

```

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5: /* THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF UNIX SYSTEM */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: /* System dependent definitions for run-time dynamic loading.
2: Copyright (C) 1996,1997,1998,1999,2000,2001 Free Software Foundation, Inc.
3: This file is part of the GNU C Library.
4:
5: The GNU C Library is free software; you can redistribute it and/or
6: modify it under the terms of the GNU Lesser General Public
7: License as published by the Free Software Foundation; either
8: version 2.1 of the License, or (at your option) any later version.
9:
10: The GNU C Library is distributed in the hope that it will be useful,
11: but WITHOUT ANY WARRANTY; without even the implied warranty of
12: MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
13: Lesser General Public License for more details.
14:
15: You should have received a copy of the GNU Lesser General Public
16: License along with the GNU C Library; if not, write to the Free
17: Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
18: 02111-1307 USA. */
19:
20: #ifndef _DLFCN_H
21: # error "Never use <bits/dlfcn.h> directly; include <dlfcn.h> instead."
22: #endif
23:
24: /* The MODE argument to `dlopen' contains one of the following: */
25: #define RTLD_LAZY 0x00001 /* Lazy function call binding. */
26: #define RTLD_NOW 0x00002 /* Immediate function call binding. */
27: #define RTLD_BINDING_MASK 0x3 /* Mask of binding time value. */
28: #define RTLD_NOLOAD 0x00004 /* Do not load the object. */
29:
30: /* If the following bit is set in the MODE argument to `dlopen',
31: the symbols of the loaded object and its dependencies are made
32: visible as if the object were linked directly into the program. */
33: #define RTLD_GLOBAL 0x00100
34:
35: /* Unix98 demands the following flag which is the inverse to RTLD_GLOBAL.
36: The implementation does this by default and so we can define the
37: value to zero. */
38: #define RTLD_LOCAL 0
39:
40: /* Do not delete object when closed. */
41: #define RTLD_NODELETE 0x01000
42:
43: #ifdef __USE_GNU
44: /* To support profiling of shared objects it is a good idea to call
45: the function found using `dlsym' using the following macro since
46: these calls do not use the PLT. But this would mean the dynamic
47: loader has no chance to find out when the function is called. The
48: macro applies the necessary magic so that profiling is possible.
49: Rewrite
50: foo = (*fctp) (arg1, arg2);
51: into
52: foo = DL_CALL_FCT (fctp, (arg1, arg2));
53: */
54: # define DL_CALL_FCT(fctp, args) \
55: (_dl_mcount_wrapper_check ((void *) (fctp)), (*fctp) args)
56:
57: __BEGIN_DECLS
58:
59: /* This function calls the profiling functions. */
60: extern void _dl_mcount_wrapper_check (void *__selfpc) __THROW;
61:
62: __END_DECLS
63:
64: #endif

```