POSIX Threads

From Wikipedia, the free encyclopedia

POSIX Threads, or **Pthreads**, is a POSIX standard for threads. The standard defines an API for creating and manipulating threads. Pthreads are most commonly used on Unix-like POSIX systems such as FreeBSD, NetBSD, GNU/Linux, Mac OS X and Solaris, but Microsoft Windows implementations also exist. For example, the pthreads-w32 is available and supports a subset of the Pthread API for the Windows 32-bit platform.[1]

Contents

Contents

Pthreads defines a set of C programming language types, functions and constants. It is implemented with a `pthread.h (http://opengroup.org/onlinepubs/007908799/xsh/pthread.h.html)` header and a thread library. Programmers can use Pthreads to create, manipulate and manage threads, as well as synchronize between threads using mutexes and signals.

Example

An example of using Pthreads in C:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <pthread.h>
#include <unistd.h>

static void *thread_func(void *vptr_args)
{
    int i;

    for (i = 0; i < 20; i++)
    {
        fputs("  b\n", stderr);
        sleep(1);  // suspend thread for one second.
    }

    return NULL;
}

int main(void)
{
    int i;
    pthread_t thread;

    if (pthread_create(&thread, NULL, thread_func, NULL) != 0)
    {
        return EXIT_FAILURE;
    }

    for (i = 0; i < 20; i++)
    {
        fputs("a\n", stdout);
        sleep(1);
    }

    if (pthread_join(thread, NULL) != 0)
    {
        return EXIT_FAILURE;
    }

    return EXIT_SUCCESS;
}
```

This program creates a new thread that prints lines containing 'b', while the main thread prints lines containing 'a'. The output is interleaved between 'a' and 'b' as a result of execution switching between the two threads, or simultaneous execution on a multicore system. In any case, the pattern of 'a's and 'b's does not strictly alternate between each letter, and can vary even between different runs on the same machine.

An example of using Pthreads in C++:

```
#include <stdio.h>
#include <time.h>
#include <unistd.h>
#include <pthread.h>
#include <stdlib.h>

class Thread{
        pthread_t thread;
        static void *thread_func(void *d){((Thread *)d)->run(); return NULL;}
public:
        Thread(){}
        virtual ~Thread(){}

        virtual void run(){}
        int start(){return pthread_create(&thread, NULL, Thread::thread_func, (void*)this);}
        int wait(){return pthread_join(thread, NULL);}
};

int main(void){
        class Thread_a:public Thread{
        public:
                void run(){for(int i=0;i<20;i++){sleep(1);printf("a  \n");}}
        };

        class Thread_b:public Thread{
        public:
                void run(){for(int i=0;i<20;i++){sleep(1);printf("  b\n");}}
        };

        Thread *a=new Thread_a();
        Thread *b=new Thread_b();

        if (a->start() != 0) {
                return EXIT_FAILURE;
        }

        if (b->start() != 0) {
                return EXIT_FAILURE;
        }

        if (a->wait() != 0) {
                return EXIT_FAILURE;
        }

        if (b->wait() != 0) {
                return EXIT_FAILURE;
        }

        delete a;
        delete b;

        return EXIT_SUCCESS;
}
```

This program creates two new threads, one which prints lines containing 'a', and one which prints lines containing 'b', while the main thread exits without printing anything. Like in the example above, the output is interleaved between 'a' and 'b' as a result of execution switching between the two threads, or simultaneous execution on a multicore system.

See also

- OpenMP
- Native POSIX Thread Library (NPTL)
- Spurious wakeup
- Thread-local storage
- GNU Portable Threads
- FSU Pthreads

References

1. ^ Pthread Win-32 (http://sources.redhat.com/pthreads-win32/)

Further reading

- David R. Butenhof: *Programming with POSIX Threads*, Addison-Wesley, ISBN 0-201-63392-2
- Bradford Nichols, Dick Buttlar, Jacqueline Proulx Farell: *Pthreads Programming*, O'Reilly & Associates, ISBN 1-56592-115-1
- Charles J. Northrup: *Programming with UNIX Threads*, John Wiley & Sons, ISBN 0-471-13751-0
- Kay A. Robbins and Steven Robbins, *UNIX Systems Programming*, Prentice-Hall, ISBN 0-13-042411-0

External links

- Pthread Win-32 (http://sources.redhat.com/pthreads-win32/) , Basic Programming
- Pthreads Tutorial (https://computing.llnl.gov/tutorials/pthreads/)
- C/C++ Tutorial: using Pthreads (http://yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html)
- Article "POSIX threads explained (http://www-128.ibm.com/developerworks/linux/library/l-posix1.html) " by Daniel Robbins (Gentoo Linux founder)
- Interview "Ten Questions with David Butenhof about Parallel Programming and POSIX Threads (http://www.thinkingparallel.com/2007/04/11/ten-questions-with-david-butenhof-about-parallel-programming-and-posix-threads/) " by Michael Suess
- Open Source POSIX Threads for Win32 (http://sources.redhat.com/pthreads-win32/)

- The Open Group Base Specifications Issue 6, IEEE Std 1003.1 (http://www.opengroup.org/onlinepubs/007904975/basedefs/pthread.h.html)
- GNU Portable threads (http://www.gnu.org/software/pth/)
- Flash Presentation on pThread (http://www.geocities.com/dipak123/software/MultiThreading.html)
- Pthreads Presentation at 2007 OSCON (O'Reilly Open Source Convention) by Adrien Lamothe. An overview of Pthreads with current trends. (http://conferences.oreillynet.com/presentations/os2007/os_lamothe.pdf)

Retrieved from "http://en.wikipedia.org/wiki/POSIX_Threads"
Categories: POSIX standards | Parallel computing | Threads