

```

1  #ifdef RUNTIME
2  #define _GNU_SOURCE
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <dlfcn.h>
6
7  /* malloc wrapper function */
8  void *malloc(size_t size)
9  {
10     void *(*mallocp)(size_t size);
11     char *error;
12
13     mallocp = dlsym(RTLD_NEXT, "malloc"); /* Get address of libc malloc */
14     if ((error = dlerror()) != NULL) {
15         fputs(error, stderr);
16         exit(1);
17     }
18     char *ptr = mallocp(size); /* Call libc malloc */
19     printf("malloc(%d) = %p\n", (int)size, ptr);
20     return ptr;
21 }
22
23 /* free wrapper function */
24 void free(void *ptr)
25 {
26     void (*freep)(void *) = NULL;
27     char *error;
28
29     if (!ptr)
30         return;
31
32     freep = dlsym(RTLD_NEXT, "free"); /* Get address of libc free */
33     if ((error = dlerror()) != NULL) {
34         fputs(error, stderr);
35         exit(1);
36     }
37     freep(ptr); /* Call libc free */
38     printf("free(%p)\n", ptr);
39 }
40 #endif

```

code/link/interpose/mymalloc.c

图 7-22 用 LD_PRELOAD 进行运行时打桩

下面是如何从 bash shell 中运行这个程序^①：

```

linux> LD_PRELOAD="./mymalloc.so" ./intr
malloc(32) = 0x1bf7010
free(0x1bf7010)

```

① 如果你不知道运行的 shell 是哪一种，在命令行上输入 `printenv SHELL`。