LinuxQuestions.org

Visit Jeremy's Blog.

Programming This forum is for all programming questions.
The question does not have to be directly related to Linux and any language is fair game.

Notices

Welcome to **LinuxQuestions.org**, a friendly and active Linux Community.

You are currently viewing LQ as a guest. By joining our community you will have the ability to post topics, receive our newsletter, use the advanced search, subscribe to threads and access many other special features. Registration is quick, simple and absolutely free. **Join our community** today!

Note that registered members see fewer ads, and ContentLink is completely disabled once you log in.

Are you new to LinuxQuestions.org? Visit the following links:
Site Howto | Site FAQ | Sitemap | Register Now

If you have any problems with the registration process or your account login, please contact us. If you need to reset your password, click here.

Having a problem logging in? Please visit this page to clear all LQ-related cookies.

Post Reply

05-30-2003, 08:50 AM #1

powerplane
LQ Newbie

Registered: Apr 2003
Location: P.R. China
Distribution: FreeBSD 4.8 stable
Posts: 26

Rep:

fork vs thread/pthread

To built a TCP server(the most simplest is a tcp echo server), there two ways to support multi-client connections in general: fork and pthread/thread.

fork is more expansive than thread, I think fork is the old fashion things, and thread should be pop in some new softwares. But when I take quick look("grep -r") on some very pop server's src, I surprisingly found that most of them use fork and only very few of them use thread/pthread.

Fork:
telnetd(freebsd), vsftpd, proftpd, Apache13, Apache2, thttpd, firebird(a bbsd, not the database), PostgreSQL, MySQL-323

pthread:
Apache13, Apache2, MySQL 323

If I want to built a new server, which should I use? fork or thread?
Any suggestions? Or other alternatives ?

06-01-2003, 05:08 AM #2

llama_meme
Member

Registered: Nov 2001
Location: London, England
Distribution: Gentoo, FreeBSD
Posts: 590

Rep:

Well fork is pretty universal; it'll work on any UNIX system. Most UNIXs support POSIX threads by now, but the quality of the implementation varies.

I'd use pthreads, because they're much easier to work with (since you can easily have access to data structures shared between threads, etc.)

Alex

06-01-2003, 05:59 AM #3

Mara
Moderator

Registered: Feb 2002
Location: Grenoble
Distribution: Debian
Posts: 9,395

Rep:

I use pthreads, too. They're 'lighter' and it's easier to share data.
You can also use 'select' (use 'man 2 select' for more info).

06-01-2003, 08:51 AM #4

powerplane
LQ Newbie

Registered: Apr 2003
Location: P.R. China
Distribution: FreeBSD 4.8 stable
Posts: 26

Original Poster
Rep:

I also post a thread on bsdforums.org, I get some other opinons, I think you may be interested.
http://www.bsdforum.org/forums/showt...threadid=10384

Quote:

> *Originally posted by AbEnd*
> **Ever heard of multiplexed (man 2 select poll kqueue) or signal driven (man 2 fcntl) IO? AFAIK multiplexed might be generally faster than threaded.**
>
> **You're wrong to categorize some of those daemons as "uses fork for parallel stuff", some uses fork for other things (not based on the connections).**
>
> **What you should use really depends on how you want it to work, not just performance... I think inetd is good for telnet because it isolates the different sessions (since telnet is important) and FTP cause you can change the creds of the processes.**
>
> **But HTTP always get alot of [new] connections, that's why thttpd uses multiplexed IO, Apache 1 uses preforking, Apache 2 can use threading, etc.**

05-05-2006, 04:07 AM #5

vskgopu
LQ Newbie

Registered: May 2006
Posts: 17

Rep:

Which is best for implementing SMTP servers in unix.. fork or pthread ?

05-06-2006, 02:19 AM #6

primo
Member

Registered: Jun 2005
Posts: 542

Rep:

On Linux it is said that "processes are threads", so there's not much of a difference if these new processes do some precise jobs without tinkering with global data that triggers the Copy-On-Write scheme so they work on their own environment. There is an entire chapter in Stevens' book on many of the approaches that you may use on servers: preemptive forks, threads and so on. With threads, be prepared to use locking around global data. It would be better to start the core of the server itself and networking/protocol stuff and later try the threads stuff.

05-06-2006, 06:28 AM #7

sibtay
Member

Registered: Aug 2004
Location: U.S
Distribution: Ubuntu
Posts: 145

Rep:

Personally i would use fork, because of the following reasons:

1) Fork is more universally accepted than threads.

2) Considering the type of application which you are working on, there wont be much of Interprocess communication (ipc) required. Actually threads really win the race when it comes to inter communication. Since threads share the same memory space hence sharing data between them is really faster as
compared to seperate processes. Where you have to either employ costly approaches like pipes, fifos, shared memory etc.

3) Developement is much easier on a fork based implementation.

4) Fork based implementations are far more maintainable.

5) If the application is in C, then it must be having some global data. In a multi threaded application, all that global data *must* be protected with locks, since it will be shared by all the threads. And locks can prove very costly (refer to the laws of mutual exclusion and critical sections). In contrast in
a multi process implementation each process has its own copy of global data.

Regards,
Sibtay

05-06-2006, 06:32 AM #8

sibtay
Member

Registered: Aug 2004
Location: U.S
Distribution: Ubuntu

(sorry, i forgot to add in my previous post)

I have'nt discussed the advantages of threads. However it should be considered that if properly designed and implemented threads give you

Posts: 145

Rep:

more speed (because there aint any process level context switching in a multi threaded application).

05-07-2006, 10:43 PM #9

vskgopu
LQ Newbie

Registered: May 2006
Posts: 17

Rep:

Thanks friend

05-08-2006, 09:07 AM #10

bigearsbilly
Senior Member

Registered: Mar 2004
Location: england
Distribution: FreeBSD, Puppy
Posts: 3,018

Rep:

the stevens' book says don't use threads.
that's good enough for me.

http://www.radwin.org/michael/blog/2...d_harmful.html

05-09-2006, 02:46 AM #11

primo
Member

Registered: Jun 2005
Posts: 542

Rep:

Beware the "XYZ sucks" stance and carefully weigh the alternatives. Please tell me where does Stevens say "don't use threads".

05-09-2006, 04:02 AM #12

vskgopu
LQ Newbie

Registered: May 2006
Posts: 17

Rep:

I got confused now....shall i use thread to implement a SMTP relay server...?
What is the max number of threads/fork that can be created a process/parent process..?
My friend is using fork to implement the SMTP relay server....I thought of using thread..
What shall i do now ??

05-09-2006, 05:42 AM #13

primo
Member

Registered: Jun 2005
Posts: 542

Rep:

Quote:

> Originally Posted by **vskgopu**
> *What is the max number of threads/fork that can be created a process/parent process..?*

Run "getconf CHILD_MAX" in the shell. It prints the value of the sysconf(_SC_CHILD_MAX) call. It's related to process limits (run "ulimit -Ha" and "ulimit -Sa") that may be increased and/or decreased by normal users with some limitations (see "man getrlimit"). Also, in some Unix systems, the hard

limits may be increased by means of a sysctl() call.

05-09-2006, 10:19 PM #14

chrism01
Guru

Registered: Aug 2004
Location: Brisbane
Distribution: Centos 5.7, Solaris 10
Posts: 11,121

Rep:

It's easier to share global values in threads, but be aware that in 2.6 kernel they are created as lightweight processes within the main thread/process, so you only get 1 pid.
If you don't need to share data, then fork is an option and each process will have it's own pid.
I've written a threaded Perl prog that works fine on 2.4 & 2.6 kernel, but I need some global shared data. Note that in Perl, NO vars are shared by default, so it's easy to avoid cross-thread variable update issues.

01-20-2010, 03:41 PM #15

manish041083
LQ Newbie

Registered: Jun 2007
Posts: 5

Rep:

I find this link which could be helpful in clearing few facts about forking and threading.

http://www.geekride.com/index.php/20...-linux-kernel/

Post Reply

Page 1 of 2 **1** 2 >

Posting Rules

You **may not** post new threads
You **may not** post replies
You **may not** post attachments
You **may not** edit your posts

BB code is **On**
Smilies are **On**
[IMG] code is **Off**
HTML code is **Off**
Trackbacks are **Off**
Pingbacks are **On**
Refbacks are **Off**

Posting Rules

Forum Rules

Similar Threads

Thread	Thread Starter	Forum	Replies	Last Post
what happens if i create a thread and then fork the process?	Thinking	Programming	7	10-31-2005 03:22 PM
Sockets, pthread, pipe, and fork -- a messaging server	ggravarr	Programming	3	06-28-2005 03:39 PM
[thread control suggestion] add a "solved" button that the thread starter can click	atom	LQ Suggestions & Feedback	3	03-24-2005 12:55 PM
Main thread sending notification to child thread	rajesh_b	Programming	1	09-22-2004 10:15 AM
configure qt thread issue (just compiled qt w/ -thread option)	cleff	Linux - Software	8	05-08-2004 12:11 AM

All times are GMT -5. The time now is 03:44 PM.