Example I - 1A Vector addition

Young W. Lim

2019-02-21 Thr

Outline

1. Based on

2. example 1 : vector addition and multiplication
   - source codes
   - Makefile

3. example 1 (vector) effects of compile and link options
   - relocatable object `addvec.o`
   - `addvec` in the executable object with no ld option
   - `addvec` in the executable object with `-static` ld option
   - `addvec` in the executable object with `-no-pie` 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

addvec.c and multvec.c

```
/*::::: addvec.c :::::::::::::::::::::::::*/
void addvec(int *x, int *y, int *z, int n)
{
  int i;

  for (i=0; i<n; i++)
    z[i] = x[i] + y[i];

}

/*::::: multvec.c :::::::::::::::::::::::::*/
void multvec(int *x, int *y, int *z, int n)
{
  int i;

  for (i=0; i<n; i++)
    z[i] = x[i] * y[i];

}
```

main.c

```
/*::::: vector.h ::::::::::::::::::::::::::*/
void addvec(int *x, int *y, int *z, int n);
void multvec(int *x, int *y, int *z, int n);


/*::::: main.c ::::::::::::::::::::::::::::*/
#include <stdio.h>
#include "vector.h"

int x[2] = { 1, 2};
int y[2] = { 3, 4};
int z[2];


int main() {

  addvec(x, y, z, 2);
  printf("z= [%d %d]\n", z[0], z[1]);

}
```

Makefile (1)

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

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

all : static dynamic cases

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

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

Makefile (2)

```
#--------------------------------------------------------------
static : addvec.c multvec.c main.c
        gcc -m32 -Wall -c addvec.c
        gcc -m32 -Wall -c multvec.c
        ar rcs libvec.a addvec.o multvec.o

        gcc -m32 -Wall -c main.c
        gcc -m32 -static -o vector.out main.o ./libvec.a


dynamic : addvec.c multvec.c main.c
        gcc -fPIC -m32 -Wall -c addvec.c -o addvec_pic.o
        gcc -fPIC -m32 -Wall -c multvec.c -o multvec_pic.o
        gcc -shared -m32 -o libvec.so addvec_pic.o multvec_pic.o

        gcc -m32 -Wall -c main.c
        gcc -m32 -o vector_dyn.out main.o ./libvec.so
```

Makefile (3)

```
#-------------------------------------------------------------
libvec.a : addvec.c multvec.c
        gcc $(CF0) -m32 -Wall -c addvec.c
        gcc $(CF0) -m32 -Wall -c multvec.c
        ar rcs libvec.a addvec.o multvec.o

libvec_pic.a : addvec.c multvec.c
        gcc $(CF1) -m32 -Wall -c addvec.c -o addvec_pic.o
        gcc $(CF1) -m32 -Wall -c multvec.c -o multvec_pic.o
        ar rcs libvec_pic.a addvec_pic.o multvec_pic.o

libvec_nopic.a : addvec.c multvec.c
        gcc $(CF2) -m32 -Wall -c addvec.c -o addvec_nopic.o
        gcc $(CF2) -m32 -Wall -c multvec.c -o multvec_nopic.o
        ar rcs libvec_nopic.a addvec_nopic.o multvec_nopic.o

libvec_noplt.a : addvec.c multvec.c
        gcc $(CF3) -m32 -Wall -c addvec.c -o addvec_noplt.o
        gcc $(CF3) -m32 -Wall -c multvec.c -o multvec_noplt.o
        ar rcs libvec_noplt.a addvec_noplt.o multvec_noplt.o
```

Makefile (4)

```
#-------------------------------------------------------------
case0 : main.c libvec.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF0)-o vector_0.out main.o ./libvec.a

case1 : main.c libvec_pic.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF0)-o vector_1_pic.out main.o ./libvec_pic.a

case2 : main.c libvec_nopic.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF0)-o vector_2_nopic.out main.o ./libvec_nopic.a

case3 : main.c libvec_noplt.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF0) -o vector_3_noplt.out main.o ./libvec_noplt.a
```

Makefile (5)

```
#-------------------------------------------------------------
case4 : main.c libvec.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF1) -o vector_4_static.out main.o ./libvec.a

case5 : main.c libvec_pic.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF1) -o vector_5_pic_static.out main.o ./libvec_pic.a

case6 : main.c libvec_nopic.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF1) -o vector_6_nopic_static.out main.o ./libvec_nopic.a

case7 : main.c libvec_noplt.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF1) -o vector_7_noplt_static.out main.o ./libvec_noplt.a
```

Makefile (6)

```
#--------------------------------------------------------------
case8 : main.c libvec.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF2) -o vector_8_nopie.out main.o ./libvec.a

case9 : main.c libvec_pic.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF2) -o vector_9_pic_nopie.out main.o ./libvec_pic.a

casea : main.c libvec_nopic.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF2) -o vector_a_nopic_nopie.out main.o ./libvec_nopic.a

caseb : main.c libvec_noplt.a
        gcc -m32 -Wall -c main.c
        gcc -m32 $(LF2) -o vector_b_noplt_nopie.out main.o ./libvec_noplt.a
```

**

analyzing commands

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

addvec.o (1)

objdump -dr addvec.o

```
00000000 <addvec>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   53                      push   %ebx
   4:   83 ec 10                sub    $0x10,%esp
   7:   e8 fc ff ff ff          call   8 <addvec+0x8>
                        8: R_386_PC32   __x86.get_pc_thunk.ax
   c:   05 01 00 00 00          add    $0x1,%eax
                        d: R_386_GOTPC  _GLOBAL_OFFSET_TABLE_
  11:   c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
  18:   eb 39                   jmp    53 <addvec+0x53>
  1a:   8b 45 f8                mov    -0x8(%ebp),%eax
  1d:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
  24:   8b 45 08                mov    0x8(%ebp),%eax
  27:   01 d0                   add    %edx,%eax
  29:   8b 08                   mov    (%eax),%ecx
  2b:   8b 45 f8                mov    -0x8(%ebp),%eax
  2e:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
  35:   8b 45 0c                mov    0xc(%ebp),%eax
  38:   01 d0                   add    %edx,%eax
  3a:   8b 10                   mov    (%eax),%edx
  3c:   8b 45 f8                mov    -0x8(%ebp),%eax
```

addvec.o (2)

objdump -dr addvec.o

```
  3f:   8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
  46:   8b 45 10                mov    0x10(%ebp),%eax
  49:   01 d8                   add    %ebx,%eax
  4b:   01 ca                   add    %ecx,%edx
  4d:   89 10                   mov    %edx,(%eax)
  4f:   83 45 f8 01             addl   $0x1,-0x8(%ebp)
  53:   8b 45 f8                mov    -0x8(%ebp),%eax
  56:   3b 45 14                cmp    0x14(%ebp),%eax
  59:   7c bf                   jl     1a <addvec+0x1a>
  5b:   90                      nop
  5c:   83 c4 10                add    $0x10,%esp
  5f:   5b                      pop    %ebx
  60:   5d                      pop    %ebp
  61:   c3                      ret

Desensamblado de la sección .text.__x86.get_pc_thunk.ax:

00000000 <__x86.get_pc_thunk.ax>:
   0:   8b 04 24                mov    (%esp),%eax
   3:   c3                      ret
```

addvec_pic.o with -fPIC (1)

objdump -dr addvec_pic.o

```
00000000 <addvec>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   53                      push   %ebx
   4:   83 ec 10                sub    $0x10,%esp
   7:   e8 fc ff ff ff          call   8 <addvec+0x8>
                        8: R_386_PC32   __x86.get_pc_thunk.ax
   c:   05 01 00 00 00          add    $0x1,%eax
                        d: R_386_GOTPC  _GLOBAL_OFFSET_TABLE_
  11:   c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
  18:   eb 39                   jmp    53 <addvec+0x53>
  1a:   8b 45 f8                mov    -0x8(%ebp),%eax
  1d:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
  24:   8b 45 08                mov    0x8(%ebp),%eax
  27:   01 d0                   add    %edx,%eax
  29:   8b 08                   mov    (%eax),%ecx
  2b:   8b 45 f8                mov    -0x8(%ebp),%eax
  2e:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
  35:   8b 45 0c                mov    0xc(%ebp),%eax
  38:   01 d0                   add    %edx,%eax
  3a:   8b 10                   mov    (%eax),%edx
  3c:   8b 45 f8                mov    -0x8(%ebp),%eax
```

addvec_pic.o with -fPIC (2)

objdump -dr addvec_pic.o

```
  3f:   8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
  46:   8b 45 10                mov    0x10(%ebp),%eax
  49:   01 d8                   add    %ebx,%eax
  4b:   01 ca                   add    %ecx,%edx
  4d:   89 10                   mov    %edx,(%eax)
  4f:   83 45 f8 01             addl   $0x1,-0x8(%ebp)
  53:   8b 45 f8                mov    -0x8(%ebp),%eax
  56:   3b 45 14                cmp    0x14(%ebp),%eax
  59:   7c bf                   jl     1a <addvec+0x1a>
  5b:   90                      nop
  5c:   83 c4 10                add    $0x10,%esp
  5f:   5b                      pop    %ebx
  60:   5d                      pop    %ebp
  61:   c3                      ret

Desensamblado de la sección .text.__x86.get_pc_thunk.ax:

00000000 <__x86.get_pc_thunk.ax>:
   0:   8b 04 24                mov    (%esp),%eax
   3:   c3                      ret
```

addvec_nopic.o with -fno-pic (1)

objdump -dr addvec_nopic.o

```
00000000 <addvec>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   53                      push   %ebx
   4:   83 ec 10                sub    $0x10,%esp
   7:   c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
   e:   eb 39                   jmp    49 <addvec+0x49>
  10:   8b 45 f8                mov    -0x8(%ebp),%eax
  13:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
  1a:   8b 45 08                mov    0x8(%ebp),%eax
  1d:   01 d0                   add    %edx,%eax
  1f:   8b 08                   mov    (%eax),%ecx
  21:   8b 45 f8                mov    -0x8(%ebp),%eax
  24:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
  2b:   8b 45 0c                mov    0xc(%ebp),%eax
  2e:   01 d0                   add    %edx,%eax
  30:   8b 10                   mov    (%eax),%edx
  32:   8b 45 f8                mov    -0x8(%ebp),%eax
```

addvec_nopic.o with -fno-pic (2)

objdump -dr addvec_nopic.o

```
35:   8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
3c:   8b 45 10                mov    0x10(%ebp),%eax
3f:   01 d8                   add    %ebx,%eax
41:   01 ca                   add    %ecx,%edx
43:   89 10                   mov    %edx,(%eax)
45:   83 45 f8 01             addl   $0x1,-0x8(%ebp)
49:   8b 45 f8                mov    -0x8(%ebp),%eax
4c:   3b 45 14                cmp    0x14(%ebp),%eax
4f:   7c bf                   jl     10 <addvec+0x10>
51:   90                      nop
52:   83 c4 10                add    $0x10,%esp
55:   5b                      pop    %ebx
56:   5d                      pop    %ebp
57:   c3                      ret
```

addvec_noplt.o with -fno-plt (1)

objdump -dr addvec_noplt.o

```
00000000 <addvec>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   53                      push   %ebx
   4:   83 ec 10                sub    $0x10,%esp
   7:   e8 fc ff ff ff          call   8 <addvec+0x8>
                        8: R_386_PC32   __x86.get_pc_thunk.ax
   c:   05 01 00 00 00          add    $0x1,%eax
                        d: R_386_GOTPC  _GLOBAL_OFFSET_TABLE_
  11:   c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
  18:   eb 39                   jmp    53 <addvec+0x53>
  1a:   8b 45 f8                mov    -0x8(%ebp),%eax
  1d:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
  24:   8b 45 08                mov    0x8(%ebp),%eax
  27:   01 d0                   add    %edx,%eax
  29:   8b 08                   mov    (%eax),%ecx
  2b:   8b 45 f8                mov    -0x8(%ebp),%eax
  2e:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
  35:   8b 45 0c                mov    0xc(%ebp),%eax
  38:   01 d0                   add    %edx,%eax
  3a:   8b 10                   mov    (%eax),%edx
  3c:   8b 45 f8                mov    -0x8(%ebp),%eax
```

addvec_noplt.o with -fno-plt (2)

objdump -dr addvec_noplt.o

```
  3f:   8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
  46:   8b 45 10                mov    0x10(%ebp),%eax
  49:   01 d8                   add    %ebx,%eax
  4b:   01 ca                   add    %ecx,%edx
  4d:   89 10                   mov    %edx,(%eax)
  4f:   83 45 f8 01             addl   $0x1,-0x8(%ebp)
  53:   8b 45 f8                mov    -0x8(%ebp),%eax
  56:   3b 45 14                cmp    0x14(%ebp),%eax
  59:   7c bf                   jl     1a <addvec+0x1a>
  5b:   90                      nop
  5c:   83 c4 10                add    $0x10,%esp
  5f:   5b                      pop    %ebx
  60:   5d                      pop    %ebp
  61:   c3                      ret

Desensamblado de la sección .text.__x86.get_pc_thunk.ax:

00000000 <__x86.get_pc_thunk.ax>:
   0:   8b 04 24                mov    (%esp),%eax
   3:   c3                      ret
```

case 0: addvec in vector_0.out (1)

objdump -d addvec_0.out

```
0000058a <addvec>:
 58a:   55                      push   %ebp
 58b:   89 e5                   mov    %esp,%ebp
 58d:   53                      push   %ebx
 58e:   83 ec 10                sub    $0x10,%esp
 591:   e8 56 00 00 00          call   5ec <__x86.get_pc_thunk.ax>
 596:   05 42 1a 00 00          add    $0x1a42,%eax
 59b:   c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 5a2:   eb 39                   jmp    5dd <addvec+0x53>
 5a4:   8b 45 f8                mov    -0x8(%ebp),%eax
 5a7:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 5ae:   8b 45 08                mov    0x8(%ebp),%eax
 5b1:   01 d0                   add    %edx,%eax
 5b3:   8b 08                   mov    (%eax),%ecx
 5b5:   8b 45 f8                mov    -0x8(%ebp),%eax
 5b8:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 5bf:   8b 45 0c                mov    0xc(%ebp),%eax
 5c2:   01 d0                   add    %edx,%eax
 5c4:   8b 10                   mov    (%eax),%edx
 5c6:   8b 45 f8                mov    -0x8(%ebp),%eax
```

case 0: addvec in vector_0.out (2)

objdump -d addvec_0.out

```
5c9:    8d 1c 85 00 00 00 00     lea     0x0(,%eax,4),%ebx
5d0:    8b 45 10                 mov     0x10(%ebp),%eax
5d3:    01 d8                    add     %ebx,%eax
5d5:    01 ca                    add     %ecx,%edx
5d7:    89 10                    mov     %edx,(%eax)
5d9:    83 45 f8 01              addl    $0x1,-0x8(%ebp)
5dd:    8b 45 f8                 mov     -0x8(%ebp),%eax
5e0:    3b 45 14                 cmp     0x14(%ebp),%eax
5e3:    7c bf                    jl      5a4 <addvec+0x1a>
5e5:    90                       nop
5e6:    83 c4 10                 add     $0x10,%esp
5e9:    5b                       pop     %ebx
5ea:    5d                       pop     %ebp
5eb:    c3                       ret
```

case 1: addvec in vector_1_pic.out (1)

objdump -d addvec_1_pic.out

```
0000058a <addvec>:
 58a:   55                      push   %ebp
 58b:   89 e5                   mov    %esp,%ebp
 58d:   53                      push   %ebx
 58e:   83 ec 10                sub    $0x10,%esp
 591:   e8 56 00 00 00          call   5ec <__x86.get_pc_thunk.ax>
 596:   05 42 1a 00 00          add    $0x1a42,%eax
 59b:   c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 5a2:   eb 39                   jmp    5dd <addvec+0x53>
 5a4:   8b 45 f8                mov    -0x8(%ebp),%eax
 5a7:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 5ae:   8b 45 08                mov    0x8(%ebp),%eax
 5b1:   01 d0                   add    %edx,%eax
 5b3:   8b 08                   mov    (%eax),%ecx
 5b5:   8b 45 f8                mov    -0x8(%ebp),%eax
 5b8:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 5bf:   8b 45 0c                mov    0xc(%ebp),%eax
 5c2:   01 d0                   add    %edx,%eax
 5c4:   8b 10                   mov    (%eax),%edx
 5c6:   8b 45 f8                mov    -0x8(%ebp),%eax
```

case 1: addvec in vector_1_pic.out (2)

objdump -d addvec_1_pic.out

```
5c9:   8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
5d0:   8b 45 10                mov    0x10(%ebp),%eax
5d3:   01 d8                   add    %ebx,%eax
5d5:   01 ca                   add    %ecx,%edx
5d7:   89 10                   mov    %edx,(%eax)
5d9:   83 45 f8 01             addl   $0x1,-0x8(%ebp)
5dd:   8b 45 f8                mov    -0x8(%ebp),%eax
5e0:   3b 45 14                cmp    0x14(%ebp),%eax
5e3:   7c bf                   jl     5a4 <addvec+0x1a>
5e5:   90                      nop
5e6:   83 c4 10                add    $0x10,%esp
5e9:   5b                      pop    %ebx
5ea:   5d                      pop    %ebp
5eb:   c3                      ret
```

case 2: addvec in vector_2_nopic.out (1)

objdump -d addvec_2_nopic.out

```
0000058a <addvec>:
 58a:   55                      push   %ebp
 58b:   89 e5                   mov    %esp,%ebp
 58d:   53                      push   %ebx
 58e:   83 ec 10                sub    $0x10,%esp
 591:   c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 598:   eb 39                   jmp    5d3 <addvec+0x49>
 59a:   8b 45 f8                mov    -0x8(%ebp),%eax
 59d:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 5a4:   8b 45 08                mov    0x8(%ebp),%eax
 5a7:   01 d0                   add    %edx,%eax
 5a9:   8b 08                   mov    (%eax),%ecx
 5ab:   8b 45 f8                mov    -0x8(%ebp),%eax
 5ae:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 5b5:   8b 45 0c                mov    0xc(%ebp),%eax
 5b8:   01 d0                   add    %edx,%eax
 5ba:   8b 10                   mov    (%eax),%edx
 5bc:   8b 45 f8                mov    -0x8(%ebp),%eax
```

case 2: addvec in vector_2_nopic.out (2)

objdump -d addvec_2_nopic.out

```
5bf:   8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
5c6:   8b 45 10                mov    0x10(%ebp),%eax
5c9:   01 d8                   add    %ebx,%eax
5cb:   01 ca                   add    %ecx,%edx
5cd:   89 10                   mov    %edx,(%eax)
5cf:   83 45 f8 01             addl   $0x1,-0x8(%ebp)
5d3:   8b 45 f8                mov    -0x8(%ebp),%eax
5d6:   3b 45 14                cmp    0x14(%ebp),%eax
5d9:   7c bf                   jl     59a <addvec+0x10>
5db:   90                      nop
5dc:   83 c4 10                add    $0x10,%esp
5df:   5b                      pop    %ebx
5e0:   5d                      pop    %ebp
5e1:   c3                      ret
5e2:   66 90                   xchg   %ax,%ax
5e4:   66 90                   xchg   %ax,%ax
5e6:   66 90                   xchg   %ax,%ax
5e8:   66 90                   xchg   %ax,%ax
5ea:   66 90                   xchg   %ax,%ax
5ec:   66 90                   xchg   %ax,%ax
5ee:   66 90                   xchg   %ax,%ax
```

case 3: addvec in vector_3_noplt.out (1)

objdump -d addvec_3_noplt.out

```
0000058a <addvec>:
 58a:   55                      push   %ebp
 58b:   89 e5                   mov    %esp,%ebp
 58d:   53                      push   %ebx
 58e:   83 ec 10                sub    $0x10,%esp
 591:   e8 56 00 00 00          call   5ec <__x86.get_pc_thunk.ax>
 596:   05 42 1a 00 00          add    $0x1a42,%eax
 59b:   c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 5a2:   eb 39                   jmp    5dd <addvec+0x53>
 5a4:   8b 45 f8                mov    -0x8(%ebp),%eax
 5a7:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 5ae:   8b 45 08                mov    0x8(%ebp),%eax
 5b1:   01 d0                   add    %edx,%eax
 5b3:   8b 08                   mov    (%eax),%ecx
 5b5:   8b 45 f8                mov    -0x8(%ebp),%eax
 5b8:   8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 5bf:   8b 45 0c                mov    0xc(%ebp),%eax
 5c2:   01 d0                   add    %edx,%eax
 5c4:   8b 10                   mov    (%eax),%edx
 5c6:   8b 45 f8                mov    -0x8(%ebp),%eax
```

case 3: addvec in vector_3_noplt.out (2)

objdump -d addvec_3_noplt.out

```
5c9:   8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
5d0:   8b 45 10                mov    0x10(%ebp),%eax
5d3:   01 d8                   add    %ebx,%eax
5d5:   01 ca                   add    %ecx,%edx
5d7:   89 10                   mov    %edx,(%eax)
5d9:   83 45 f8 01             addl   $0x1,-0x8(%ebp)
5dd:   8b 45 f8                mov    -0x8(%ebp),%eax
5e0:   3b 45 14                cmp    0x14(%ebp),%eax
5e3:   7c bf                   jl     5a4 <addvec+0x1a>
5e5:   90                      nop
5e6:   83 c4 10                add    $0x10,%esp
5e9:   5b                      pop    %ebx
5ea:   5d                      pop    %ebp
5eb:   c3                      ret
```

case 4: addvec in vector_4_static.out (1)

objdump -d addvec_4_static.out

```
08048912 <addvec>:
 8048912:       55                      push   %ebp
 8048913:       89 e5                   mov    %esp,%ebp
 8048915:       53                      push   %ebx
 8048916:       83 ec 10                sub    $0x10,%esp
 8048919:       e8 56 00 00 00          call   8048974 <__x86.get_pc_thunk.ax>
 804891e:       05 e2 06 09 00          add    $0x906e2,%eax
 8048923:       c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 804892a:       eb 39                   jmp    8048965 <addvec+0x53>
 804892c:       8b 45 f8                mov    -0x8(%ebp),%eax
 804892f:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 8048936:       8b 45 08                mov    0x8(%ebp),%eax
 8048939:       01 d0                   add    %edx,%eax
 804893b:       8b 08                   mov    (%eax),%ecx
 804893d:       8b 45 f8                mov    -0x8(%ebp),%eax
 8048940:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 8048947:       8b 45 0c                mov    0xc(%ebp),%eax
 804894a:       01 d0                   add    %edx,%eax
 804894c:       8b 10                   mov    (%eax),%edx
 804894e:       8b 45 f8                mov    -0x8(%ebp),%eax
```

case 4: addvec in vector_4_static.out (2)

objdump -d addvec_4_static.out

```
8048951:       8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
8048958:       8b 45 10                mov    0x10(%ebp),%eax
804895b:       01 d8                   add    %ebx,%eax
804895d:       01 ca                   add    %ecx,%edx
804895f:       89 10                   mov    %edx,(%eax)
8048961:       83 45 f8 01             addl   $0x1,-0x8(%ebp)
8048965:       8b 45 f8                mov    -0x8(%ebp),%eax
8048968:       3b 45 14                cmp    0x14(%ebp),%eax
804896b:       7c bf                   jl     804892c <addvec+0x1a>
804896d:       90                      nop
804896e:       83 c4 10                add    $0x10,%esp
8048971:       5b                      pop    %ebx
8048972:       5d                      pop    %ebp
8048973:       c3                      ret
```

case 5: addvec in vector_5_pic_static.out (1)

objdump -d addvec_5_pic_static.out

```
08048912 <addvec>:
 8048912:       55                      push   %ebp
 8048913:       89 e5                   mov    %esp,%ebp
 8048915:       53                      push   %ebx
 8048916:       83 ec 10                sub    $0x10,%esp
 8048919:       e8 56 00 00 00          call   8048974 <__x86.get_pc_thunk.ax>
 804891e:       05 e2 06 09 00          add    $0x906e2,%eax
 8048923:       c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 804892a:       eb 39                   jmp    8048965 <addvec+0x53>
 804892c:       8b 45 f8                mov    -0x8(%ebp),%eax
 804892f:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 8048936:       8b 45 08                mov    0x8(%ebp),%eax
 8048939:       01 d0                   add    %edx,%eax
 804893b:       8b 08                   mov    (%eax),%ecx
 804893d:       8b 45 f8                mov    -0x8(%ebp),%eax
 8048940:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 8048947:       8b 45 0c                mov    0xc(%ebp),%eax
 804894a:       01 d0                   add    %edx,%eax
 804894c:       8b 10                   mov    (%eax),%edx
 804894e:       8b 45 f8                mov    -0x8(%ebp),%eax
```

case 5: addvec in vector_5_pic_static.out (2)

objdump -d addvec_5_pic_static.out

```
8048951:       8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
8048958:       8b 45 10                mov    0x10(%ebp),%eax
804895b:       01 d8                   add    %ebx,%eax
804895d:       01 ca                   add    %ecx,%edx
804895f:       89 10                   mov    %edx,(%eax)
8048961:       83 45 f8 01             addl   $0x1,-0x8(%ebp)
8048965:       8b 45 f8                mov    -0x8(%ebp),%eax
8048968:       3b 45 14                cmp    0x14(%ebp),%eax
804896b:       7c bf                   jl     804892c <addvec+0x1a>
804896d:       90                      nop
804896e:       83 c4 10                add    $0x10,%esp
8048971:       5b                      pop    %ebx
8048972:       5d                      pop    %ebp
8048973:       c3                      ret
```

case 6: addvec in vector_6_nopic_static.out (1)

objdump -d addvec_6_nopic_static.out

```
08048912 <addvec>:
 8048912:       55                      push   %ebp
 8048913:       89 e5                   mov    %esp,%ebp
 8048915:       53                      push   %ebx
 8048916:       83 ec 10                sub    $0x10,%esp
 8048919:       c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 8048920:       eb 39                   jmp    804895b <addvec+0x49>
 8048922:       8b 45 f8                mov    -0x8(%ebp),%eax
 8048925:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 804892c:       8b 45 08                mov    0x8(%ebp),%eax
 804892f:       01 d0                   add    %edx,%eax
 8048931:       8b 08                   mov    (%eax),%ecx
 8048933:       8b 45 f8                mov    -0x8(%ebp),%eax
 8048936:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 804893d:       8b 45 0c                mov    0xc(%ebp),%eax
 8048940:       01 d0                   add    %edx,%eax
 8048942:       8b 10                   mov    (%eax),%edx
 8048944:       8b 45 f8                mov    -0x8(%ebp),%eax
```

case 6: addvec in vector_6_nopic_static.out (2)

objdump -d addvec_6_nopic_static.out

```
8048947:      8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
804894e:      8b 45 10                mov    0x10(%ebp),%eax
8048951:      01 d8                   add    %ebx,%eax
8048953:      01 ca                   add    %ecx,%edx
8048955:      89 10                   mov    %edx,(%eax)
8048957:      83 45 f8 01             addl   $0x1,-0x8(%ebp)
804895b:      8b 45 f8                mov    -0x8(%ebp),%eax
804895e:      3b 45 14                cmp    0x14(%ebp),%eax
8048961:      7c bf                   jl     8048922 <addvec+0x10>
8048963:      90                      nop
8048964:      83 c4 10                add    $0x10,%esp
8048967:      5b                      pop    %ebx
8048968:      5d                      pop    %ebp
8048969:      c3                      ret
804896a:      66 90                   xchg   %ax,%ax
804896c:      66 90                   xchg   %ax,%ax
804896e:      66 90                   xchg   %ax,%ax
```

case 7: addvec in vector_7_noplt_static.out (1)

objdump -d addvec_7_noplt_static.out

```
08048912 <addvec>:
 8048912:       55                      push   %ebp
 8048913:       89 e5                   mov    %esp,%ebp
 8048915:       53                      push   %ebx
 8048916:       83 ec 10                sub    $0x10,%esp
 8048919:       e8 56 00 00 00          call   8048974 <__x86.get_pc_thunk.ax>
 804891e:       05 e2 06 09 00          add    $0x906e2,%eax
 8048923:       c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 804892a:       eb 39                   jmp    8048965 <addvec+0x53>
 804892c:       8b 45 f8                mov    -0x8(%ebp),%eax
 804892f:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 8048936:       8b 45 08                mov    0x8(%ebp),%eax
 8048939:       01 d0                   add    %edx,%eax
 804893b:       8b 08                   mov    (%eax),%ecx
 804893d:       8b 45 f8                mov    -0x8(%ebp),%eax
 8048940:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 8048947:       8b 45 0c                mov    0xc(%ebp),%eax
 804894a:       01 d0                   add    %edx,%eax
 804894c:       8b 10                   mov    (%eax),%edx
 804894e:       8b 45 f8                mov    -0x8(%ebp),%eax
```

case 7: addvec in vector_7_noplt_static.out (2)

objdump -d addvec_7_noplt_static.out

```
8048951:       8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
8048958:       8b 45 10                mov    0x10(%ebp),%eax
804895b:       01 d8                   add    %ebx,%eax
804895d:       01 ca                   add    %ecx,%edx
804895f:       89 10                   mov    %edx,(%eax)
8048961:       83 45 f8 01             addl   $0x1,-0x8(%ebp)
8048965:       8b 45 f8                mov    -0x8(%ebp),%eax
8048968:       3b 45 14                cmp    0x14(%ebp),%eax
804896b:       7c bf                   jl     804892c <addvec+0x1a>
804896d:       90                      nop
804896e:       83 c4 10                add    $0x10,%esp
8048971:       5b                      pop    %ebx
8048972:       5d                      pop    %ebp
8048973:       c3                      ret
```

case 8: addvec in vector_8_nopie.out (1)

objdump -d nets_8_nopie.out

```
08048493 <addvec>:
 8048493:       55                      push   %ebp
 8048494:       89 e5                   mov    %esp,%ebp
 8048496:       53                      push   %ebx
 8048497:       83 ec 10                sub    $0x10,%esp
 804849a:       e8 56 00 00 00          call   80484f5 <__x86.get_pc_thunk.ax>
 804849f:       05 61 1b 00 00          add    $0x1b61,%eax
 80484a4:       c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 80484ab:       eb 39                   jmp    80484e6 <addvec+0x53>
 80484ad:       8b 45 f8                mov    -0x8(%ebp),%eax
 80484b0:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 80484b7:       8b 45 08                mov    0x8(%ebp),%eax
 80484ba:       01 d0                   add    %edx,%eax
 80484bc:       8b 08                   mov    (%eax),%ecx
 80484be:       8b 45 f8                mov    -0x8(%ebp),%eax
 80484c1:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 80484c8:       8b 45 0c                mov    0xc(%ebp),%eax
 80484cb:       01 d0                   add    %edx,%eax
 80484cd:       8b 10                   mov    (%eax),%edx
 80484cf:       8b 45 f8                mov    -0x8(%ebp),%eax
```

case 8: addvec in vector_8_nopie.out (2)

objdump -d nets_8_nopie.out

```
80484d2:      8d 1c 85 00 00 00 00    lea     0x0(,%eax,4),%ebx
80484d9:      8b 45 10                mov     0x10(%ebp),%eax
80484dc:      01 d8                   add     %ebx,%eax
80484de:      01 ca                   add     %ecx,%edx
80484e0:      89 10                   mov     %edx,(%eax)
80484e2:      83 45 f8 01             addl    $0x1,-0x8(%ebp)
80484e6:      8b 45 f8                mov     -0x8(%ebp),%eax
80484e9:      3b 45 14                cmp     0x14(%ebp),%eax
80484ec:      7c bf                   jl      80484ad <addvec+0x1a>
80484ee:      90                      nop
80484ef:      83 c4 10                add     $0x10,%esp
80484f2:      5b                      pop     %ebx
80484f3:      5d                      pop     %ebp
80484f4:      c3                      ret
```

case 9: addvec in vector_9_pic_nopie.out (1)

objdump -d addvec_9_pic_nopie.out

```
08048493 <addvec>:
 8048493:       55                      push   %ebp
 8048494:       89 e5                   mov    %esp,%ebp
 8048496:       53                      push   %ebx
 8048497:       83 ec 10                sub    $0x10,%esp
 804849a:       e8 56 00 00 00          call   80484f5 <__x86.get_pc_thunk.ax>
 804849f:       05 61 1b 00 00          add    $0x1b61,%eax
 80484a4:       c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 80484ab:       eb 39                   jmp    80484e6 <addvec+0x53>
 80484ad:       8b 45 f8                mov    -0x8(%ebp),%eax
 80484b0:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 80484b7:       8b 45 08                mov    0x8(%ebp),%eax
 80484ba:       01 d0                   add    %edx,%eax
 80484bc:       8b 08                   mov    (%eax),%ecx
 80484be:       8b 45 f8                mov    -0x8(%ebp),%eax
 80484c1:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 80484c8:       8b 45 0c                mov    0xc(%ebp),%eax
 80484cb:       01 d0                   add    %edx,%eax
 80484cd:       8b 10                   mov    (%eax),%edx
 80484cf:       8b 45 f8                mov    -0x8(%ebp),%eax
```

case 9: addvec in vector_9_pic_nopie.out (2)

objdump -d addvec_9_pic_nopie.out

```
80484d2:        8d 1c 85 00 00 00 00    lea     0x0(,%eax,4),%ebx
80484d9:        8b 45 10                mov     0x10(%ebp),%eax
80484dc:        01 d8                   add     %ebx,%eax
80484de:        01 ca                   add     %ecx,%edx
80484e0:        89 10                   mov     %edx,(%eax)
80484e2:        83 45 f8 01             addl    $0x1,-0x8(%ebp)
80484e6:        8b 45 f8                mov     -0x8(%ebp),%eax
80484e9:        3b 45 14                cmp     0x14(%ebp),%eax
80484ec:        7c bf                   jl      80484ad <addvec+0x1a>
80484ee:        90                      nop
80484ef:        83 c4 10                add     $0x10,%esp
80484f2:        5b                      pop     %ebx
80484f3:        5d                      pop     %ebp
80484f4:        c3                      ret
```

case 10: addvec in vector_10$_{n}opic\_nopie.out(1)$

objdump -d addvec_10_nopic_nopie.out

```
08048493 <addvec>:
 8048493:      55                     push   %ebp
 8048494:      89 e5                  mov    %esp,%ebp
 8048496:      53                     push   %ebx
 8048497:      83 ec 10               sub    $0x10,%esp
 804849a:      c7 45 f8 00 00 00 00   movl   $0x0,-0x8(%ebp)
 80484a1:      eb 39                  jmp    80484dc <addvec+0x49>
 80484a3:      8b 45 f8               mov    -0x8(%ebp),%eax
 80484a6:      8d 14 85 00 00 00 00   lea    0x0(,%eax,4),%edx
 80484ad:      8b 45 08               mov    0x8(%ebp),%eax
 80484b0:      01 d0                  add    %edx,%eax
 80484b2:      8b 08                  mov    (%eax),%ecx
 80484b4:      8b 45 f8               mov    -0x8(%ebp),%eax
 80484b7:      8d 14 85 00 00 00 00   lea    0x0(,%eax,4),%edx
 80484be:      8b 45 0c               mov    0xc(%ebp),%eax
 80484c1:      01 d0                  add    %edx,%eax
 80484c3:      8b 10                  mov    (%eax),%edx
 80484c5:      8b 45 f8               mov    -0x8(%ebp),%eax
```

case 10: addvec in vector_10$_{n}opic\_nopie.out$(2)

objdump -d addvec_10_nopic_nopie.out

```
80484c8:     8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
80484cf:     8b 45 10                mov    0x10(%ebp),%eax
80484d2:     01 d8                   add    %ebx,%eax
80484d4:     01 ca                   add    %ecx,%edx
80484d6:     89 10                   mov    %edx,(%eax)
80484d8:     83 45 f8 01             addl   $0x1,-0x8(%ebp)
80484dc:     8b 45 f8                mov    -0x8(%ebp),%eax
80484df:     3b 45 14                cmp    0x14(%ebp),%eax
80484e2:     7c bf                   jl     80484a3 <addvec+0x10>
80484e4:     90                      nop
80484e5:     83 c4 10                add    $0x10,%esp
80484e8:     5b                      pop    %ebx
80484e9:     5d                      pop    %ebp
80484ea:     c3                      ret
80484eb:     66 90                   xchg   %ax,%ax
80484ed:     66 90                   xchg   %ax,%ax
80484ef:     90                      nop
```

case 11: addvec in vector_11_noplt_nopie.out (1)

objdump -d addvec_11_noplt_nopie.out

```
08048493 <addvec>:
 8048493:       55                      push   %ebp
 8048494:       89 e5                   mov    %esp,%ebp
 8048496:       53                      push   %ebx
 8048497:       83 ec 10                sub    $0x10,%esp
 804849a:       e8 56 00 00 00          call   80484f5 <__x86.get_pc_thunk.ax>
 804849f:       05 61 1b 00 00          add    $0x1b61,%eax
 80484a4:       c7 45 f8 00 00 00 00    movl   $0x0,-0x8(%ebp)
 80484ab:       eb 39                   jmp    80484e6 <addvec+0x53>
 80484ad:       8b 45 f8                mov    -0x8(%ebp),%eax
 80484b0:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 80484b7:       8b 45 08                mov    0x8(%ebp),%eax
 80484ba:       01 d0                   add    %edx,%eax
 80484bc:       8b 08                   mov    (%eax),%ecx
 80484be:       8b 45 f8                mov    -0x8(%ebp),%eax
 80484c1:       8d 14 85 00 00 00 00    lea    0x0(,%eax,4),%edx
 80484c8:       8b 45 0c                mov    0xc(%ebp),%eax
 80484cb:       01 d0                   add    %edx,%eax
 80484cd:       8b 10                   mov    (%eax),%edx
 80484cf:       8b 45 f8                mov    -0x8(%ebp),%eax
```

case 11: addvec in vector_11_noplt_nopie.out (2)

objdump -d addvec_11_noplt_nopie.out

```
80484d2:     8d 1c 85 00 00 00 00    lea    0x0(,%eax,4),%ebx
80484d9:     8b 45 10                mov    0x10(%ebp),%eax
80484dc:     01 d8                   add    %ebx,%eax
80484de:     01 ca                   add    %ecx,%edx
80484e0:     89 10                   mov    %edx,(%eax)
80484e2:     83 45 f8 01             addl   $0x1,-0x8(%ebp)
80484e6:     8b 45 f8                mov    -0x8(%ebp),%eax
80484e9:     3b 45 14                cmp    0x14(%ebp),%eax
80484ec:     7c bf                   jl     80484ad <addvec+0x1a>
80484ee:     90                      nop
80484ef:     83 c4 10                add    $0x10,%esp
80484f2:     5b                      pop    %ebx
80484f3:     5d                      pop    %ebp
80484f4:     c3                      ret
```

case 0: addvec in vector$_0$.*out*

```
objdump -d addvec_0.out
```

case 0: addvec in vector$_0$.*out*

```
objdump -d addvec_0.out
```