A Jaywalker's Diary

Friday, December 31, 2010

Difference Between POSIX and Standard C Library

If everything you learn about Unix development using C/C++ is POSIX then what the hell does the standard C library do?

First things first. POSIX stands for "Portable Operating System Interface (for UniX)." Now that's a fancy name but the requirement becomes very clear when you see the disparity amongst so many flavors of Unix. There was a need to unify the programming of these flavors of Unix (include Linux); POSIX provides this unified API. Though the standard is not tied to just C, there is no known full implementation of POSIX in any other environment.

What does POSIX contain?

A lot of things! For example, threads, semaphores, file system access API, etc.

What's wrong with/ limiting in Standard C Library?

Standard C library (i.e., ANSI/ ISO C) assumes a very minimalist operating system. For example, no standard C library functions exist to manage threads. Heck, there is no IPC (inter process communication); in other words, ISO C doesn't assume that the program would be running on a multi-processing operating system!

Again, what exactly does POSIX add to the standard C library?

To answer this, one must know which header files ISO/ ANSI C contains. There are 24 header files including the well known `<stdio.h>`, `<stdlib.h>`, etc.
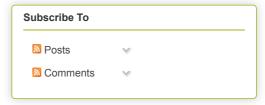
```
<assert.h> <complex.h>  <ctype.h>  <errno.h>  <fenv.h>
<float.h>  <inttypes.h> <iso646.h> <limits.h> <locale.h>
<math.h>   <setjmp.h>   <signal.h> <stdarg.h> <stdbool.h>
<stddef.h> <stdint.h>   <stdio.h>  <stdlib.h> <string.h>
<tgmath.h> <time.h>     <whcar.h>  <wctype.h>
```

As you can see, there is nothing like `<socket.h>`, `<thread.h>`, `<sharedmemory.h>`, `<graphics.h>`, etc., since this kind of functionality is heavily dependent on the kind of environment the application would run in while C was designed to be a "portable" language.

Not having this "advanced" functionality, C becomes severely limited and each operating environment is bound to introduce its own standard for defining multi-processing, interprocess communication, advanced memory management, etc. That's where POSIX (and other kinds of libraries such as graphics toolkit) kicks in.

If you have programmed for Unix/ Linux, you know that there are several library functions beyond system calls which are not in the standard C list. This includes functions like `fork()`, which is neither a system call nor part of the standard C library (remember C doesn't assume a multi-processing OS?). In fact, POSIX goes well beyond a programming API and defines the "environment" in which the application would run. For example, POSIX defines that there shall be a root directory "/" in the operating environment---no such definition is part of C standard library.

The complete list of POSIX header files should be referred as well. Below is a listing of the header files additional to ISO C:

```
<aio.h>        <arpa/inet.h>  <assert.h>     <ctype.h>
<dirent.h>     <dlfcn.h>      <fcntl.h>      <fmtmsg.h>
```

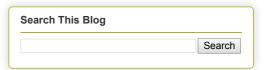## Popular Posts

TB Test for UK Visa: Finding IOM
TB Test for UK Visa in Pakistan Since March 2007, an additional requirement for Pakistani applicants of UK visas (6 months or more durati...

Getting NADRA B Form in Pakistan
This is one of the series where a location and a process are explained for the common good. The post uses the term "Bay Form" or...

Sending Money via Western Union from Pakistan
This is being written with a hope that an astray soul in search of the titled information will find this useful. You may have seen Wester...

Search This Blog

_____ [Search]

Labels

software (49)

life (45)

tools (29)

pakistan (27)

personal (23)

programming (23)

education (22)

sweden (22)

IT Pakistan (21)

world (19)

religion (16)

book review (15)

computer-science (14)

nature (14)

productivity (13)

money (11)

mortals (10)

science (9)

weird (9)

And there is...

Pakistani Drama Serials

Must Have Things!

Chip and PIN

```
<fnmatch.h>      <ftw.h>        <glob.h>        <grp.h>
<iconv.h>        <langinfo.h>   <libgen.h>      <monetary.h>
<mqueue.h>       <ndbm.h>       <net/if.h>      <netdb.h>
<netinet/in.h>   <netinet/tcp.h><nl_types.h>    <poll.h>
<pthread.h>      <pwd.h>        <regex.h>       <sched.h>
<search.h>       <semaphore.h>  <spawn.h>       <strings.h>
<stropts.h>      <sys/ipc.h>    <sys/mman.h>    <sys/msg.h>
<sys/resource.h> <sys/select.h> <sys/sem.h>     <sys/shm.h>
<sys/socket.h>   <sys/stat.h>   <sys/statvfs.h> <sys/time.h>
<sys/times.h>    <sys/types.h>  <sys/uio.h>     <sys/un.h>
<sys/utsname.h>  <sys/wait.h>   <syslog.h>      <tar.h>
<termios.h>      <trace.h>      <ulimit.h>
<unistd.h>       <utime.h>      <utmpx.h>       <wordexp.h>
```

Is POSIX a superset of standard C library?

Yes! POSIX, for example, redefines all standard C header files (sometimes augmenting them with advanced functionality), and uses the following statement in its documentation:

> The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1-2008 defers to the ISO C standard.

What's the use of standard C library, then?

Nothing if you are programming for a POSIX system but not all systems are POSIX compliant. To ensure maximum portability---ranging from desktop machines to handheld devices running a C runtime---restrict yourself to the standard C library and you can fly quite a distance with that.

Posted by Muhammad Ali Shah at 5:17 PM

Labels: programming

7 comments:

Anonymous 5:52 PM

fork() is a system call... (in fact sys_fork is number 2). Not trying to troll or anything but most of the information you provided regarding C and POSIX and how they relate is wrong. Look into the book 'Advanced Programming in the Unix Environment'. The beginning of this awesome book will explain the true differences and give you an idea of what POSIX truly means

Reply

jaywalker 6:52 PM

Thanks for the information, anonymous. While I would take a detailed look at the fork() comment, can you please enlighten me what else is wrong as you say that "most of the information" is wrong?
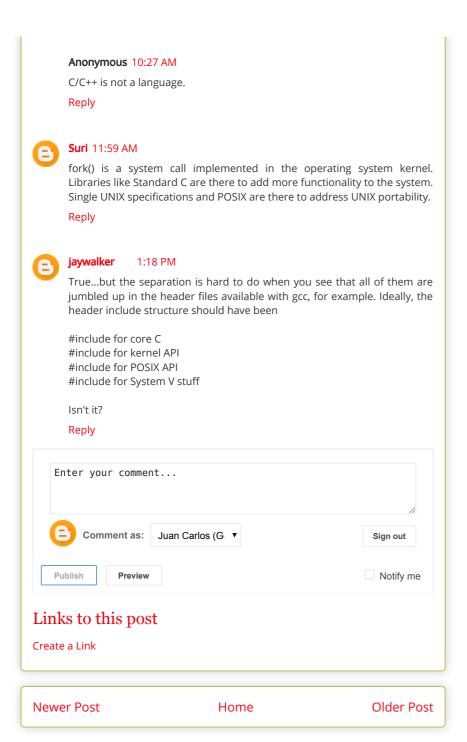
Reply

Anonymous 10:14 AM

thanks . valuable info.

Reply

Anonymous 9:36 AM

Very educational, thank you.

Reply

Anonymous 10:27 AM

C/C++ is not a language.

Reply

Suri 11:59 AM

fork() is a system call implemented in the operating system kernel. Libraries like Standard C are there to add more functionality to the system. Single UNIX specifications and POSIX are there to address UNIX portability.

Reply

jaywalker 1:18 PM

True...but the separation is hard to do when you see that all of them are jumbled up in the header files available with gcc, for example. Ideally, the header include structure should have been

#include for core C
#include for kernel API
#include for POSIX API
#include for System V stuff

Isn't it?

Reply

```
Enter your comment...
```

Comment as: Juan Carlos (G ▼   Sign out

Publish   Preview   ☐ Notify me

Links to this post

Create a Link

Subscribe to: Post Comments (Atom)