① = ~~foobar.c : 23 : c~~. ⑧ = ~~foobar.c. : 22 : d~~ ⑨ :

1-1 :

① : barbaz.c : 3 : x ② : foobar.c : 12 : b ③ : foobar.c : 13 : c.
④ : barbaz.c : 22 : bar ⑤ : barbaz.c : 3 : x ⑥ : foobar.c : 7 : b.
⑦ : foobar.c : 23 : c ⑧ : foobar.c : 22 : d ⑨ : barbaz.c : 3 : x
⑩ : barbaz.c : 12 : b ⑪ : barbaz.c : 13 : c ⑫ : barbaz.c : 4 : y.
⑬ : barbaz.c : ~~22~~ 9 : bar ⑭ : barbaz.c : 3 : x ⑮ : barbaz.c : 12 : b.
⑯ : barbaz.c : 24 : c ⑰ : foobar.c : 5 : z.

1-2 :

A : gcc -static -o prog2 p.o ./libx.a

B : gcc -no-static -o -prog2 p.o ./libx.a ./liby.a

C : gcc -static -o -prog2 p.o ./libx.a ./liby.a | gcc -stat
-o -prog2 ./liby.a ./libx.a p.o

D : gcc -static -o -prog2 ./liby.a ./libx.a ./libz.a |
gcc -static -o -prog2 p.o ./libx.a ./liby.a ./libz.a

1-3 :

1-3.1 : ① : gcc -o prog2 dynamic_link.c ./libvector.so
~~gcc -rdynamic -o2 -o p2 main.c~~ -ldl

② : void (*subvec)(int *, int *, int *, int);
void * handle = dlopen("./libvector.so", RTLD_LAZY);
subvec = dlsym(handle, "subvec");

1-3-2 : 因为 main.c 以及 subvec.c 中定义的 dcint、x、y、z 等 global variable 的地址需要通过 GOT 才可在 run-time 中获取.

1-3-3 : Before : ~~0x4~~ 0x404566

After : 0x400128

扫描全能王 创建

2-2-1:

①: 2 ②: 1 ③: 1

④: P(a); P(b) ⑤ V(b)

⑥: P(c); P(b) ⑦ V(b).

2-2-2: 1. pthread_create(&tid[i], NULL, print_thread, &ptr).

2: 没有 race, 因为所有 thread 将共用一个 ptr 指针