/\*

\* Module: sched.h

\*

\* Purpose:

\* Provides an implementation of POSIX realtime extensions

\* as defined in

\*

\* POSIX 1003.1b-1993 (POSIX.1b)

\*

\* --------------------------------------------------------------------------

\*

\* Pthreads-win32 - POSIX Threads Library for Win32

\* Copyright(C) 1998 John E. Bossom

\* Copyright(C) 1999,2005 Pthreads-win32 contributors

\*

\* Contact Email: rpj@callisto.canberra.edu.au

\*

\* The current list of contributors is contained

\* in the file CONTRIBUTORS included with the source

\* code distribution. The list can also be seen at the

\* following World Wide Web location:

\* http://sources.redhat.com/pthreads-win32/contributors.html

\*

\* This library is free software; you can redistribute it and/or

\* modify it under the terms of the GNU Lesser General Public

\* License as published by the Free Software Foundation; either

\* version 2 of the License, or (at your option) any later version.

\*

\* This library is distributed in the hope that it will be useful,

\* but WITHOUT ANY WARRANTY; without even the implied warranty of

\* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

\* Lesser General Public License for more details.

\*

\* You should have received a copy of the GNU Lesser General Public

\* License along with this library in the file COPYING.LIB;

\* if not, write to the Free Software Foundation, Inc.,

\* 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

\*/

#if !defined(\_SCHED\_H)

#define \_SCHED\_H

#undef PTW32\_SCHED\_LEVEL

#if defined(\_POSIX\_SOURCE)

#define PTW32\_SCHED\_LEVEL 0

/\* Early POSIX \*/

#endif

#if defined(\_POSIX\_C\_SOURCE) && \_POSIX\_C\_SOURCE >= 199309

#undef PTW32\_SCHED\_LEVEL

#define PTW32\_SCHED\_LEVEL 1

/\* Include 1b, 1c and 1d \*/

#endif

#if defined(INCLUDE\_NP)

#undef PTW32\_SCHED\_LEVEL

#define PTW32\_SCHED\_LEVEL 2

/\* Include Non-Portable extensions \*/

#endif

#define PTW32\_SCHED\_LEVEL\_MAX 3

#if ( defined(\_POSIX\_C\_SOURCE) && \_POSIX\_C\_SOURCE >= 200112 ) || !defined(PTW32\_SCHED\_LEVEL)

#define PTW32\_SCHED\_LEVEL PTW32\_SCHED\_LEVEL\_MAX

/\* Include everything \*/

#endif

#if defined(\_\_GNUC\_\_) && !defined(\_\_declspec)

# error Please upgrade your GNU compiler to one that supports \_\_declspec.

#endif

/\*

\* When building the library, you should define PTW32\_BUILD so that

\* the variables/functions are exported correctly. When using the library,

\* do NOT define PTW32\_BUILD, and then the variables/functions will

\* be imported correctly.

\*/

#if !defined(PTW32\_STATIC\_LIB)

# if defined(PTW32\_BUILD)

# define PTW32\_DLLPORT \_\_declspec (dllexport)

# else

# define PTW32\_DLLPORT \_\_declspec (dllimport)

# endif

#else

# define PTW32\_DLLPORT

#endif

/\*

\* This is a duplicate of what is in the autoconf config.h,

\* which is only used when building the pthread-win32 libraries.

\*/

#if !defined(PTW32\_CONFIG\_H)

# if defined(WINCE)

# define NEED\_ERRNO

# define NEED\_SEM

# endif

# if defined(\_\_MINGW64\_\_)

# define HAVE\_STRUCT\_TIMESPEC

# define HAVE\_MODE\_T

# elif defined(\_UWIN) || defined(\_\_MINGW32\_\_)

# define HAVE\_MODE\_T

# endif

#endif

/\*

\*

\*/

#if PTW32\_SCHED\_LEVEL >= PTW32\_SCHED\_LEVEL\_MAX

#if defined(NEED\_ERRNO)

#include "need\_errno.h"

#else

#include <errno.h>

#endif

#endif /\* PTW32\_SCHED\_LEVEL >= PTW32\_SCHED\_LEVEL\_MAX \*/

#if (defined(\_\_MINGW64\_\_) || defined(\_\_MINGW32\_\_)) || defined(\_UWIN)

# if PTW32\_SCHED\_LEVEL >= PTW32\_SCHED\_LEVEL\_MAX

/\* For pid\_t \*/

# include <sys/types.h>

/\* Required by Unix 98 \*/

# include <time.h>

# else

typedef int pid\_t;

# endif

#else

typedef int pid\_t;

#endif

/\* Thread scheduling policies \*/

enum {

SCHED\_OTHER = 0,

SCHED\_FIFO,

SCHED\_RR,

SCHED\_MIN = SCHED\_OTHER,

SCHED\_MAX = SCHED\_RR

};

struct sched\_param {

int sched\_priority;

};

#if defined(\_\_cplusplus)

extern "C"

{

#endif /\* \_\_cplusplus \*/

PTW32\_DLLPORT int \_\_cdecl sched\_yield (void);

PTW32\_DLLPORT int \_\_cdecl sched\_get\_priority\_min (int policy);

PTW32\_DLLPORT int \_\_cdecl sched\_get\_priority\_max (int policy);

PTW32\_DLLPORT int \_\_cdecl sched\_setscheduler (pid\_t pid, int policy);

PTW32\_DLLPORT int \_\_cdecl sched\_getscheduler (pid\_t pid);

/\*

\* Note that this macro returns ENOTSUP rather than

\* ENOSYS as might be expected. However, returning ENOSYS

\* should mean that sched\_get\_priority\_{min,max} are

\* not implemented as well as sched\_rr\_get\_interval.

\* This is not the case, since we just don't support

\* round-robin scheduling. Therefore I have chosen to

\* return the same value as sched\_setscheduler when

\* SCHED\_RR is passed to it.

\*/

#define sched\_rr\_get\_interval(\_pid, \_interval) \

( errno = ENOTSUP, (int) -1 )

#if defined(\_\_cplusplus)

} /\* End of extern "C" \*/

#endif /\* \_\_cplusplus \*/

#undef PTW32\_SCHED\_LEVEL

#undef PTW32\_SCHED\_LEVEL\_MAX

#endif /\* !\_SCHED\_H \*/