【Linux Programming】Day1

Tags	
Date	@May 13, 2022
Summary	

【Ch1】Getting Started

Linux Programs

When we type in to search for a file by name, the directories to search are stored in a shell variable, PATH.

The search path is configured by the system administrator and will usually contain some standard places where system programs are stored. These include:

- `/bin`: Binaries, programs used in booting the system
- `/usr/bin`: User binaries, standard programs available to users
- `/usr/local/bin`: Local binaries, programs specific to an installation

Linux uses the colon(`:`) character to separate entries in the PATH variable. Here's a sample PATH variable:

```
/usr/local/bin:/bin:/usr/bin:.:/home/neil/bin:/usr/X11R6/bin
```

Here the Path variable contains entries for the standard program locations, the current directory(.), a user's home directory, and the X Window System.

Development System Roadmap

Applications

Applications are usually kept in directories reserved for them. Applications supplied by the system for general use, including program development, are found in `/usr/bin`.

Applications added by system administrators for a specific host computer or local network are often found in `/usr/local/bin` or `/opt`.

Header Files

For C header files, they are almost always located in `/usr/include` and subdirectories thereof.

Other programming systems will also have header files that are stored in directories that get searched automatically by the appropriate compiler. Example: `/usr/include/c++` for GNU C++.

We can use header files in subdirectories or nonstandard places by specifying the `-I` flag(for include) to the C compiler. For example:

```
$ gcc -I /usr/openwin/include fred.c
```

It's often convenient to use the `grep` command to search header files for particular definitions and function prototypes.

Suppose we need to know the name of the `#define` used for returning the exit status from a program. Simply change to the `/usr/include` directory and grep for a portable part of the name like this:

```
$ grep EXIT_ *.h
```

Here grep searches all the files in the directory with a name ending in `.h` for the string `EXIT_`.

Library Files

Libraries are collections of precompiled functions that have been written to be reusable.

Standard system libraries are usually stored in `/lib` and `/usr/lib`. The C compiler needs to be told which libraries to search, because by default it searches only the

standard C library.

A library filename always starts with lib. Then follows the part indicating what library this is(like c for the C library, or m for the mathematical library).

The last part of the name starts with a dot(.), and specifies the type of the library:

- `.a` for traditional, static libraries
- `.so` for shared libraries

The libraries usually exist in both static and shared formats, as a quick `ls /usr/lib` will show.

We can instruct the compiler to search a library either by giving it the full path name or using the -l flag. For example,

```
$ gcc -o fred fred.c /usr/lib/libm.a
```

tells the compiler to compile `fred.c` and search the mathematical library in addition to the standard C library to resolve references to functions.

Static Libraries

The simplest form of library is just a collection of object files kept together in a ready-to-use form.

When a program needs to use a function stored in the library, it includes a header file that declares the function.

Static libraries, also known as archives, conventionally have names that end with `.a`.