

```

1: /* Copyright (c) 1993 UNIX System Laboratories, Inc. */
2: /* (a wholly-owned subsidiary of Novell, Inc.). */
3: /* All Rights Reserved. */
4:
5: /* THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF UNIX SYSTEM */
6: /* LABORATORIES, INC. (A WHOLLY-OWNED SUBSIDIARY OF NOVELL, INC.). */
7: /* The copyright notice above does not evidence any actual or */
8: /* intended publication of such source code. */
9:
10: #ident "@(#)sgs-head:common/head/dlfcn.h 1.6"
11:
12: #ifndef _DLFCN_H
13: #define _DLFCN_H
14:
15: /* declarations used for dynamic linking support routines */
16:
17: #ifdef __STDC__
18: extern void *dlopen(const char *, int);
19: extern void *dlsym(void *, const char *);
20: extern int dlclose(void *);
21: extern char *dlerror(void);
22: #else
23: extern void *dlopen();
24: extern void *dlsym();
25: extern int dlclose();
26: extern char *dlerror();
27: #endif
28:
29: /* valid values for mode argument to dlopen */
30:
31: #define RTLD_LAZY 1 /* lazy function call binding */
32: #define RTLD_NOW 2 /* immediate function call binding */
33: #define RTLD_GLOBAL 4 /* all symbols available for binding */
34:
35: #endif /* _DLFCN_H */

```

```

1: /* User functions for run-time dynamic loading.
2: Copyright (C) 1995-1999, 2000, 2001 Free Software Foundation, Inc.
3: This file is part of the GNU C Library.
...
40:
41: __BEGIN_DECLS
42:
43: /* Open the shared object FILE and map it in; return a handle that can be
44: passed to `dlsym' to get symbol values from it. */
45: extern void *dlopen (__const char *__file, int __mode) __THROW;
46:
47: /* Unmap and close a shared object opened by `dlopen'.
48: The handle cannot be used again after calling `dlclose'. */
49: extern int dlclose (void *__handle) __THROW;
50:
51: /* Find the run-time address in the shared object HANDLE refers to
52: of the symbol called NAME. */
53: extern void *dlsym (void *__restrict __handle,
54: __const char *__restrict __name) __THROW;
55:
56: #ifdef __USE_GNU
57: /* Find the run-time address in the shared object HANDLE refers to
58: of the symbol called NAME with VERSION. */
59: extern void *dlsym (void *__restrict __handle,
60: __const char *__restrict __name,
61: __const char *__restrict __version) __THROW;
62: #endif
63:
64: /* When any of the above functions fails, call this function
65: to return a string describing the error. Each call resets
66: the error string so that a following call returns null. */
67: extern char *dlerror (void) __THROW;
68:
69:
70: #ifdef __USE_GNU
71: /* Structure containing information about object searched using
72: `dladdr'. */
73: typedef struct
74: {
75: __const char *dli_fname; /* File name of defining object. */
76: void *dli_fbase; /* Load address of that object. */
77: __const char *dli_sname; /* Name of nearest symbol. */
78: void *dli_saddr; /* Exact value of nearest symbol. */
79: } Dl_info;
80:
81: /* Fill in *INFO with the following information about ADDRESS.
82: Returns 0 iff no shared object's segments contain that address. */
83: extern int dladdr (__const void *__address, Dl_info *__info) __THROW;
84: #endif
85:
86: __END_DECLS
87:
88: #endif /* dlfcn.h */

```