NOMBRE

dlsym: obtiene la dirección de un símbolo de un objeto dlopen()

SINOPSIS

```
#include <dlfcn.h>

void *dlsym(void *handle, const char *name);
```

DESCRIPCIÓN

dlsym() permite que un proceso obtenga la dirección de un símbolo definido dentro de un objeto accesible a través de una llamada *dlopen()* . *handle* es el valor devuelto por una llamada a *dlopen()* (y que no se ha liberado desde entonces a través de una llamada a *dlclose()*), *name* es el nombre del símbolo como una cadena de caracteres.

dlsym() buscará el símbolo nombrado en todos los objetos cargados automáticamente como resultado de cargar el objeto al que hace referencia *handle* (ver *dlopen()*). *El orden de carga* se usa en las operaciones *dlsym()* sobre el objeto de símbolo global. El algoritmo de resolución de símbolos utilizado será el orden de *dependencia* como se describe en *dlopen()* .

VALOR DEVUELTO

Si *handle* no hace referencia a un objeto válido abierto por *dlopen()* , o si el símbolo nombrado no se puede encontrar dentro de ninguno de los objetos asociados con *handle* , *dlsym()* devolverá NULL. La información de diagnóstico más detallada estará disponible a través *de dlerror()* .

ERRORES

No se definen errores.

EJEMPLOS

El siguiente ejemplo muestra cómo se pueden usar *dlopen()* y *dlsym()* para acceder a funciones oa objetos de datos. Para simplificar, se ha omitido la comprobación de errores.

```
void    *handle;
int     *iptr, (*fptr)(int);

/* open the needed object */
handle = dlopen("/usr/home/me/libfoo.so.1", RTLD_LAZY);

/* find the address of function and data objects */
fptr = (int (*)(int))dlsym(handle, "my_function");
iptr = (int *)dlsym(handle, "my_object");

/* invoke function, passing value of integer as a parameter */
(*fptr)(*iptr);
```

USO DE LA APLICACIÓN

Los valores de propósito especial para el *mango* están reservados para uso futuro. Estos valores y sus significados son:

RTLD_NEXT
: Specifies the next object after this one that defines *name*. *This one* refers to the object containing the invocation of *dlsym()*. The *next* object is the one found upon the application of a *load* order symbol resolution algorithm (see *dlopen()*). The next object is either one of global scope (because it was introduced as part of the original process image or because it was added with a *dlopen()* operation including the RTLD_GLOBAL flag), or is an object that was included in the same *dlopen()* operation that loaded this one. The RTLD_NEXT flag is useful to navigate an intentionally created hierarchy of multiply defined symbols created through *interposition*. For example, if a program wished to create an implementation of *malloc()* that embedded some statistics gathering about memory allocations, such an implementation could use the real *malloc()* definition to perform the memory allocation - and itself only embed the necessary logic to implement the statistics gathering function.

FUTURE DIRECTIONS

None.

SEE ALSO

dlclose(), *dlerror()*, *dlopen()*.