NAME

INTRO − Introduction to the Shore Threads Package

SYNOPSIS

The Shore Thread Package (sthread) provides C++ style light-weight processes. It provides the following facilities (further described by documents referenced in the See Also section):

Threads

Mutexes

Conditional variables

Event semaphores

Asynchronous disk I/O

Asynchronous file I/O handlers

DESCRIPTION

The thread mechanism allows several threads of control to share the same address space. Each thread is represented by an instance of **sthread_t** class. Once created, a thread is an independent entity with its own stack.

In a C++ program, the sthread initialization code is built into the library such it will execute before the **main()** function begins. The initialization code is responsible for spawning a **main_thread,** such that, when the initialization function returns, it returns in the context of the **main_thread.** This ensures that the program executes in a threaded environment from the beginning.

ENVIRONMENT

Sthread currently runs on the following processors/platforms:

DecStation 3100 and 5000 series (MIPS, Ultrix)

Intel Paragon (i860, OSF/1)

HP 700 series (HP-PA, HP-UX 8.0 and 9.0)

Sparcstation (SunOS 4.3.1, Solaris)

x86 (Linux, Solaris)

Silicon Graphics (Irix)

ERRORS

See **errors(sthread)**

VERSION

This manual page applies to Version 2.0 of the Shore Storage Manager.

SPONSORSHIP

The Shore project is sponsored by the Advanced Research Project Agency, ARPA order number 018 (formerly 8230), monitored by the U.S. Army Research Laboratory under contract DAAB07-91-C-Q518. Further funding for this work was provided by DARPA through Rome Research Laboratory Contract No. F30602-97-2-0247.

COPYRIGHT

SEE ALSO

sthread_t(sthread)

smutex_t(sthread)

scond_t(sthread)

sevsem_t(sthread)

file_handlers(sthread)