

ELF1 7 Examples - 5 Executable run_dynamic - ELF Study 1999

Young W. Lim

2020-02-28 Fri

- 1 Based on
- 2 Summary of relocation results for `run_dynamic`
 - TOC
 - 1. Reloc summary for `run_dynamic`
 - 2. Symbols and sections for `run_dynamic`
 - 3. Relocation listings for `run_dynamic`
- 3 Linking for `run_dynamic`
 - TOC
 - Linking the `.text` section for `run_dynamic`
 - Undefined symbols in `run_dynamic`
- 4 Locating relocs and symbol references of `run_dynamic`
 - TOC
 - Locating `.text` section relocs of `librel.so`
 - Locating `.text` section symbol references of `librel.so`

"Study of ELF loading and relocs", 1999

http://netwinder.osuosl.org/users/p/patb/public_html/elf_relocs.html

I, the copyright holder of this work, hereby publish it under the following licenses: GNU head Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled GNU Free Documentation License.

CC BY SA This file is licensed under the Creative Commons Attribution ShareAlike 3.0 Unported License. In short: you are free to share and make derivative works of the file under the conditions that you appropriately attribute it, and that you distribute it only under a license compatible with this one.

Compiling 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`
- `-Wl,-q`

TOC: Summary of relocation results for `run_dynamic`

- 1 Reloc summary for `run_dynamic`
- 2 Symbols and sections for `librel.so`

TOC: 1. run_dynamic shared object file relocs

- Relocation listing sections for a shared library
- Relocation table section for `run_dynamic` executable
- Relocation listing section for `run_dyamic` executable
- a) `data` section relocs of `run_dynamic` executable
- b) `text` section relocs of `run_dynamic` executable
- c) `data` section reloc listing of `run_dynamic` executable
- d) `text` section reloc listing of `run_dynamic` executable

Relocation sections for executable files

- based on "Study of ELF loading and relocs"

<code>.rel.bss</code>	<code>R_386_COPY</code>	<i>non-PIC</i> reference of a global symbol
<code>.rel.got</code>	<code>R_386_GLOB_DAT</code>	<i>PIC</i> reference of a global symbol
<code>.rel.plt</code>	<code>R_386_JUMP_SLOT</code>	<i>PIC</i> reference of a function symbol

- from the results of `readelf -r`

<code>.rel.dyn</code>	<code>R_386_COPY</code>	<i>non-PIC</i> reference of a global symbol
	<code>R_386_GLOB_DAT</code>	<i>PIC</i> reference of a global symbol
<code>.rel.plt</code>	<code>R_386_JUMP_SLOT</code>	<i>PIC</i> reference of a function symbol

http://netwinder.osuosl.org/users/p/patb/public_html/elf_relocs.html

Relocation table sections for run_dynamic executable

- for run_dynamic

	-fno-pic	default	-fPIC
.plt	✓	✓	✓
.plt.got	✓	✓	✓
.got	✓	✓	✓
.got.plt			

```
readelf -t run-fno-pic | grep -e .plt -e .got -e .rel
```


Relocation listing sections for run_dynamic executable

- for run_dynamic
 - * only when linked with -Wl,-q
 - ✓ regardless of -Wl,-q

	-fno-pic	default	-fPIC
.rel.data	*	*	*
.rel.text	*	*	*
.rel.dyn	✓	✓	✓
.rel.plt	✓	✓	✓
.rel.got			

```
readelf -t run-fno-pic | grep -e .plt -e .got -e .rel
```

a) `text` section dynamic relocs of `run_dynamic` executable

- dynamic relocation sections (`.rel.dyn`, `.rel.plt`)
 - global data symbol reference (cPub) in `.text`
 - when GOT is used (default, `-fPIC`)
`R_386_GOT32` in `.text` → `R_386_GLOB_DAT` in `.got`
 - otherwise (`-fno-pic`)
`R_386_32` in `.text` →
`R_386_32` in `.text` , `R_386_COPY` in `.bss`
 - global function symbol reference (fPub) in `.text`
 - when PLT is used (default, `-fPIC`)
`R_386_PLT32` in `.plt` → `R_386_JUMP_SLOT` in `.got`
 - otherwise (`-fno-pic`) PLT is used to access shared library functions
`R_386_PC32` in `.text` →
`R_386_PC32` in `.text` , `R_386_JUMP_SLOT` in `.got`

b) `text` section normal relocs of `run_dynamic` executable

- normal relocation section (`.rel.text`)
 - global data symbol reference (cPub)
 - when GOT is used (default, `-fPIC`)
`R_386_GOT32` in `.text`
 - otherwise (`-fno-pic`)
`R_386_32` in `.text`
 - global function symbol reference (fPub)
 - when PLT is used (default, `-fPIC`)
`R_386_PLT32` in `.text`
 - otherwise (`-fno-pic`)
`R_386_PC32` in `.text`

c) `text` section `dynamic` reloc `listing` of `run_dynamic`

- `text` section related listing of `.rel.dyn`

	-fno-pic	default	-fPIC
cPub	<code>R_386_32</code> in <code>.text</code> <code>R_386_COPY</code> in <code>.bss</code>	<code>R_386_GLOB_DAT</code> in <code>.got</code>	<code>R_386_GLOB_DAT</code> in <code>.got</code>
fPub	<code>R_386_PC32</code> in <code>.text</code>	not applicable	not applicable

- `text` section related listing of `.rel.plt`

	-fno-pic	default	-fPIC
fPub	<code>R_386_JUMP_SLOT</code> in <code>.got</code>	<code>R_386_JUMP_SLOT</code> in <code>.got</code>	<code>R_386_JUMP_SLOT</code> in <code>.got</code>

d) `text` section normal reloc listing of `run_dynamic`

- `text` section related listing of `.rel.text`
 - exists only when compiled with `-Wl,-q`
 - the same as the `rel.text` of `main.o`
 - `ld -q (--emit-relocs)` :
leave relocation sections and contents in fully linked executables.

	<code>-fno-pic</code>	<code>default</code>	<code>-fPIC</code>
<code>cPub</code>	<code>R_386_32</code> in <code>.text</code>	<code>R_386_GOT32x</code> in <code>.text</code>	<code>R_386_GOT32x</code> in <code>.text</code>
<code>fPub</code>	<code>R_386_PC32</code> in <code>.text</code>	<code>R_386_PLT32</code> in <code>.text</code>	<code>R_386_PLT32</code> in <code>.text</code>

- `fPub` is defined in other module (`rel.o`)

TOC: 2. Symbols and sections for `run_dynamic`

- `-fno-pic` case
 - (1.a) Symbol table in `run_dynamic` (`-fno-pic`)
 - (1.b) Section header in `run_dynamic` (`-fno-pic`)
 - (1.c) Symbol's section listing in `run_dynamic` (`-fno-pic`)
 - (1.d) Zero value symbol listing in `run_dynamic` (`-fno-pic`)
- default case
 - (2.a) Symbol table in `run_dynamic` (default)
 - (2.b) Section header in `run_dynamic` (default)
 - (2.c) Symbol's section listing in `run_dynamic` (default)
 - (2.d) Zero value symbol listing in `run_dynamic` (default)
- `-fPIC` case
 - (3.a) Symbol table in `run_dynamic` (`-fPIC`)
 - (3.b) Section header in `run_dynamic` (`-fPIC`)
 - (3.c) Symbol's section listing in `run_dynamic` (`-fPIC`)
 - (2.d) Zero value symbol listing in `run_dynamic` (`-fPIC`)

(1.a) Symbol table in `run_dynamic` (-fno-pic)

```
young@USys2:~$ readelf -s run-fno-pic
```

```
Symbol table '.dynsym' contains 14 entries:
```

6:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub
11:	00002008	1	OBJECT	GLOBAL	DEFAULT	32	cPub

```
Symbol table '.symtab' contains 69 entries:
```

63:	000005fd	47	FUNC	GLOBAL	DEFAULT	15	main
64:	00002008	1	OBJECT	GLOBAL	DEFAULT	32	cPub
67:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

(1.b) Section header in `run_dynamic` (-fno-pic)

```
young@USys2:~$ readelf -S run-fno-pic
```

[9]	.rel.dyn	REL	000003e8	0003e8	000058	08	A	5	0	4
[10]	.rel.plt	REL	00000440	000440	000010	08	AI	5	28	4
[13]	.plt	PROGBITS	00000480	000480	000030	04	AX	0	0	16
[14]	.plt.got	PROGBITS	000004b0	0004b0	000010	08	AX	0	0	8
[15]	.text	PROGBITS	000004c0	0004c0	0001d2	00	AX	0	0	16
[16]	.rel.text	REL	00000000	0016dc	0000f0	08	I	34	15	4
[27]	.dynamic	DYNAMIC	00001ed0	000ed0	000108	08	WA	6	0	4
[28]	.got	PROGBITS	00001fd8	000fd8	000028	04	WA	0	0	4
[29]	.data	PROGBITS	00002000	001000	000008	00	WA	0	0	4
[30]	.rel.data	REL	00000000	00180c	000008	08	I	34	29	4
[32]	.bss	NOBITS	00002008	001008	000004	00	WA	0	0	1

<https://stackoverflow.com/questions/1685483/how-can-i-examine-contents-of-a-data->

(1.c) Symbol's section listing in `run_dynamic` (-fno-pic)

```
young@USys2:~$ readelf -s run-fno-pic
```

```
Symbol table '.dynsym' contains 14 entries:
```

6:	00000000	[]	0	FUNC	GLOBAL	DEFAULT	UND	fPub
11:	00002008	[.data]	1	OBJECT	GLOBAL	DEFAULT	32	cPub

```
Symbol table '.symtab' contains 69 entries:
```

63:	000005fd	[.text]	47	FUNC	GLOBAL	DEFAULT	15	main
64:	00002008	[.bss]	1	OBJECT	GLOBAL	DEFAULT	32	cPub
67:	00000000	[]	0	FUNC	GLOBAL	DEFAULT	UND	fPub

(1.d) Zero value symbol listing in `run_dynamic` (-fno-pic)

```
young@USys2:~$ readelf -s run-fno-pic
```

```
Symbol table '.dynsym' contains 14 entries:
```

0:	00000000	0	NOTYPE	LOCAL	DEFAULT	UND	
1:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_deregisterTMCloneTab
2:	00000000	0	FUNC	WEAK	DEFAULT	UND	__cxa_finalize@GLIBC_2.1.3 (2)
3:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	__gmon_start__
4:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	__libc_start_main@GLIBC_2.0 (3)
5:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_registerTMCloneTable
6:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

```
Symbol table '.symtab' contains 69 entries:
```

0:	00000000	0	NOTYPE	LOCAL	DEFAULT	UND	
26:	00000000	0	SECTION	LOCAL	DEFAULT	33	
27:	00000000	0	FILE	LOCAL	DEFAULT	ABS	crtstuff.c
36:	00000000	0	FILE	LOCAL	DEFAULT	ABS	main.c
37:	00000000	0	FILE	LOCAL	DEFAULT	ABS	crtstuff.c
39:	00000000	0	FILE	LOCAL	DEFAULT	ABS	
46:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_deregisterTMCloneTab
52:	00000000	0	FUNC	WEAK	DEFAULT	UND	__cxa_finalize@@GLIBC_2.1
54:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	__gmon_start__
57:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	__libc_start_main@@GLIBC_
66:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_registerTMCloneTable
67:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

(2.a) Symbol table in `run_dynamic` (default)

```
young@USys2:~$ readelf -s run-default
```

```
Symbol table '.dynsym' contains 14 entries:
```

5:	00000000	0	OBJECT	GLOBAL	DEFAULT	UND	cPub
7:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

```
Symbol table '.symtab' contains 69 entries:
```

63:	000005dd	61	FUNC	GLOBAL	DEFAULT	15	main
64:	00000000	0	OBJECT	GLOBAL	DEFAULT	UND	cPub
67:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

(2.b) Section header in `run_dynamic` (default)

```
young@USys2:~$ readelf -S run-default
```

```
Encabezados de Sección:
```

[5]	.dynsym	DYNSYM	000001e8	0001e8	0000e0	10	A	6	1	4
[6]	.dynstr	STRTAB	000002c8	0002c8	0000ce	00	A	0	0	1
[9]	.rel.dyn	REL	000003e4	0003e4	000048	08	A	5	0	4
[10]	.rel.plt	REL	0000042c	00042c	000010	08	AI	5	28	4
[13]	.plt	PROGBITS	00000460	000460	000030	04	AX	0	0	16
[14]	.plt.got	PROGBITS	00000490	000490	000010	08	AX	0	0	8
[15]	.text	PROGBITS	000004a0	0004a0	0001e2	00	AX	0	0	16
[16]	.rel.text	REL	00000000	0016dc	000100	08	I	34	15	4
[27]	.dynamic	DYNAMIC	00001ed4	000ed4	000100	08	WA	6	0	4
[28]	.got	PROGBITS	00001fd4	000fd4	00002c	04	WA	0	0	4
[29]	.data	PROGBITS	00002000	001000	000008	00	WA	0	0	4
[30]	.rel.data	REL	00000000	001824	000008	08	I	34	29	4
[32]	.bss	NOBITS	00002008	001008	000004	00	WA	0	0	1

<https://stackoverflow.com/questions/1685483/how-can-i-examine-contents-of-a-data->

(2.c) Symbol's section listing in `run_dynamic` (default)

```
young@USys2:~$ readelf -s run-default
```

```
Symbol table '.dynsym' contains 14 entries:
```

```
 5: 00000000 [    ]    0 OBJECT GLOBAL DEFAULT UND cPub
 7: 00000000 [    ]    0 FUNC   GLOBAL DEFAULT UND fPub
```

```
Symbol table '.symtab' contains 69 entries:
```

```
63: 000005dd [ .text ]   61 FUNC   GLOBAL DEFAULT  15 main
64: 00000000 [    ]    0 OBJECT GLOBAL DEFAULT UND cPub
67: 00000000 [    ]    0 FUNC   GLOBAL DEFAULT UND fPub
```

(2.d) Zero value symbol listing in `run_dynamic` (default)

```
young@USys2:~$ readelf -s run-default
```

```
Symbol table '.dynsym' contains 14 entries:
```

0:	00000000	0	NOTYPE	LOCAL	DEFAULT	UND	
1:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_deregisterTMCloneTab
2:	00000000	0	FUNC	WEAK	DEFAULT	UND	__cxa_finalize@GLIBC_2.1.3 (2)
3:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	__gmon_start__
4:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	__libc_start_main@GLIBC_2.0 (3)
5:	00000000	0	OBJECT	GLOBAL	DEFAULT	UND	cPub
6:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_registerTMCloneTable
7:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

```
Symbol table '.symtab' contains 69 entries:
```

0:	00000000	0	NOTYPE	LOCAL	DEFAULT	UND	
26:	00000000	0	SECTION	LOCAL	DEFAULT	33	
27:	00000000	0	FILE	LOCAL	DEFAULT	ABS	crtstuff.c
36:	00000000	0	FILE	LOCAL	DEFAULT	ABS	main.c
37:	00000000	0	FILE	LOCAL	DEFAULT	ABS	crtstuff.c
39:	00000000	0	FILE	LOCAL	DEFAULT	ABS	
46:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_deregisterTMCloneTab
52:	00000000	0	FUNC	WEAK	DEFAULT	UND	__cxa_finalize@@GLIBC_2.1
54:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	__gmon_start__
57:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	__libc_start_main@@GLIBC_
64:	00000000	0	OBJECT	GLOBAL	DEFAULT	UND	cPub
66:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_registerTMCloneTable
67:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

(3.a) Symbol table in `run_dynamic` (-fPIC)

```
young@USys2:~$ readelf -s run-fPIC
```

```
Symbol table '.dynsym' contains 14 entries:
```

5:	00000000	0	OBJECT	GLOBAL	DEFAULT	UND	cPub
7:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

```
Symbol table '.symtab' contains 69 entries:
```

63:	000005dd	61	FUNC	GLOBAL	DEFAULT	15	main
64:	00000000	0	OBJECT	GLOBAL	DEFAULT	UND	cPub
67:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

(3.b) Section header in `run_dynamic` (-fPIC)

```
young@USys2:~$ readelf -S run-fPIC
```

```
Encabezados de Sección:
```

[9]	.rel.dyn	REL	000003e4	0003e4	000048 08	A	5	0	4
[10]	.rel.plt	REL	0000042c	00042c	000010 08	AI	5	28	4
[13]	.plt	PROGBITS	00000460	000460	000030 04	AX	0	0	16
[14]	.plt.got	PROGBITS	00000490	000490	000010 08	AX	0	0	8
[15]	.text	PROGBITS	000004a0	0004a0	0001e2 00	AX	0	0	16
[16]	.rel.text	REL	00000000	0016dc	000100 08	I	34	15	4
[27]	.dynamic	DYNAMIC	00001ed4	000ed4	000100 08	WA	6	0	4
[28]	.got	PROGBITS	00001fd4	000fd4	00002c 04	WA	0	0	4
[29]	.data	PROGBITS	00002000	001000	000008 00	WA	0	0	4
[30]	.rel.data	REL	00000000	001824	000008 08	I	34	29	4
[32]	.bss	NOBITS	00002008	001008	000004 00	WA	0	0	1

<https://stackoverflow.com/questions/1685483/how-can-i-examine-contents-of-a-data->

(3.c) Symbol's section listing in `run_dynamic` (-fPIC)

```
young@USys2:~$ readelf -s run-fPIC
```

```
Symbol table '.dynsym' contains 14 entries:
```

```
 5: 00000000 [    ]    0 OBJECT  GLOBAL DEFAULT  UND cPub
 7: 00000000 [    ]    0 FUNC    GLOBAL DEFAULT  UND fPub
```

```
Symbol table '.symtab' contains 69 entries:
```

```
63: 000005dd [ .text ]   61 FUNC    GLOBAL DEFAULT   15 main
64: 00000000 [    ]    0 OBJECT  GLOBAL DEFAULT  UND cPub
67: 00000000 [    ]    0 FUNC    GLOBAL DEFAULT  UND fPub
```

(3.d) Zero value symbol listing in `run_dynamic` (-fPIC)

```
young@USys2:~$ readelf -s run-fPIC
```

```
Symbol table '.dynsym' contains 14 entries:
```

0:	00000000	0	NOTYPE	LOCAL	DEFAULT	UND	
1:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_deregisterTMCloneTab
2:	00000000	0	FUNC	WEAK	DEFAULT	UND	__cxa_finalize@GLIBC_2.1.3 (2)
3:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	__gmon_start__
4:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	__libc_start_main@GLIBC_2.0 (3)
5:	00000000	0	OBJECT	GLOBAL	DEFAULT	UND	cPub
6:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_registerTMCloneTable
7:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

```
Symbol table '.symtab' contains 69 entries:
```

0:	00000000	0	NOTYPE	LOCAL	DEFAULT	UND	
26:	00000000	0	SECTION	LOCAL	DEFAULT	33	
27:	00000000	0	FILE	LOCAL	DEFAULT	ABS	crtstuff.c
36:	00000000	0	FILE	LOCAL	DEFAULT	ABS	main.c
37:	00000000	0	FILE	LOCAL	DEFAULT	ABS	crtstuff.c
39:	00000000	0	FILE	LOCAL	DEFAULT	ABS	
46:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_deregisterTMCloneTab
52:	00000000	0	FUNC	WEAK	DEFAULT	UND	__cxa_finalize@@GLIBC_2.1
54:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	__gmon_start__
57:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	__libc_start_main@@GLIBC_
64:	00000000	0	OBJECT	GLOBAL	DEFAULT	UND	cPub
66:	00000000	0	NOTYPE	WEAK	DEFAULT	UND	_ITM_registerTMCloneTable
67:	00000000	0	FUNC	GLOBAL	DEFAULT	UND	fPub

TOC: 3. Relocation listings for `run_dynamic`

- `-fno-pic` case
 - (1.a) Symbol table in `run_dynamic` (`-fno-pic`)
 - (1.b) Section header in `run_dynamic` (`-fno-pic`)
 - (1.c) Reloc Info field of `run_dynamic` (`-fno-pic`)
 - (1.d) Zero value symbols of `run_dynamic` (`-fno-pic`)
- default case
 - (2.a) Symbol table in `run_dynamic` (default)
 - (2.b) Section header in `run_dynamic` (default)
 - (2.c) Reloc Info field of `run_dynamic` (default)
 - (2.d) Zero value symbols of `run_dynamic` (default)
- `-fPIC` case
 - (3.a) Symbol table in `run_dynamic` (`-fPIC`)
 - (3.b) Section header in `run_dynamic` (`-fPIC`)
 - (3.c) Reloc Info field of `run_dynamic` (`-fPIC`)
 - (3.d) Zero value symbols of `run_dynamic` (`-fPIC`)

(1.a) Relocs of `run_dynamic` (no-PIC)

La sección de reubicación `'.rel.dyn'` at offset `0x3e8` contains 11 entries:

```
00001ec8 [000000] [08] R_386_RELATIVE
00001ecc [000000] [08] R_386_RELATIVE
00001ff8 [000000] [08] R_386_RELATIVE    ... .got    00001fd8
00002004 [000000] [08] R_386_RELATIVE    ... .data  00002000
00000614 [000006] [02] R_386_PC32      00000000   fPub
0000061e [00000b] [01] R_386_32      00002008   cPub
00002008 [00000b] [05] R_386_COPY     00002008   cPub    ... .bss  00002008
```

La sección de reubicación `'.rel.plt'` at offset `0x440` contains 2 entries:

```
00001fe8 00000607 R_386_JUMP_SLOT 00000000   fPub
```

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 30 entries:

```
000004e8 [00003f] [03] R_386_GOT32    000005fd   main
00000614 [000043] [02] R_386_PC32    00000000   fPub
0000061e [000040] [01] R_386_32      00002008   cPub
```

`gcc -Wl,-q` is used

(1.b) Reloc sections of `run_dynamic` (no-PIC)

La sección de reubicación `'rel.dyn'` at offset `0x3e8` contains 11 entries:

```
00001ec8 [      ] R_386_RELATIVE
00001ecc [      ] R_386_RELATIVE
00001ff8 [.got ] R_386_RELATIVE    ... .got    00001fd8
00002004 [.data] R_386_RELATIVE    ... .data   00002000
00000614 [.text] R_386_PC32        00000000    fPub
0000061e [.text] R_386_32         00002008    cPub
00002008 [,bss ] R_386_COPY        00002008    cPub    ... .bss 00002008
```

La sección de reubicación `'rel.plt'` at offset `0x440` contains 2 entries:

```
00001fe8 00000607 R_386_JUMP_SLOT    00000000    fPub
```

La sección de reubicación `'rel.text'` at offset `0x16dc` contains 30 entries:

```
000004e8 [.text] R_386_GOT32        000005fd    main
00000614 [.text] R_386_PC32        00000000    fPub
0000061e [.text] R_386_32         00002008    cPub
```

`gcc -Wl,-q` is used

(1.c) Reloc Info field of `run_dynamic` (no-PIC)

La sección de reubicación `'.rel.dyn'` at offset `0x3e8` contains 11 entries:

```
00001ec8 [000000] [08] R_386_RELATIVE
00001ecc [000000] [08] R_386_RELATIVE
00001ff8 [000000] [08] R_386_RELATIVE
00002004 [000000] [08] R_386_RELATIVE
00000614 [000006] fPub [02] R_386_PC32 00000000
0000061e [00000b] cPub [01] R_386_32 00002008
00002008 [00000b] cPub [05] R_386_COPY 00002008
```

La sección de reubicación `'.rel.plt'` at offset `0x440` contains 2 entries:

```
00001fe8 [000006] fPub [07] R_386_JUMP_SLOT 00000000
```

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 30 entries:

```
000004e8 [00003f] main [03] R_386_GOT32 000005fd
00000614 [000043] fPub [02] R_386_PC32 00000000
0000061e [000040] cPub [01] R_386_32 00002008
```

`gcc -Wl,-q` is used

(1.d) Zero value symbols of `run_dynamic` (no-PIC)

La sección de reubicación `'.rel.dyn'` at offset `0x3e8` contains 11 entries:
00000614 [000006][02] R_386_PC32 00000000 fPub ... in [.text]

La sección de reubicación `'.rel.plt'` at offset `0x440` contains 2 entries:
00001fe8 [000006][07] R_386_JUMP_SLOT 00000000 fPub ... in [.got]

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 30 entries:
00000614 [000043][02] R_386_PC32 00000000 fPub ... in [.text]

`gcc -Wl,-q` is used

(2.a) Relocs of `run_dynamic` (default)

La sección de reubicación `'.rel.dyn'` at offset `0x3e4` contains 9 entries:

```
00001ecc [000000] [08] R_386_RELATIVE
00001ed0 [000000] [08] R_386_RELATIVE
00001ff4 [000000] [08] R_386_RELATIVE    ... .got    00001fd4
00002004 [000000] [08] R_386_RELATIVE    ... .data  00002000
00001ff8 [000005] [06] R_386_GLOB_DAT    00000000    cPub
```

La sección de reubicación `'.rel.plt'` at offset `0x42c` contains 2 entries:

```
00001fe4 00000707 R_386_JUMP_SLOT    00000000    fPub
```

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 32 entries:

```
000004c8 [00003f] [03] R_386_GOT32        000005dd    main
00000594 [000019] [09] R_386_GOTOFF      00002008    .bss
000005bd [000019] [09] R_386_GOTOFF      00002008    .bss
000005fd [000043] [04] R_386_PLT32       00000000    fPub
00000608 [000040] [2b] R_386_GOT32X      00000000    cPub
```

`gcc -Wl,-q` is used

(2.b) Reloc sections of `run_dynamic` (default)

La sección de reubicación `'.rel.dyn'` at offset `0x3e4` contains 9 entries:

```
00001ecc  [] R_386_RELATIVE
00001ed0  [] R_386_RELATIVE
00001ff4  [.got ] R_386_RELATIVE    ... .got    00001fd4
00002004  [.data] R_386_RELATIVE    ... .data   00002000
00001ff8  [.got ] R_386_GLOB_DAT    00000000   cPub
```

La sección de reubicación `'.rel.plt'` at offset `0x42c` contains 2 entries:

```
00001fe4  [.got ] R_386_JUMP_SLOT    00000000   fPub
```

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 32 entries:

```
000004c8  [.text] R_386_GOT32        000005dd   main
00000594  [.text] R_386_GOTOFF      00002008   .bss
000005bd  [,text] R_386_GOTOFF      00002008   .bss
000005fd  [.text] R_386_PLT32       00000000   fPub
00000608  [.text] R_386_GOT32X      00000000   cPub
```

`gcc -Wl,-q` is used

(2.c) Reloc Info field of `run_dynamic` (default)

La sección de reubicación `'.rel.dyn'` at offset `0x3e4` contains 9 entries:

```
00001ecc [000000] [08] R_386_RELATIVE
00001ed0 [000000] [08] R_386_RELATIVE
00001ff4 [000000] [08] R_386_RELATIVE
00002004 [000000] [08] R_386_RELATIVE
00001ff8 [000005] cPub [06] R_386_GLOB_DAT 00000000
```

La sección de reubicación `'.rel.plt'` at offset `0x42c` contains 2 entries:

```
00001fe4 [000007] fPub [07] R_386_JUMP_SLOT 00000000
```

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 32 entries:

```
000004c8 [00003f] main [03] R_386_GOT32 000005dd
00000594 [000019] .bss [09] R_386_GOTOFF 00002008
000005bd [000019] .bss [09] R_386_GOTOFF 00002008
000005fd [000043] fPub [04] R_386_PLT32 00000000
00000608 [000040] cPub [2b] R_386_GOT32X 00000000
```

`gcc -Wl,-q` is used

(2.d) Zero value symbols of `run_dynamic` (default)

La sección de reubicación `'.rel.dyn'` at offset `0x3e4` contains 9 entries:

```
00001ff8 [000005][06] R_386_GLOB_DAT 00000000 cPub ... in [.got]
```

La sección de reubicación `'.rel.plt'` at offset `0x42c` contains 2 entries:

```
00001fe4 [000007][07] R_386_JUMP_SLOT 00000000 fPub ... in [.got]
```

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 32 entries:

```
000005fd [000043][04] R_386_PLT32 00000000 fPub ... in [.text]
```

```
00000608 [000040][2b] R_386_GOT32X 00000000 cPub ... in [.text]
```

```
gcc -Wl,-q is used
```

(3.a) Relocs of `run_dynamic` (PIC)

La sección de reubicación `'.rel.dyn'` at offset `0x3e4` contains 9 entries:

```
00001ecc [000000] [08] R_386_RELATIVE
00001ed0 [000000] [08] R_386_RELATIVE
00001ff4 [000000] [08] R_386_RELATIVE    ... .got    00001fd4
00002004 [000000] [08] R_386_RELATIVE    ... .data  00002000
00001ff8 [000005] [06] R_386_GLOB_DAT    00000000    cPub
```

La sección de reubicación `'.rel.plt'` at offset `0x42c` contains 2 entries:

```
00001fe4 00000707 R_386_JUMP_SLOT    00000000    fPub
```

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 32 entries:

```
000004c8 [00003f] [03] R_386_GOT32        000005dd    main
00000594 [000019] [09] R_386_GOTOFF      00002008    .bss
000005bd [000019] [09] R_386_GOTOFF      00002008    .bss
000005fd [000043] [04] R_386_PLT32       00000000    fPub
00000608 [000040] [2b] R_386_GOT32X      00000000    cPub
```

`gcc -Wl,-q` is used

(3.b) Reloc sections of `run_dynamic` (PIC)

La sección de reubicación `'.rel.dyn'` at offset `0x3e4` contains 9 entries:

```
00001ecc [      ] R_386_RELATIVE
00001ed0 [      ] R_386_RELATIVE
00001ff4 [ .got ] R_386_RELATIVE    ... .got    00001fd4
00002004 [ .data] R_386_RELATIVE    ... .data   00002000
00001ff8 [ .got ] R_386_GLOB_DAT    00000000   cPub
```

La sección de reubicación `'.rel.plt'` at offset `0x42c` contains 2 entries:

```
00001fe4 00000707 R_386_JUMP_SLOT 00000000   fPub
```

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 32 entries:

```
000004c8 [ .text] R_386_GOT32      000005dd   main
00000594 [ .text] R_386_GOTOFF    00002008   .bss
000005bd [ .text] R_386_GOTOFF    00002008   .bss
000005fd [ .text] R_386_PLT32     00000000   fPub
00000608 [ .text] R_386_GOT32X    00000000   cPub
```

`gcc -Wl,-q` is used

(3.c) Relocs Info field of `run_dynamic` (PIC)

La sección de reubicación `'.rel.dyn'` at offset `0x3e4` contains 9 entries:

```
00001ecc [000000] [08] R_386_RELATIVE
00001ed0 [000000] [08] R_386_RELATIVE
00001ff4 [000000] [08] R_386_RELATIVE
00002004 [000000] [08] R_386_RELATIVE
00001ff8 [000005] cPub [06] R_386_GLOB_DAT 00000000
```

La sección de reubicación `'.rel.plt'` at offset `0x42c` contains 2 entries:

```
00001fe4 [000007] fPub [07] R_386_JUMP_SLOT 00000000
```

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 32 entries:

```
000004c8 [00003f] main [03] R_386_GOT32 000005dd
00000594 [000019] .bss [09] R_386_GOTOFF 00002008
000005bd [000019] .bss [09] R_386_GOTOFF 00002008
000005fd [000043] fPub [04] R_386_PLT32 00000000
00000608 [000040] cPub [2b] R_386_GOT32X 00000000
```

`gcc -Wl,-q` is used

(3.d) Zero value symbols of `run_dynamic` (PIC)

La sección de reubicación `'.rel.dyn'` at offset `0x3e4` contains 9 entries:

```
00001ff8 [000005][06] R_386_GLOB_DAT 00000000 cPub ... in [.got]
```

La sección de reubicación `'.rel.plt'` at offset `0x42c` contains 2 entries:

```
00001fe4 [000007][07] R_386_JUMP_SLOT 00000000 fPub ... in [.got]
```

La sección de reubicación `'.rel.text'` at offset `0x16dc` contains 32 entries:

```
000005fd [000043][04] R_386_PLT32 00000000 fPub ... in [.text]
```

```
00000608 [000040][2b] R_386_GOT32X 00000000 cPub ... in [.text]
```

```
gcc -Wl,-q is used
```

TOC: Linking for `run_dynamic`

- Linking the `.text` section for `run_dynamic`
- Undefined symbols in `run_dynamic`

TOC: linking the .text section

- resolving global function symbol references fPub(123)
- resolving global data symbol references cPub
- (1) resolving global symbol references (non-PIC)
- (2) resolving global symbol references (PIE)
- (3) resolving global symbol references (PIC)

```
int main() {  
    return fPub(123)  
        + cPub;  
}
```

resolving global function symbol references : fPub(123)

- the **PIC** reloc of a global function reference in `.text` section will cause the linker to add a **PLT entry** and a corresponding **GOT entry**
 - the reloc of `fPub(123)` is translated into a **indirect call** through the **PLT entry**
 - the **GOT entry** gets a **R_386_JUMP_SLOT** reloc using the symbol `fPub`

http://netwinder.osuosl.org/users/p/patb/public_html/elf_relocs.html

resolving global data symbol references : cPub

- the **PIC** relocs of a global data symbol reference in `.text` section will cause the linker to add a **GOT entry** to hold them
- the relocs at `&cPub` (address) and `cPub` (data) will have an **GOT entry** to hold `&cPub`
 - the symbol value is an address of the symbol
- the **GOT entry** is marked with a **R_386_GLOB_DAT** reloc asking the dynamic linker for the full 32-bit absolute address

http://netwinder.osuosl.org/users/p/patb/public_html/elf_relocs.html

(1) resolving global symbol references (non-PIC)

- for a non-PIC

- cPub reference in .text section has **R_386_32** reloc
cPub reference in .bss section has **R_386_COPY** reloc

```
[readelf -r]
0000061e 00000b01 R_386_32          00002008  cPub ... copy to address
00002008 00000b05 R_386_COPY        00002008  cPub ... copy to address
reference at 61e in .text has a value at 2008 in .bss (=copy to address)
```

- fPub call in .text section has **R_386_PC32** reloc
fPub call in .got section has **R_386_JUMP_SLOT** reloc

```
[readelf -r]
00000614 00000602 R_386_PC32        00000000  fPub
00001fe8 00000607 R_386_JUMP_SLOT  00000000  fPub
function call at 602 in .text may result in call to <fPub@plt> at 4a0
```

- the **dynamic linker** will store at the reloc target
the full 32-bit absolute and relative addresses

http://netwinder.osuosl.org/users/p/patb/public_html/elf_relocs.html

(2) resolving global symbol references (PIE)

- for a PIE (default)

- cPub reference in `.text` section has `R_386_GOT32` reloc
→ `R_386_GLOB_DAT` in `.got`

```
[readelf -r]
00001ff8 00000506 R_386_GLOB_DAT 00000000 cPub
symbol value of cPub is stored at 1ff8 in .got
```

- fPub call in `.text` section has `R_386_PLT32` reloc
→ `R_386_JUMP_SLOT` in `.got`

```
[readelf -r]
00001fe4 00000707 R_386_JUMP_SLOT 00000000 fPub
symbol value of fPub is stored at 1fe4 in .got
```

- the `PLT` is used
because `fPub` is defined in the other module (`rel.c`)

http://netwinder.osuosl.org/users/p/patb/public_html/elf_relocs.html

(3) resolving global symbol references (PIC)

- for a PIC
 - cPub reference in `.text` section has `R_386_GOT32` reloc
→ `R_386_GLOB_DAT` in `.got`

```
[readelf -r]
00001ff8 00000506 R_386_GLOB_DAT 00000000 cPub
symbol value of cPub is stored at 1ff8 in .got
```

- fPub call in `.text` section has `R_386_PLT32` reloc
→ `R_386_JUMP_SLOT` in `.got`

```
[readelf -r]
00001fe4 00000707 R_386_JUMP_SLOT 00000000 fPub
symbol value of fPub is stored at 1fe4 in .got
```

http://netwinder.osuosl.org/users/p/patb/public_html/elf_relocs.html

TOC: Undefined symbols in `run_dynamic`

- Undefined symbols
- Undefined symbols in an executable
- Undefined symbols in a dynamic executable
- Undefined weak symbols in a dynamic executable
- Undefined weak symbols

Undefined symbols

- after all of the input files have been read and all **symbol resolution** is complete, the **link-editor** searches the internal **symbol table** for any **unbound** symbol references
 - that have not been bound to **symbol definitions**
 - these are referred to as **undefined symbols**

<https://docs.oracle.com/cd/E19120-01/open.solaris/819-0690/chapter2-9/index.html>

Undefined symbols in an executable

- when generating an **executable** output file, by default, the **link-editor** terminates with an appropriate **error** message if there is any **undefined symbol**
 - when a symbol reference in a relocatable object is never matched to any symbol definition.

<https://docs.oracle.com/cd/E19120-01/open.solaris/819-0690/chapter2-9/index.html>

Undefined symbols in a dynamic executable

- Symbols can remain undefined when a symbol reference in a relocatable object is bound to a symbol definition in an implicitly defined **shared object**
- the implicit reference must be made explicit by referencing the library directly during the **link-edit** of an executable

<https://docs.oracle.com/cd/E19120-01/open.solaris/819-0690/chapter2-9/index.html>

Undefined Weak symbols (1)

- **Weak** symbol references that remain unresolved, do not result in a fatal error condition, no matter what output file type is being generated.
- for static executable the symbol is converted to an absolute symbol with an assigned value of zero.
- for a dynamic executable or shared object the symbol is left as an undefined weak reference with an assigned value of zero.

<https://docs.oracle.com/cd/E19120-01/open.solaris/819-0690/chapter2-11/index.html>

Undefined Weak symbols (2)

- During process execution, the **runtime linker** searches for this symbol.
- If the runtime linker does not find a match, the reference is bound to an address of zero instead of generating a fatal relocation error.

<https://docs.oracle.com/cd/E19120-01/open.solaris/819-0690/chapter2-11/index.html>

Undefined Weak symbols in a dynamic executable

- Weak symbol references that remain unresolved, do not result in a fatal error condition, no matter what output file type is being generated.

'`.dynsym`' and '`.symtab`'

```
?: 00000000      0 NOTYPE WEAK  DEFAULT UND _ITM_deregisterTMCloneTab
?: 00000000      0 FUNC   WEAK  DEFAULT UND __cxa_finalize@GLIBC_2.1.3 (2)
?: 00000000      0 NOTYPE WEAK  DEFAULT UND __gmon_start__
?: 00000000      0 FUNC   GLOBAL DEFAULT UND __libc_start_main@GLIBC_2.0 (3)
?: 00000000      0 NOTYPE WEAK  DEFAULT UND _ITM_registerTMCloneTable
```

Relocs for undefined weak symbols selected

```
[.got] 00000106 R_386_GLOB_DAT 00000000 _ITM_deregisterTMClone
[.got] 00000206 R_386_GLOB_DAT 00000000 __cxa_finalize@GLIBC_2.1.3
[.got] 00000306 R_386_GLOB_DAT 00000000 __gmon_start__
[.got] 00000506 R_386_GLOB_DAT 00000000 _ITM_registerTMCloneTa
[.got] 00000407 R_386_JUMP_SLOT 00000000 __libc_start_main@GLIBC_2.0
```

<https://docs.oracle.com/cd/E19120-01/open.solaris/819-0690/chapter2-11/index.html>

TOC: Locating relocs and symbol references of `run_dynamic`

- Locating `.text` section relocs of `run_dynamic`
- Locating `.text` section symbol references of `run_dynamic`

TOC: Relocations in an executable file

- Finding .text section relocs (-fno-PIC)
- Finding .text section relocs (default)
- Finding .text section relocs (-fPIC)
- Locating R_386_JUMP_SLOT relocs in .plt section (-fPIC)
- Locating R_386_GLOB_DAT relocs in .got section (-fPIC)

```
int main() {  
    return fPub(123)    // 1  global function symbol reference  
        + cPub;        // 2  global data symbol reference  
}
```

Finding .text section relocs (no-PIC) for run_dynamic

```
[readelf -S]
```

```
[15] .text          PROGBITS          000004c0 0004c0 0001d2 00  AX  0   0 16  
Address: 000004c0 Size: 0001d2 ---> [4c0, 691]
```

```
[readelf - r]
```

```
00000614 00000602 R_386_PC32          00000000  fPub .... .text 000004c0  
0000061e 00000b01 R_386_32           00002008  cPub .... .text 000004c0
```


Finding .text section relocs (default) for run_dynamic

```
[readelf -S]
```

```
[15] .text          PROGBITS          000004a0 0004a0 0001e2 00  AX  0   0 16  
Address: 000004a0 Size: 0001e2 ---> [3a0, 681]
```

```
[28] .got            PROGBITS          00001fd4 000fd4 00002c 04  WA  0   0  4  
Address: 00001fd4 Size: 00002c ---> [1fd4, 1fff]
```

```
[readelf - r]
```

```
00001fe4 00000707 R_386_JUMP_SLOT 00000000  fPub .... .got 00001fd4  
00001ff8 00000506 R_386_GLOB_DAT  00000000  cPub .... .got 00001fd4
```

Finding .text section relocs (-fPIC) for run_dynamic

```
[readelf -S]
```

```
[15] .text          PROGBITS          000004a0 0004a0 0001e2 00  AX  0   0 16  
Address: 000004a0 Size: 0001e2 ---> [4a0, 681]
```

```
[28] .got             PROGBITS          00001fd4 000fd4 00002c 04  WA  0   0  4  
Address: 00001fd4 Size: 00002c ---> [1fec, 1fff]
```

```
[readelf - r]
```

```
00001fe4 00000707 R_386_JUMP_SLOT 00000000  fPub .... .got 00001fd4  
00001ff8 00000506 R_386_GLOB_DAT  00000000  cPub .... .got 00001fd4
```

Locating R_386_JUMP_SLOT relocs in .plt section (-fPIC)

- .plt section address

```
[readelf -S]
.plt = 00000460
```

- symbol value

```
[readelf -s]
fPub = 00000000 (UND)
```

- R_386_JUMP_SLOT relocs in .rel.dyn

```
[readelf -r]
00001fe4 00000707 R_386_JUMP_SLOT 00000000 fPub
```

- hexadumps of .got section

```
[objdump -s -j .got.plt]
1fe4 86040000 00000000 00000000 00000000 .....
---> 1fe4 00000486
```

- hexadumps of .plt section

```
[objdump -dr]
00000480 <fPub@plt>:
480: ff a3 10 00 00 00      jmp     *0x10(%ebx)
486: 68 08 00 00 00      push   $0x8
48b: e9 d0 ff ff ff      jmp     460 <.plt>
```

Locating R_386_GLOB_DAT relocs in .got section (-fPIC)

- .got section address

```
[readelf -S]
.got = 00001fd4
```

- symbol value

```
[readelf -s]
cPub = 00000000 (UND) ... not in the same module
```

- R_386_GLOB_DAT relocs in .rel.dyn

```
[readelf -r]
00001ff8 00000506 R_386_GLOB_DAT 00000000 cPub
```

- hexadumps of .got section

```
[objdump -s -j .got]
1fd4 d41e0000 00000000 00000000 76040000 .....v...
1fe4 86040000 00000000 00000000 00000000 .....
1ff4 dd050000[00000000]00000000 .....

----> 1ff8 00000000
```

Locating R_386_COPY relocs in .bss section (-fno-pic)

- .bss section address

```
[readelf -S]  
.bss = 00002008
```

- symbol value

```
[readelf -s]  
cPub = 00002008
```

- R_386_COPY relocs in .rel.dyn

```
[readelf -r]  
00002008 0000b05 R_386_COPY          00002008  cPub ... copy to address  
0000061e 0000b01 R_386_32             00002008  cPub ... copy to address
```

- hexadumps of .bss section

```
[objdump -s -j .bss]  
objdump: section '.bss' mentioned in a -j option,  
but not found in any input file
```

TOC: Relocations in an executable file

- (a) calling `fPub` in `.text` section of `run_dynamic`
- (b) referencing `cPub` in `.text` section of `run_dynamic`
- (c) hexadump `.got` section of `librel.so`
- (d) hexadump `.plt` section of `librel.so`
- (e) hexadump `.plt.got` section of `librel.so`
- (f) disassemble `.plt` section of `librel.so`
- (g) disassemble `.plt.got` section of `librel.so`
- Examining `.got` and `.plt` section (-fPIC)

```
int main() {  
    return fPub(123)    // 1  global function symbol reference  
        + cPub;        // 2  global data symbol reference  
}
```

(a) calling fPub in .text section of `run-dynamic`

- `run-dynamic` with `-fno-pic`

```
613: e8 fc ff ff ff          call    614 <main+0x17> ; call func at 614
614: R_386_PC32 fPub
      ; 614 = 5fd + 17 ; fPub func ref location
      ; -4 = ffffffff; offset
      ; 000005fd <main>: ...
```

- `run_dynamic` with default (fPub : `PLT`)

```
5fc: e8 7f fe ff ff          call    480 <fPub@plt> ; call func at 480
5fd: R_386_PLT32          fPub
      ; 5fd = fPub func ref location
      ; -181 = fffffe7f offset (5fd+4-181=480)
      ; 00000480 <fPub@plt>:
      ; 000005dd <main>: ...
```

- `run_dynamic` with `-fPIC` (fPub : `PLT`)

```
5fc: e8 7f fe ff ff          call    480 <fPub@plt> ; call func at 480
5fd: R_386_PLT32          fPub
      ; 5fd = fPub func ref location
      ; -181 = fffffe7f offset (5fd+4-181=480)
      ; 00000480 <fPub@plt>:
      ; 000005dd <main>: ...
```

(b) referencing cPub in .text section of `run-dynamic`

- `run-dynamic` with `-fno-pic`

```
61d: a1 00 00 00 00      mov     0x0,%eax
61e: R_386_32    cPub
; 61e = cPub symbol ref location
; 0 = offset (no pc adjust)
```

- `run_dynamic` with default (cPub : `GOT`)

```
606: 8b 83 24 00 00 00    mov     0x24(%ebx),%eax
608: R_386_GOT32X    cPub ... -Wl,-q
; 608 = cPub symbol ref location
; 24 = offset (1fd4+24 = 1ff8)
; 00001fd4 <_GLOBAL_OFFSET_TABLE_>: ...
; 00001fd4 <.got>: ...
```

- `run_dynamic` with `-fPIC` (cPub : `GOT`)

```
606: 8b 83 24 00 00 00    mov     0x24(%ebx),%eax
608: R_386_GOT32X    cPub ... -Wl,-q
; 608 = cPub symbol ref location
; 24 = offset (1fd4 +24 = 1ff8)
; 00001fd4 <_GLOBAL_OFFSET_TABLE_>: ...
; 00001fd4 <.got>: ...
```

gcc -Wl,-q is used

(c) Hexadump .got section in `run_dynamic` (-fPIC)

```
objdump -s -j .got run-fPIC
```

```
run-fPIC:      file format elf32-i386
```

```
Contents of section .got:
```

```
1fd4 d41e0000 00000000 00000000 76040000 .....v...
1fe4 86040000 00000000 00000000 00000000 .....
1ff4 dd050000 00000000 00000000 .....

```

```
00001ed4 <_DYNAMIC>:
```

```
00000470 <__libc_start_main@plt>:
```

```
476: 68 00 00 00 00          push   $0x0
```

```
00000480 <fPub@plt>:
```

```
486: 68 08 00 00 00          push   $0x8
```

```
000005dd <main>:
```

<https://stackoverflow.com/questions/1685483/how-can-i-examine-contents-of-a-data->

(d) Hexadump .plt section in `run_dynamic` (-fPIC)

```
objdump -s -j .plt run-fPIC
```

```
run-fPIC:      file format elf32-i386
```

```
Contents of section .plt:
```

```
0460 ffb30400 0000ffa3 08000000 00000000 .....  
0470 ffa30c00 00006800 000000e9 e0ffffff .....h.....  
0480 ffa31000 00006808 000000e9 d0ffffff .....h.....
```

<https://stackoverflow.com/questions/1685483/how-can-i-examine-contents-of-a-data->

(e) Hexadump .plt.got section in `run_dynamic` (-fPIC)

```
objdump -s -j .plt.got run-fPIC
```

```
run-fPIC:      file format elf32-i386
```

```
Contents of section .plt.got:
```

```
0490 ffa31800 00006690 ffa31c00 00006690  ....f.....f.
```

<https://stackoverflow.com/questions/1685483/how-can-i-examine-contents-of-a-data->

(f) Disassembly of .plt section in `run_dynamic` (-fPIC)

```
objdump -dr run-fPIC
```

```
00000460 <.plt>:
```

```
460:  ff b3 04 00 00 00      pushl  0x4(%ebx)
466:  ff a3 08 00 00 00      jmp     *0x8(%ebx)
46c:  00 00                  add     %al, (%eax)
    ...
```

```
00000470 <__libc_start_main@plt>:
```

```
470:  ff a3 0c 00 00 00      jmp     *0xc(%ebx)
476:  68 00 00 00 00        push   $0x0
47b:  e9 e0 ff ff ff        jmp     460 <.plt>
```

```
00000480 <fPub@plt>:
```

```
480:  ff a3 10 00 00 00      jmp     *0x10(%ebx)
486:  68 08 00 00 00        push   $0x8
48b:  e9 d0 ff ff ff        jmp     460 <.plt>
```

<https://stackoverflow.com/questions/1685483/how-can-i-examine-contents-of-a-data->

(g) Disassembly of .plt.got section in `run_dynamic` (-fPIC)

```
objdump -dr run-fPIC
```

```
00000490 <__cxa_finalize@plt>:
```

```
490: ff a3 18 00 00 00    jmp     *0x18(%ebx)
```

```
496: 66 90              xchg   %ax,%ax
```

```
00000498 <__gmon_start__@plt>:
```

```
498: ff a3 1c 00 00 00    jmp     *0x1c(%ebx)
```

```
49e: 66 90              xchg   %ax,%ax
```

<https://stackoverflow.com/questions/1685483/how-can-i-examine-contents-of-a-data->

Examining .got and .plt section (-fPIC)

- hexadumps of .got section

```
00001ed4 ... .dynamic          ... at 1fd4
00000476 ... __libc_start_main    ... at 1fe0
00000486 ... fPub                  ... at 1fe4
000005dd ... main                  ... at 1ff4
```

- .plt section disassembly

```
00000470 <__libc_start_main@plt>:
470:  ff a3 0c 00 00 00          jmp     *0xc(%ebx)
476:  68 00 00 00 00            push   $0x0
47b:  e9 e0 ff ff ff           jmp     460 <.plt>

00000480 <fPub@plt>:
480:  ff a3 10 00 00 00          jmp     *0x10(%ebx)
486:  68 08 00 00 00            push   $0x8
48b:  e9 d0 ff ff ff           jmp     460 <.plt>
```