_pic.o

```
00000000 <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   53                      push   %ebx
   4:   83 ec 14                sub    $0x14,%esp
   7:   e8 fc ff ff ff          call   8 <func1+0x8>
                        8: R_386_PC32   __x86.get_pc_thunk.bx
   c:   81 c3 02 00 00 00       add    $0x2,%ebx
                        e: R_386_GOTPC  _GLOBAL_OFFSET_TABLE_
```

func1_pic.o with -fPIC (2)

objdump -drS func1_pic.o

```
int c = b + func2(a);
12:   83 ec 0c                sub    $0xc,%esp
15:   ff 75 08                pushl  0x8(%ebp)
18:   e8 fc ff ff ff          call   19 <func1+0x19>
                      19: R_386_PLT32 func2
1d:   83 c4 10                add    $0x10,%esp
20:   89 c2                   mov    %eax,%edx
22:   8b 45 0c                mov    0xc(%ebp),%eax
25:   01 d0                   add    %edx,%eax
27:   89 45 f4                mov    %eax,-0xc(%ebp)
```

func1_pic.o with -fPIC (3)

objdump -drS func1_pic.o

```
  g += c;
  2a:   8b 83 00 00 00 00       mov    0x0(%ebx),%eax
                        2c: R_386_GOT32X        g
  30:   8b 10                   mov    (%eax),%edx
  32:   8b 45 f4                mov    -0xc(%ebp),%eax
  35:   01 c2                   add    %eax,%edx
  37:   8b 83 00 00 00 00       mov    0x0(%ebx),%eax
                        39: R_386_GOT32X        g
  3d:   89 10                   mov    %edx,(%eax)
  return b + g;
  3f:   8b 83 00 00 00 00       mov    0x0(%ebx),%eax
                        41: R_386_GOT32X        g
  45:   8b 10                   mov    (%eax),%edx
  47:   8b 45 0c                mov    0xc(%ebp),%eax
  4a:   01 d0                   add    %edx,%eax
}
  4c:   8b 5d fc                mov    -0x4(%ebp),%ebx
  4f:   c9                      leave
  50:   c3                      ret
```

func1_nopic.o with -fno-pic (1)

objdump -drS func1_nopic.o

```
00000000 <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   83 ec 18                sub    $0x18,%esp
  int c = b + func2(a);
   6:   83 ec 0c                sub    $0xc,%esp
   9:   ff 75 08                pushl  0x8(%ebp)
   c:   e8 fc ff ff ff          call   d <func1+0xd>
                        d: R_386_PC32   func2
  11:   83 c4 10                add    $0x10,%esp
  14:   89 c2                   mov    %eax,%edx
  16:   8b 45 0c                mov    0xc(%ebp),%eax
  19:   01 d0                   add    %edx,%eax
  1b:   89 45 f4                mov    %eax,-0xc(%ebp)
```

func1_nopic.o with -fno-pic (2)

objdump -drS func1_nopic.o

```
  g += c;
  1e:   8b 15 00 00 00 00       mov    0x0,%edx
                        20: R_386_32    g
  24:   8b 45 f4                mov    -0xc(%ebp),%eax
  27:   01 d0                   add    %edx,%eax
  29:   a3 00 00 00 00          mov    %eax,0x0
                        2a: R_386_32    g
  return b + g;
  2e:   8b 15 00 00 00 00       mov    0x0,%edx
                        30: R_386_32    g
  34:   8b 45 0c                mov    0xc(%ebp),%eax
  37:   01 d0                   add    %edx,%eax
}
  39:   c9                      leave
  3a:   c3                      ret    #+end_src
```

func1_noplt.o with -fno-plt (1)

objdump -drS func1_noplt.o

```
00000000 <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   53                      push   %ebx
   4:   83 ec 14                sub    $0x14,%esp
   7:   e8 fc ff ff ff          call   8 <func1+0x8>
                        8: R_386_PC32   __x86.get_pc_thunk.bx
   c:   81 c3 02 00 00 00       add    $0x2,%ebx
                        e: R_386_GOTPC  _GLOBAL_OFFSET_TABLE_
```

func1_noplt.o with -fno-plt (2)

objdump -drS func1_noplt.o

```
int c = b + func2(a);
12:   83 ec 0c                sub    $0xc,%esp
15:   ff 75 08                pushl  0x8(%ebp)
18:   ff 93 00 00 00 00       call   *0x0(%ebx)
                      1a: R_386_GOT32X        func2
1e:   83 c4 10                add    $0x10,%esp
21:   89 c2                   mov    %eax,%edx
23:   8b 45 0c                mov    0xc(%ebp),%eax
26:   01 d0                   add    %edx,%eax
28:   89 45 f4                mov    %eax,-0xc(%ebp)
```

func1_noplt.o with -fno-plt (3)

objdump -drS func1_noplt.o

```
  g += c;
  2b:   8b 83 00 00 00 00       mov    0x0(%ebx),%eax
                        2d: R_386_GOT32X        g
  31:   8b 10                   mov    (%eax),%edx
  33:   8b 45 f4                mov    -0xc(%ebp),%eax
  36:   01 c2                   add    %eax,%edx
  38:   8b 83 00 00 00 00       mov    0x0(%ebx),%eax
                        3a: R_386_GOT32X        g
  3e:   89 10                   mov    %edx,(%eax)
  return b + g;
  40:   8b 83 00 00 00 00       mov    0x0(%ebx),%eax
                        42: R_386_GOT32X        g
  46:   8b 10                   mov    (%eax),%edx
  48:   8b 45 0c                mov    0xc(%ebp),%eax
  4b:   01 d0                   add    %edx,%eax
}
  4d:   8b 5d fc                mov    -0x4(%ebp),%ebx
  50:   c9                      leave
  51:   c3                      ret
```

case 0: func1 in nest_0.out (1)

objdump -dS nest_0.out

```
00000583 <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 583:   55                      push   %ebp
 584:   89 e5                   mov    %esp,%ebp
 586:   53                      push   %ebx
 587:   83 ec 14                sub    $0x14,%esp
 58a:   e8 91 fe ff ff          call   420 <__x86.get_pc_thunk.bx>
 58f:   81 c3 49 1a 00 00       add    $0x1a49,%ebx
  int c = b + func2(a);
 595:   83 ec 0c                sub    $0xc,%esp
 598:   ff 75 08                pushl  0x8(%ebp)
 59b:   e8 34 00 00 00          call   5d4 <func2>
 5a0:   83 c4 10                add    $0x10,%esp
 5a3:   89 c2                   mov    %eax,%edx
 5a5:   8b 45 0c                mov    0xc(%ebp),%eax
 5a8:   01 d0                   add    %edx,%eax
 5aa:   89 45 f4                mov    %eax,-0xc(%ebp)
```

case 0: func1 in nest_0.out (2)

objdump -dS nest_0.out

```
  g += c;
 5ad:   8d 83 30 00 00 00       lea    0x30(%ebx),%eax
 5b3:   8b 10                   mov    (%eax),%edx
 5b5:   8b 45 f4                mov    -0xc(%ebp),%eax
 5b8:   01 c2                   add    %eax,%edx
 5ba:   8d 83 30 00 00 00       lea    0x30(%ebx),%eax
 5c0:   89 10                   mov    %edx,(%eax)
  return b + g;
 5c2:   8d 83 30 00 00 00       lea    0x30(%ebx),%eax
 5c8:   8b 10                   mov    (%eax),%edx
 5ca:   8b 45 0c                mov    0xc(%ebp),%eax
 5cd:   01 d0                   add    %edx,%eax
}
 5cf:   8b 5d fc                mov    -0x4(%ebp),%ebx
 5d2:   c9                      leave
 5d3:   c3                      ret
```

case 1: func1 in nest_1_pic.out (1)

objdump -dS nest_1_pic.out

```
00000583 <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 583:   55                      push   %ebp
 584:   89 e5                   mov    %esp,%ebp
 586:   53                      push   %ebx
 587:   83 ec 14                sub    $0x14,%esp
 58a:   e8 91 fe ff ff          call   420 <__x86.get_pc_thunk.bx>
 58f:   81 c3 49 1a 00 00       add    $0x1a49,%ebx
  int c = b + func2(a);
 595:   83 ec 0c                sub    $0xc,%esp
 598:   ff 75 08                pushl  0x8(%ebp)
 59b:   e8 34 00 00 00          call   5d4 <func2>
 5a0:   83 c4 10                add    $0x10,%esp
 5a3:   89 c2                   mov    %eax,%edx
 5a5:   8b 45 0c                mov    0xc(%ebp),%eax
 5a8:   01 d0                   add    %edx,%eax
 5aa:   89 45 f4                mov    %eax,-0xc(%ebp)
```

case 1: func1 in nest_1_pic.out (2)

objdump -dS nest_1_pic.out

```
  g += c;
 5ad:   8d 83 30 00 00 00       lea    0x30(%ebx),%eax
 5b3:   8b 10                   mov    (%eax),%edx
 5b5:   8b 45 f4                mov    -0xc(%ebp),%eax
 5b8:   01 c2                   add    %eax,%edx
 5ba:   8d 83 30 00 00 00       lea    0x30(%ebx),%eax
 5c0:   89 10                   mov    %edx,(%eax)
  return b + g;
 5c2:   8d 83 30 00 00 00       lea    0x30(%ebx),%eax
 5c8:   8b 10                   mov    (%eax),%edx
 5ca:   8b 45 0c                mov    0xc(%ebp),%eax
 5cd:   01 d0                   add    %edx,%eax
}
 5cf:   8b 5d fc                mov    -0x4(%ebp),%ebx
 5d2:   c9                      leave
 5d3:   c3                      ret
```

case 2: func1 in nest_2_nopic.out (1)

objdump -dS nest_2_nopic.out

```
000005a3 <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 5a3:   55                      push   %ebp
 5a4:   89 e5                   mov    %esp,%ebp
 5a6:   83 ec 18                sub    $0x18,%esp
  int c = b + func2(a);
 5a9:   83 ec 0c                sub    $0xc,%esp
 5ac:   ff 75 08                pushl  0x8(%ebp)
 5af:   e8 2a 00 00 00          call   5de <func2>
 5b4:   83 c4 10                add    $0x10,%esp
 5b7:   89 c2                   mov    %eax,%edx
 5b9:   8b 45 0c                mov    0xc(%ebp),%eax
 5bc:   01 d0                   add    %edx,%eax
 5be:   89 45 f4                mov    %eax,-0xc(%ebp)
```

case 2: func1 in nest_2_nopic.out (2)

objdump -dS nest_2_nopic.out

```
  g += c;
 5c1:   8b 15 08 20 00 00       mov    0x2008,%edx
 5c7:   8b 45 f4                mov    -0xc(%ebp),%eax
 5ca:   01 d0                   add    %edx,%eax
 5cc:   a3 08 20 00 00          mov    %eax,0x2008
  return b + g;
 5d1:   8b 15 08 20 00 00       mov    0x2008,%edx
 5d7:   8b 45 0c                mov    0xc(%ebp),%eax
 5da:   01 d0                   add    %edx,%eax
}
 5dc:   c9                      leave
 5dd:   c3                      ret
```

case 3: func1 in nest_3_noplt.out (1)

objdump -dS nest_3_noplt.out

```
00000583 <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 583:   55                      push   %ebp
 584:   89 e5                   mov    %esp,%ebp
 586:   53                      push   %ebx
 587:   83 ec 14                sub    $0x14,%esp
 58a:   e8 91 fe ff ff          call   420 <__x86.get_pc_thunk.bx>
 58f:   81 c3 49 1a 00 00       add    $0x1a49,%ebx
  int c = b + func2(a);
 595:   83 ec 0c                sub    $0xc,%esp
 598:   ff 75 08                pushl  0x8(%ebp)
 59b:   67 e8 34 00 00 00       addr16 call 5d5 <func2>
 5a1:   83 c4 10                add    $0x10,%esp
 5a4:   89 c2                   mov    %eax,%edx
 5a6:   8b 45 0c                mov    0xc(%ebp),%eax
 5a9:   01 d0                   add    %edx,%eax
 5ab:   89 45 f4                mov    %eax,-0xc(%ebp)
```

case 3: func1 in nest_3_noplt.out (2)

objdump -dS nest_3_noplt.out

```
  g += c;
 5ae:   8d 83 30 00 00 00       lea    0x30(%ebx),%eax
 5b4:   8b 10                   mov    (%eax),%edx
 5b6:   8b 45 f4                mov    -0xc(%ebp),%eax
 5b9:   01 c2                   add    %eax,%edx
 5bb:   8d 83 30 00 00 00       lea    0x30(%ebx),%eax
 5c1:   89 10                   mov    %edx,(%eax)
  return b + g;
 5c3:   8d 83 30 00 00 00       lea    0x30(%ebx),%eax
 5c9:   8b 10                   mov    (%eax),%edx
 5cb:   8b 45 0c                mov    0xc(%ebp),%eax
 5ce:   01 d0                   add    %edx,%eax
}
 5d0:   8b 5d fc                mov    -0x4(%ebp),%ebx
 5d3:   c9                      leave
 5d4:   c3                      ret
```

case 4: func1 in nest_4_static.out (1)

objdump -dS nest_4_static.out

```
0804890b <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 804890b:       55                      push   %ebp
 804890c:       89 e5                   mov    %esp,%ebp
 804890e:       53                      push   %ebx
 804890f:       83 ec 14                sub    $0x14,%esp
 8048912:       e8 69 fe ff ff          call   8048780 <__x86.get_pc_thunk.bx>
 8048917:       81 c3 e9 06 09 00       add    $0x906e9,%ebx
  int c = b + func2(a);
 804891d:       83 ec 0c                sub    $0xc,%esp
 8048920:       ff 75 08                pushl  0x8(%ebp)
 8048923:       e8 34 00 00 00          call   804895c <func2>
 8048928:       83 c4 10                add    $0x10,%esp
 804892b:       89 c2                   mov    %eax,%edx
 804892d:       8b 45 0c                mov    0xc(%ebp),%eax
 8048930:       01 d0                   add    %edx,%eax
 8048932:       89 45 f4                mov    %eax,-0xc(%ebp)
```

case 4: func1 in nest_4_static.out (2)

objdump -dS nest_4_static.out

```
  g += c;
 8048935:       c7 c0 68 90 0d 08       mov    $0x80d9068,%eax
 804893b:       8b 10                   mov    (%eax),%edx
 804893d:       8b 45 f4                mov    -0xc(%ebp),%eax
 8048940:       01 c2                   add    %eax,%edx
 8048942:       c7 c0 68 90 0d 08       mov    $0x80d9068,%eax
 8048948:       89 10                   mov    %edx,(%eax)
  return b + g;
 804894a:       c7 c0 68 90 0d 08       mov    $0x80d9068,%eax
 8048950:       8b 10                   mov    (%eax),%edx
 8048952:       8b 45 0c                mov    0xc(%ebp),%eax
 8048955:       01 d0                   add    %edx,%eax
}
 8048957:       8b 5d fc                mov    -0x4(%ebp),%ebx
 804895a:       c9                      leave
 804895b:       c3                      ret
```

case 5: func1 in nest_5_pic_static.out (1)

objdump -dS nest_5_pic_static.out

```
0804890b <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 804890b:       55                      push   %ebp
 804890c:       89 e5                   mov    %esp,%ebp
 804890e:       53                      push   %ebx
 804890f:       83 ec 14                sub    $0x14,%esp
 8048912:       e8 69 fe ff ff          call   8048780 <__x86.get_pc_thunk.bx>
 8048917:       81 c3 e9 06 09 00       add    $0x906e9,%ebx
  int c = b + func2(a);
 804891d:       83 ec 0c                sub    $0xc,%esp
 8048920:       ff 75 08                pushl  0x8(%ebp)
 8048923:       e8 34 00 00 00          call   804895c <func2>
 8048928:       83 c4 10                add    $0x10,%esp
 804892b:       89 c2                   mov    %eax,%edx
 804892d:       8b 45 0c                mov    0xc(%ebp),%eax
 8048930:       01 d0                   add    %edx,%eax
 8048932:       89 45 f4                mov    %eax,-0xc(%ebp)
```

case 5: func1 in nest_5_pic_static.out (2)

objdump -dS nest_5_pic_static.out

```
  g += c;
 8048935:       c7 c0 68 90 0d 08       mov    $0x80d9068,%eax
 804893b:       8b 10                   mov    (%eax),%edx
 804893d:       8b 45 f4                mov    -0xc(%ebp),%eax
 8048940:       01 c2                   add    %eax,%edx
 8048942:       c7 c0 68 90 0d 08       mov    $0x80d9068,%eax
 8048948:       89 10                   mov    %edx,(%eax)
  return b + g;
 804894a:       c7 c0 68 90 0d 08       mov    $0x80d9068,%eax
 8048950:       8b 10                   mov    (%eax),%edx
 8048952:       8b 45 0c                mov    0xc(%ebp),%eax
 8048955:       01 d0                   add    %edx,%eax
}
 8048957:       8b 5d fc                mov    -0x4(%ebp),%ebx
 804895a:       c9                      leave
 804895b:       c3                      ret
```

case 6: func1 in nest_6_nopic_static.out (1)

objdump -dS nest_6_nopic_static.out

```
0804890b <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 804890b:       55                      push   %ebp
 804890c:       89 e5                   mov    %esp,%ebp
 804890e:       83 ec 18                sub    $0x18,%esp
  int c = b + func2(a);
 8048911:       83 ec 0c                sub    $0xc,%esp
 8048914:       ff 75 08                pushl  0x8(%ebp)
 8048917:       e8 2a 00 00 00          call   8048946 <func2>
 804891c:       83 c4 10                add    $0x10,%esp
 804891f:       89 c2                   mov    %eax,%edx
 8048921:       8b 45 0c                mov    0xc(%ebp),%eax
 8048924:       01 d0                   add    %edx,%eax
 8048926:       89 45 f4                mov    %eax,-0xc(%ebp)
```

case 6: func1 in nest_6_nopic_static.out (2)

objdump -dS nest_6_nopic_static.out

```
  g += c;
 8048929:       8b 15 68 90 0d 08       mov    0x80d9068,%edx
 804892f:       8b 45 f4                mov    -0xc(%ebp),%eax
 8048932:       01 d0                   add    %edx,%eax
 8048934:       a3 68 90 0d 08          mov    %eax,0x80d9068
  return b + g;
 8048939:       8b 15 68 90 0d 08       mov    0x80d9068,%edx
 804893f:       8b 45 0c                mov    0xc(%ebp),%eax
 8048942:       01 d0                   add    %edx,%eax
}
 8048944:       c9                      leave
 8048945:       c3                      ret
```

case 7: func1 in nest_7_noplt_static.out (1)

objdump -dS nest_7_noplt_static.out

```
0804890b <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 804890b:       55                      push   %ebp
 804890c:       89 e5                   mov    %esp,%ebp
 804890e:       53                      push   %ebx
 804890f:       83 ec 14                sub    $0x14,%esp
 8048912:       e8 69 fe ff ff          call   8048780 <__x86.get_pc_thunk.bx>
 8048917:       81 c3 e9 06 09 00       add    $0x906e9,%ebx
  int c = b + func2(a);
 804891d:       83 ec 0c                sub    $0xc,%esp
 8048920:       ff 75 08                pushl  0x8(%ebp)
 8048923:       67 e8 34 00 00 00       addr16 call 804895d <func2>
 8048929:       83 c4 10                add    $0x10,%esp
 804892c:       89 c2                   mov    %eax,%edx
 804892e:       8b 45 0c                mov    0xc(%ebp),%eax
 8048931:       01 d0                   add    %edx,%eax
 8048933:       89 45 f4                mov    %eax,-0xc(%ebp)
```

case 7: func1 in nest_7_noplt_static.out (2)

objdump -dS nest_7_noplt_static.out

```
  g += c;
 8048936:       c7 c0 68 90 0d 08       mov    $0x80d9068,%eax
 804893c:       8b 10                   mov    (%eax),%edx
 804893e:       8b 45 f4                mov    -0xc(%ebp),%eax
 8048941:       01 c2                   add    %eax,%edx
 8048943:       c7 c0 68 90 0d 08       mov    $0x80d9068,%eax
 8048949:       89 10                   mov    %edx,(%eax)
  return b + g;
 804894b:       c7 c0 68 90 0d 08       mov    $0x80d9068,%eax
 8048951:       8b 10                   mov    (%eax),%edx
 8048953:       8b 45 0c                mov    0xc(%ebp),%eax
 8048956:       01 d0                   add    %edx,%eax
}
 8048958:       8b 5d fc                mov    -0x4(%ebp),%ebx
 804895b:       c9                      leave
 804895c:       c3                      ret
```

case 8: func1 in nest_8_nopie.out (1)

objdump -dS nets_8_nopie.out

```
0804848c <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 804848c:       55                      push   %ebp
 804848d:       89 e5                   mov    %esp,%ebp
 804848f:       53                      push   %ebx
 8048490:       83 ec 14                sub    $0x14,%esp
 8048493:       e8 c8 fe ff ff          call   8048360 <__x86.get_pc_thunk.bx>
 8048498:       81 c3 68 1b 00 00       add    $0x1b68,%ebx
  int c = b + func2(a);
 804849e:       83 ec 0c                sub    $0xc,%esp
 80484a1:       ff 75 08                pushl  0x8(%ebp)
 80484a4:       e8 34 00 00 00          call   80484dd <func2>
 80484a9:       83 c4 10                add    $0x10,%esp
 80484ac:       89 c2                   mov    %eax,%edx
 80484ae:       8b 45 0c                mov    0xc(%ebp),%eax
 80484b1:       01 d0                   add    %edx,%eax
 80484b3:       89 45 f4                mov    %eax,-0xc(%ebp)
```

case 8: func1 in nest_8_nopie.out (2)

objdump -dS nets_8_nopie.out

```
  g += c;
 80484b6:       c7 c0 1c a0 04 08       mov    $0x804a01c,%eax
 80484bc:       8b 10                   mov    (%eax),%edx
 80484be:       8b 45 f4                mov    -0xc(%ebp),%eax
 80484c1:       01 c2                   add    %eax,%edx
 80484c3:       c7 c0 1c a0 04 08       mov    $0x804a01c,%eax
 80484c9:       89 10                   mov    %edx,(%eax)
  return b + g;
 80484cb:       c7 c0 1c a0 04 08       mov    $0x804a01c,%eax
 80484d1:       8b 10                   mov    (%eax),%edx
 80484d3:       8b 45 0c                mov    0xc(%ebp),%eax
 80484d6:       01 d0                   add    %edx,%eax
}
 80484d8:       8b 5d fc                mov    -0x4(%ebp),%ebx
 80484db:       c9                      leave
 80484dc:       c3                      ret
```

case 9: func1 in nest_9_pic_nopie.out (1)

objdump -dS nest_9_pic_nopie.out

```
0804848c <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 804848c:       55                      push   %ebp
 804848d:       89 e5                   mov    %esp,%ebp
 804848f:       53                      push   %ebx
 8048490:       83 ec 14                sub    $0x14,%esp
 8048493:       e8 c8 fe ff ff          call   8048360 <__x86.get_pc_thunk.bx>
 8048498:       81 c3 68 1b 00 00       add    $0x1b68,%ebx
  int c = b + func2(a);
 804849e:       83 ec 0c                sub    $0xc,%esp
 80484a1:       ff 75 08                pushl  0x8(%ebp)
 80484a4:       e8 34 00 00 00          call   80484dd <func2>
 80484a9:       83 c4 10                add    $0x10,%esp
 80484ac:       89 c2                   mov    %eax,%edx
 80484ae:       8b 45 0c                mov    0xc(%ebp),%eax
 80484b1:       01 d0                   add    %edx,%eax
 80484b3:       89 45 f4                mov    %eax,-0xc(%ebp)
```

case 9: func1 in nest_9_pic_nopie.out (2)

objdump -dS nest_9_pic_nopie.out

```
  g += c;
 80484b6:       c7 c0 1c a0 04 08       mov    $0x804a01c,%eax
 80484bc:       8b 10                   mov    (%eax),%edx
 80484be:       8b 45 f4                mov    -0xc(%ebp),%eax
 80484c1:       01 c2                   add    %eax,%edx
 80484c3:       c7 c0 1c a0 04 08       mov    $0x804a01c,%eax
 80484c9:       89 10                   mov    %edx,(%eax)
  return b + g;
 80484cb:       c7 c0 1c a0 04 08       mov    $0x804a01c,%eax
 80484d1:       8b 10                   mov    (%eax),%edx
 80484d3:       8b 45 0c                mov    0xc(%ebp),%eax
 80484d6:       01 d0                   add    %edx,%eax
}
 80484d8:       8b 5d fc                mov    -0x4(%ebp),%ebx
 80484db:       c9                      leave
 80484dc:       c3                      ret
```

case 10: func1 in nest_10_nopic_nopie.out(1)

objdump -dS nest_10_nopic_nopie.out

```
0804848c <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 804848c:       55                      push   %ebp
 804848d:       89 e5                   mov    %esp,%ebp
 804848f:       83 ec 18                sub    $0x18,%esp
  int c = b + func2(a);
 8048492:       83 ec 0c                sub    $0xc,%esp
 8048495:       ff 75 08                pushl  0x8(%ebp)
 8048498:       e8 2a 00 00 00          call   80484c7 <func2>
 804849d:       83 c4 10                add    $0x10,%esp
 80484a0:       89 c2                   mov    %eax,%edx
 80484a2:       8b 45 0c                mov    0xc(%ebp),%eax
 80484a5:       01 d0                   add    %edx,%eax
 80484a7:       89 45 f4                mov    %eax,-0xc(%ebp)
```

case 10: func1 in nest_10_nopic_nopie.out(2)

objdump -dS nest_10_nopic_nopie.out

```
  g += c;
 80484aa:       8b 15 1c a0 04 08       mov    0x804a01c,%edx
 80484b0:       8b 45 f4                mov    -0xc(%ebp),%eax
 80484b3:       01 d0                   add    %edx,%eax
 80484b5:       a3 1c a0 04 08          mov    %eax,0x804a01c
  return b + g;
 80484ba:       8b 15 1c a0 04 08       mov    0x804a01c,%edx
 80484c0:       8b 45 0c                mov    0xc(%ebp),%eax
 80484c3:       01 d0                   add    %edx,%eax
}
 80484c5:       c9                      leave
 80484c6:       c3                      ret
```

case 11: func1 in nest_11_noplt_nopie.out (1)

objdump -dS nest_11_noplt_nopie.out

```
0804848c <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 804848c:       55                      push   %ebp
 804848d:       89 e5                   mov    %esp,%ebp
 804848f:       53                      push   %ebx
 8048490:       83 ec 14                sub    $0x14,%esp
 8048493:       e8 c8 fe ff ff          call   8048360 <__x86.get_pc_thunk.bx>
 8048498:       81 c3 68 1b 00 00       add    $0x1b68,%ebx
  int c = b + func2(a);
 804849e:       83 ec 0c                sub    $0xc,%esp
 80484a1:       ff 75 08                pushl  0x8(%ebp)
 80484a4:       67 e8 34 00 00 00       addr16 call 80484de <func2>
 80484aa:       83 c4 10                add    $0x10,%esp
 80484ad:       89 c2                   mov    %eax,%edx
 80484af:       8b 45 0c                mov    0xc(%ebp),%eax
 80484b2:       01 d0                   add    %edx,%eax
 80484b4:       89 45 f4                mov    %eax,-0xc(%ebp)
```

case 11: func1 in nest_11_noplt_nopie.out (2)

objdump -dS nest_11_noplt_nopie.out

```
  g += c;
 80484b7:       c7 c0 1c a0 04 08       mov    $0x804a01c,%eax
 80484bd:       8b 10                   mov    (%eax),%edx
 80484bf:       8b 45 f4                mov    -0xc(%ebp),%eax
 80484c2:       01 c2                   add    %eax,%edx
 80484c4:       c7 c0 1c a0 04 08       mov    $0x804a01c,%eax
 80484ca:       89 10                   mov    %edx,(%eax)
  return b + g;
 80484cc:       c7 c0 1c a0 04 08       mov    $0x804a01c,%eax
 80484d2:       8b 10                   mov    (%eax),%edx
 80484d4:       8b 45 0c                mov    0xc(%ebp),%eax
 80484d7:       01 d0                   add    %edx,%eax
}
 80484d9:       8b 5d fc                mov    -0x4(%ebp),%ebx
 80484dc:       c9                      leave
 80484dd:       c3                      ret
```

func1 in libfunc.so (1)

objdump -dS nets_8_nopie.out

```
0000046d <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 46d:   55                      push   %ebp
 46e:   89 e5                   mov    %esp,%ebp
 470:   53                      push   %ebx
 471:   83 ec 14                sub    $0x14,%esp
 474:   e8 f7 fe ff ff          call   370 <__x86.get_pc_thunk.bx>
 479:   81 c3 87 1b 00 00       add    $0x1b87,%ebx
  int c = b + func2(a);
 47f:   83 ec 0c                sub    $0xc,%esp
 482:   ff 75 08                pushl  0x8(%ebp)
 485:   e8 c6 fe ff ff          call   350 <func2@plt>
 48a:   83 c4 10                add    $0x10,%esp
 48d:   89 c2                   mov    %eax,%edx
 48f:   8b 45 0c                mov    0xc(%ebp),%eax
 492:   01 d0                   add    %edx,%eax
 494:   89 45 f4                mov    %eax,-0xc(%ebp)
```

func1 in libfunc.so (2)

objdump -dS nets_8_nopie.out

```
  g += c;
 497:   8b 83 f0 ff ff ff       mov    -0x10(%ebx),%eax
 49d:   8b 10                   mov    (%eax),%edx
 49f:   8b 45 f4                mov    -0xc(%ebp),%eax
 4a2:   01 c2                   add    %eax,%edx
 4a4:   8b 83 f0 ff ff ff       mov    -0x10(%ebx),%eax
 4aa:   89 10                   mov    %edx,(%eax)
  return b + g;
 4ac:   8b 83 f0 ff ff ff       mov    -0x10(%ebx),%eax
 4b2:   8b 10                   mov    (%eax),%edx
 4b4:   8b 45 0c                mov    0xc(%ebp),%eax
 4b7:   01 d0                   add    %edx,%eax
}
 4b9:   8b 5d fc                mov    -0x4(%ebp),%ebx
 4bc:   c9                      leave
 4bd:   c3                      ret
```

func1 in libfunc_pic.so (1)

objdump -dS nest_9_pic_nopie.out

```
0000046d <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 46d:   55                      push   %ebp
 46e:   89 e5                   mov    %esp,%ebp
 470:   53                      push   %ebx
 471:   83 ec 14                sub    $0x14,%esp
 474:   e8 f7 fe ff ff          call   370 <__x86.get_pc_thunk.bx>
 479:   81 c3 87 1b 00 00       add    $0x1b87,%ebx
  int c = b + func2(a);
 47f:   83 ec 0c                sub    $0xc,%esp
 482:   ff 75 08                pushl  0x8(%ebp)
 485:   e8 c6 fe ff ff          call   350 <func2@plt>
 48a:   83 c4 10                add    $0x10,%esp
 48d:   89 c2                   mov    %eax,%edx
 48f:   8b 45 0c                mov    0xc(%ebp),%eax
 492:   01 d0                   add    %edx,%eax
 494:   89 45 f4                mov    %eax,-0xc(%ebp)
```

func1 in libfunc_pic.so (2)

objdump -dS nest_9_pic_nopie.out

```
  g += c;
 497:   8b 83 f0 ff ff ff       mov    -0x10(%ebx),%eax
 49d:   8b 10                   mov    (%eax),%edx
 49f:   8b 45 f4                mov    -0xc(%ebp),%eax
 4a2:   01 c2                   add    %eax,%edx
 4a4:   8b 83 f0 ff ff ff       mov    -0x10(%ebx),%eax
 4aa:   89 10                   mov    %edx,(%eax)
  return b + g;
 4ac:   8b 83 f0 ff ff ff       mov    -0x10(%ebx),%eax
 4b2:   8b 10                   mov    (%eax),%edx
 4b4:   8b 45 0c                mov    0xc(%ebp),%eax
 4b7:   01 d0                   add    %edx,%eax
}
 4b9:   8b 5d fc                mov    -0x4(%ebp),%ebx
 4bc:   c9                      leave
 4bd:   c3                      ret
```

func1 in libfunc_nopic.so (1)

objdump -dS nest_10_nopic_nopie.out

```
0000046d <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 46d:   55                      push   %ebp
 46e:   89 e5                   mov    %esp,%ebp
 470:   83 ec 18                sub    $0x18,%esp
  int c = b + func2(a);
 473:   83 ec 0c                sub    $0xc,%esp
 476:   ff 75 08                pushl  0x8(%ebp)
 479:   e8 fc ff ff ff          call   47a <func1+0xd>
 47e:   83 c4 10                add    $0x10,%esp
 481:   89 c2                   mov    %eax,%edx
 483:   8b 45 0c                mov    0xc(%ebp),%eax
 486:   01 d0                   add    %edx,%eax
 488:   89 45 f4                mov    %eax,-0xc(%ebp)
```

func1 in libfunc_nopic.so (2)

objdump -dS nest_10_nopic_nopie.out

```
  g += c;
 48b:   8b 15 00 00 00 00       mov    0x0,%edx
 491:   8b 45 f4                mov    -0xc(%ebp),%eax
 494:   01 d0                   add    %edx,%eax
 496:   a3 00 00 00 00          mov    %eax,0x0
  return b + g;
 49b:   8b 15 00 00 00 00       mov    0x0,%edx
 4a1:   8b 45 0c                mov    0xc(%ebp),%eax
 4a4:   01 d0                   add    %edx,%eax
}
 4a6:   c9                      leave
 4a7:   c3                      ret
```

func1 in libfunc_noplt.so (1)

objdump -dS nest_11_noplt_nopie.out

```
0000045d <func1>:
extern int g;
int func2(int a);
int func1(int a, int b) {
 45d:   55                      push   %ebp
 45e:   89 e5                   mov    %esp,%ebp
 460:   53                      push   %ebx
 461:   83 ec 14                sub    $0x14,%esp
 464:   e8 f7 fe ff ff          call   360 <__x86.get_pc_thunk.bx>
 469:   81 c3 97 1b 00 00       add    $0x1b97,%ebx
  int c = b + func2(a);
 46f:   83 ec 0c                sub    $0xc,%esp
 472:   ff 75 08                pushl  0x8(%ebp)
 475:   ff 93 f8 ff ff ff       call   *-0x8(%ebx)
 47b:   83 c4 10                add    $0x10,%esp
 47e:   89 c2                   mov    %eax,%edx
 480:   8b 45 0c                mov    0xc(%ebp),%eax
 483:   01 d0                   add    %edx,%eax
 485:   89 45 f4                mov    %eax,-0xc(%ebp)
```

func1 in libfunc_noplt.so (2)

objdump -dS nest_11_noplt_nopie.out

```
  g += c;
 488:   8b 83 ec ff ff ff       mov    -0x14(%ebx),%eax
 48e:   8b 10                   mov    (%eax),%edx
 490:   8b 45 f4                mov    -0xc(%ebp),%eax
 493:   01 c2                   add    %eax,%edx
 495:   8b 83 ec ff ff ff       mov    -0x14(%ebx),%eax
 49b:   89 10                   mov    %edx,(%eax)
  return b + g;
 49d:   8b 83 ec ff ff ff       mov    -0x14(%ebx),%eax
 4a3:   8b 10                   mov    (%eax),%edx
 4a5:   8b 45 0c                mov    0xc(%ebp),%eax
 4a8:   01 d0                   add    %edx,%eax
}
 4aa:   8b 5d fc                mov    -0x4(%ebp),%ebx
 4ad:   c9                      leave
 4ae:   c3                      ret
```

case 0: func1 in $nest_0.out$

```
objdump -d nest_0.out
```

case 0: func1 in $nest_0.out$

```
objdump -d nest_0.out
```