dlopen和dlsym是用於打開動態鏈接庫中的函數，將動態鏈接庫中的函數或類導入到本程序中：

dlopen函數：

功能：打開一個動態鏈接庫

dlsym函數：

函數原型是

void\* dlsym(void\* handle,const char\* symbol)

該函數在<dlfcn.h>文件中。

handle是由dlopen打開動態鏈接庫後返回的指針，symbol就是要求獲取的函數的名稱，函數返回值是void\*,指向函數的地址，供調用使用。

先寫一個so 來讀:

root@bejo: cat libso.c

#include <stdio.h>  
int bejo\_lib(char \*name, int i)  
{  
    printf("I am %s, do i=%d\n", name, i);  
    return 0;  
}

包成 dynamic using

#gcc -c libso.c

#gcc -shared -o libso.so libso.o

寫一個使用者~\*

root@bejo: cat use.c

#include <stdio.h>  
#include <stdlib.h>  
#include <dlfcn.h>

int main(int \*argc, char \*\*argv)  
{  
    void \*fHandle;  
    void (\*func)();

//RTLD\_LAZY 暫緩決定，等有需要時再解出符號

　//RTLD\_NOW 立即決定，返回前解除所有未決定的符號  
    fHandle = dlopen("./libso.so",**RTLD\_LAZY**);

    if (!fHandle) {  
        fprintf (stderr, "%s\n", dlerror());  
        exit(1);  
    }  
    dlerror();  
  
    func = (void(\*)())dlsym(fHandle,"bejo\_lib");  
  
    if (func) {  
        func("BEJO", 999);  
    }  
  
    dlclose(fHandle);  
    return 0;  
}

包一包，記得加上 -ldl   ---> libdl.so

#gcc use.c -ldl -o use

# ./use  
I am BEJO, do i=999