

Basic Multithreading using POSIX threads

Tushar B. Kute,
<http://tusharkute.com>

Threads

- A Thread is an independent stream of instructions that can be schedule to run as such by the OS.
- Think of a thread as a “procedure” that runs independently from its main program.
- Multi-threaded programs are where several procedures are able to be scheduled to run simultaneously and/or independently by the OS.
- A Thread exists within a process and uses the process resources.

Threads

- Threads only duplicate the essential resources it needs to be independently schedulable.
- A thread will die if the parent process dies.
- A thread is “lightweight” because most of the overhead has already been accomplished through the creation of the process.

The POSIX threads

- POSIX – Portable Operating System Interface for Unix.
- For UNIX systems, implementations of threads that adhere to the IEEE POSIX 1003.1c standard are Pthreads.
- Pthreads are C language programming types defined in the pthread.h header/include file.

The Pthreads

- The primary motivation behind Pthreads is improving program performance.
- Can be created with much less OS overhead.
- Needs fewer system resources to run.
- View comparison of forking processes to using a `pthread_create` subroutine.
Timings reflect 50,000 processes/thread creations.

Find multithreaded programs in Linux

- `ps -aux`

```
sitrc 4148 2.0 8.0 493048 162860 ? S 10:17 0:55 /usr/lib/libreoffice/progr
sitrc 4263 0.0 0.9 101460 19344 ? S 10:18 0:00 /usr/lib/i386-linux-gnu/un
sitrc 4279 0.0 1.1 105812 23516 ? S 10:18 0:00 /usr/bin/unity-scope-loade
sitrc 4280 0.0 0.7 101828 15492 ? S 10:18 0:00 /usr/lib/i386-linux-gnu/un
sitrc 4283 0.0 0.5 76656 11504 ? S 10:18 0:00 /usr/lib/i386-linux-gnu/un
sitrc 4321 0.0 1.5 253052 31876 ? S 10:18 0:01 gedit
root 4363 0.0 0.0 0 0 ? S 10:19 0:00 [kworker/u16:2]
root 4691 0.0 0.0 0 0 ? S 10:50 0:00 [kworker/u16:0]
root 4759 0.0 0.0 0 0 ? S 10:58 0:00 [kworker/u16:1]
sitrc 4761 7.2 7.3 640728 148648 ? S 10:58 0:14 /usr/lib/firefox/firefox
sitrc 4786 0.0 0.2 34808 4756 ? S 10:59 0:00 /usr/lib/libunity-webapps/
sitrc 4944 1.5 1.4 226980 28960 ? SL 11:01 0:00 gnome-terminal
sitrc 4950 0.0 0.0 2428 1760 ? S 11:01 0:00 gnome-pty-helper
sitrc 4951 0.0 0.1 5720 3268 pts/4 Ss 11:01 0:00 bash
sitrc 4998 0.0 0.1 5222 2288 pts/4 R 11:02 0:00 ps -aux
```

Single threaded application

Multithreaded application

Single threaded program

```
#include<stdio.h>
int main()
{
    while(1)
    {
        printf("Hello...\n");
    }
    return 0;
}
```

Check the entry in process list...

Pthread library

- Pthread Library (60+ functions)
 - Thread management: create, exit, detach, join, . . .
 - Thread cancellation
 - Mutex locks: init, destroy, lock, unlock, . . .
 - Condition variables: init, destroy, wait, timed wait,
- Programs must include the file pthread.h
- Programs must be linked with the pthread library (**-lpthread**)

Pthread: naming conventions

- Types: `pthread[_object]_t`
- Functions: `pthread[_object]_action`
- Constants/Macros: `PTHREAD_PURPOSE`

- Examples:
 - `pthread_t`: the type of a thread
 - `pthread_create()`: creates a thread
 - `pthread_mutex_t`: the type of a mutex lock
 - `pthread_mutex_lock()`: lock a mutex
 - `PTHREAD_CREATE_DETACHED`

pthread_create

- **Creates a new thread**

- `int pthread_create (pthread_t *thread, pthread_attr_t *attr, void * (*start_routine) (void *), void *arg);`

- Returns 0 to indicate success, otherwise returns error code...

- `thread`: output argument for the id of the new thread
- `attr`: input argument that specifies the attributes of the thread to be created (NULL = default attributes)
- `start_routine`: function to use as the start of the new thread must have prototype: `void * foo(void*)`
- `arg`: argument to pass to the new thread routine. If the thread routine requires multiple arguments, they must be passed bundled up in an array or a structure

pthread_exit

- **Terminates the calling thread**
- `void pthread_exit(void *retval);`
- The return value is made available to another thread calling a `pthread_join()`
- The return value of the function serves as the argument to the (implicitly called) `pthread_exit()`.

pthread_join

- **Causes the calling thread to wait for another thread to terminate**
- `int pthread_join(pthread_t thread, void **value_ptr);`
 - `thread`: input parameter, id of the thread to wait on
 - `value_ptr`: output parameter, value given to `pthread_exit()` by the terminating thread (which happens to always be a void *)
- Returns 0 to indicate success, error code otherwise
- Multiple simultaneous calls for the same thread are not allowed

Single threaded program

```
int first()
{
    int i;
    for(i=0; ;i++)
    {
        printf("\nFirst: %d",i);
        sleep(1);
    }
}
```

```
int main()
{
    int i;
    first( );
    for(i=0;;i++)
    {
        printf("\nMain: %d",i);

        sleep(1);
    }
    return 0;
}
```

Multithreaded program

```
#include<unistd.h>
#include<stdio.h>
#include<pthread.h>
int first()
{
    int i;
    for(i=0;;i++)
    {
        printf("\nFirst: %d",i);
        sleep(1);
    }
}
```

```
int main()
{
    pthread_t th;
    int i;
    pthread_create(&th, 0, (void
        *) &first, NULL);
    for(i=0;;i++)
    {
        printf("\nMain: %d",i);
        sleep(1);
    }
    pthread_join(th, NULL);
    return 0;
}
```

Output

```
Terminal
sitrc@tushar:~$ gcc single.c -lpthread
sitrc@tushar:~$ ./a.out

Main: 0
First: 0
Main: 1
First: 1
Main: 2
First: 2
Main: 3
First: 3
Main: 4
First: 4
```

Process list



```
root      6459  0.0  0.0      0      0 ?        S   12:11   0:00 [kworker/u16:1]
root      6488  0.0  0.1   3092   2096 ?        S   12:12   0:00 /lib/systemd/sy
sitrc     6502  0.0  0.4  135968  9368 ?        SNl  12:12   0:00 /usr/lib/tracke
sitrc     6551  0.0  0.0   10480    592 pts/4    Sl+  12:13   0:00 ./a.out
sitrc     6558  0.0  0.1    5720   3320 pts/10   Ss   12:13   0:00 bash
sitrc     6595  0.0  0.1    5232   2316 pts/10   R+   12:13   0:00 ps -aux
sitrc@tushar:~$
```

Thank you

This presentation is created using LibreOffice Impress 4.2.7.2, can be used freely as per GNU General Public License

Web Resources

<http://tusharkute.com>

Blogs

<http://digitallocha.blogspot.in>
<http://kyamputar.blogspot.in>

tushar@tusharkute.com