

```

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5: /* THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF UNIX SYSTEM */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/ucontext.h 1.2"
11:
12: #ifndef _UCONTEXT_H
13: #define _UCONTEXT_H
14:
15: #include <sys/ucontext.h>
16:
17: #if defined(__STDC__)
18:
19: extern int getcontext(ucontext_t *);
20: extern int setcontext(ucontext_t *);
21: extern int swapcontext(ucontext_t *, ucontext_t *);
22: extern void makecontext(ucontext_t *, void (*)(void), int, ...);
23:
24: #else
25:
26: extern int getcontext();
27: extern int setcontext();
28: extern int swapcontext();
29: extern void makecontext();
30:
31: #endif
32:
33: #endif /* _UCONTEXT_H */

```

```

1: /* Copyright (C) 1997, 1998, 1999, 2000 Free Software Foundation, Inc.
2: This file is part of the GNU C Library.
3:
4: The GNU C Library is free software; you can redistribute it and/or
5: modify it under the terms of the GNU Lesser General Public
6: License as published by the Free Software Foundation; either
7: version 2.1 of the License, or (at your option) any later version.
8:
9: The GNU C Library is distributed in the hope that it will be useful,
10: but WITHOUT ANY WARRANTY; without even the implied warranty of
11: MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
12: Lesser General Public License for more details.
13:
14: You should have received a copy of the GNU Lesser General Public
15: License along with the GNU C Library; if not, write to the Free
16: Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
17: 02111-1307 USA. */
18:
19: /* System V ABI compliant user-level context switching support. */
20:
21: #ifndef _UCONTEXT_H
22: #define _UCONTEXT_H 1
23:
24: #include <features.h>
25:
26: /* Get machine dependent definition of data structures. */
27: #include <sys/ucontext.h>
28:
29: __BEGIN_DECLS
30:
31: /* Get user context and store it in variable pointed to by UCP. */
32: extern int getcontext (ucontext_t *__ucp) __THROW;
33:
34: /* Set user context from information of variable pointed to by UCP. */
35: extern int setcontext (__const ucontext_t *__ucp) __THROW;
36:
37: /* Save current context in context variable pointed to by OUCP and set
38: context from variable pointed to by UCP. */
39: extern int swapcontext (ucontext_t *__restrict __oucp,
40:                        __const ucontext_t *__restrict __ucp) __THROW;
41:
42: /* Manipulate user context UCP to continue with calling functions FUNC
43: and the ARGV-1 parameters following ARGV when the context is used
44: the next time in `setcontext' or `swapcontext'.
45:
46: We cannot say anything about the parameters FUNC takes; `void'
47: is as good as any other choice. */
48: extern void makecontext (ucontext_t *__ucp, void (*__func) (void),
49:                          int __argc, ...) __THROW;
50:
51: __END_DECLS
52:
53: #endif /* ucontext.h */

```