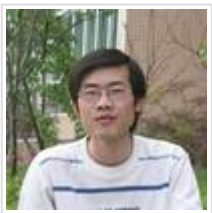


个人资料



lixianlin

访问： 535335次
积分： 6031
等级： 5
排名： 第4109名

原创： 115篇 转载： 13篇
译文： 2篇 评论： 108条

文章分类

- [Ajax](#) (5)
- [ActionScript](#) (1)
- [Anti-Virus](#) (11)
- [Android](#) (1)
- [ASP.NET](#) (1)
- [ASM](#) (2)
- [C & C++](#) (6)
- [C#](#) (5)
- [Crack](#) (3)
- [JavaScript](#) (12)
- [Linux](#) (9)
- [Lua](#) (2)
- [MySQL](#) (4)
- [Nodejs](#) (4)
- [Other](#) (48)
- [PHP](#) (8)
- [cocos2d-x](#) (1)
- [Redis](#) (2)
- [Security](#) (5)
- [SEO](#) (1)
- [System](#) (6)
- [VB](#) (2)
- [Web](#) (10)

文章存档

- [2014年05月](#) (2)
- [2014年04月](#) (2)
- [2013年09月](#) (1)
- [2013年04月](#) (1)
- [2012年10月](#) (4)

展开

一些不错的网站

- [看雪软件安全](#)
- [黑基网](#)
- [Google翻译](#)
- [爱词霸](#)
- [Yahoo!字典](#)
- [OGRE 3D](#)
- [OGRE 3D 中文](#)
- [FMOD](#)
- [Anti-Virus Sofeware Review](#)
- [Anti-Spyware Sofeware Review](#)
- [Virus Bulletin](#)
- [CKEditor](#)
- [PHP官方网站](#)
- [Apache官方网站](#)
- [MySQL官方网站](#)
- [SVN官方网站](#)
- [编程语言排名](#)
- [Linux Kernel](#)
- [W3C官方网站](#)
- [Memcached](#)
- [CentOS](#)
- [Ubuntu](#)
- [nginx官方网站](#)
- [redis官方网站](#)
- [GitHub](#)

常用在线手册

- [Apache HTTP Server官方手册](#)
- [CSS在线手册](#)
- [DHTML在线手册](#)
- [Google Maps API 参考](#)
- [JavaScript在线手册](#)
- [MSDN Library](#)

赠书 | [AI专栏（AI圣经！《深度学习》中文版）](#) 每周荐书：[Kotlin](#)、[分布式](#)、[Keras](#)（[评论送书](#)） [【获奖公布】征文 | 你会为 AI 转型么？](#)

linux c setitimer用法说明

标签： [setitimer](#) [c](#) [定时器](#) [计时器](#) [编程](#)

2014-05-12 12:25 9723人阅读 评论(0) 收藏 举报

分类： [C & C++](#) (5)

版权声明：本文为博主原创文章，未经博主允许不得转载。

在Linux c编程中，setitimer是一个比较常用的函数，可用来实现延时和定时的功能，网上有各种零零散散的用法说明，都只提到了个别用法，今天抽空实践整理了一份比较详细的：

使用时需要引入的头文件：

```
[cpp]
01. #include <sys/time.h>
```

setitimer函数原型：

```
[cpp]
01. int setitimer(int which, const struct itimerval *new_value,
02.              struct itimerval *old_value);
```

其中which参数表示类型，可选的值有：

ITIMER_REAL：以系统真实的时间来计算，它送出SIGALRM信号。

ITIMER_VIRTUAL：以该进程在用户态下花费的时间来计算，它送出SIGVTALRM信号。

ITIMER_PROF：以该进程在用户态下和内核态下所费的时间来计算，它送出SIGPROF信号。

紧接着的new_value和old_value均为itimerval结构体，先看一下itimerval结构体定义：

```
[cpp]
01. struct itimerval {
02.     struct timeval it_interval; /* next value */
03.     struct timeval it_value;    /* current value */
04. };
05.
06. struct timeval {
07.     time_t      tv_sec;         /* seconds */
08.     suseconds_t tv_usec;        /* microseconds */
09. };
```

itimeval又是由两个timeval结构体组成，timeval包含tv_sec和tv_usec两部分，其中tv_se为秒，tv_usec为微秒(即1/1000000秒)

其中的new_value参数用来对计时器进行设置，it_interval为计时间隔，it_value为延时时长，下面例子中表示的是在setitimer方法调用成功后，延时1微秒便触发一次SIGALRM信号，以后每隔200毫秒触发一次SIGALRM信号。

settimer工作机制是，先对it_value倒计时，当it_value为零时触发信号，然后重置为it_interval，继续对it_value倒计时，一直这样循环下去。

基于此机制，setitimer既可以用来延时执行，也可定时执行。

假如it_value为0是不会触发信号的，所以要能触发信号，it_value得大于0；如果it_interval为零，只会延时，不会定时（也就是说只会触发一次信号）。

old_value参数，通常用不上，设置为NULL，它是用来存储上一次setitimer调用时设置的new_value值。

以下是一个简单的使用例子：

```
[cpp]
01. #include <stdio.h>
02. #include <signal.h>
03. #include <sys/time.h>
04.
05. void signalHandler(int signo)
06. {
07.     switch (signo){
08.         case SIGALRM:
09.             printf("Caught the SIGALRM signal!\n");
10.             break;
11.     }
12. }
13.
14. int main(int argc, char *argv[])
15. {
16.     signal(SIGALRM, signalHandler);
17.
18.     struct itimerval new_value, old_value;
19.     new_value.it_value.tv_sec = 0;
20.     new_value.it_value.tv_usec = 1;
21.     new_value.it_interval.tv_sec = 0;
22.     new_value.it_interval.tv_usec = 200000;
23.     setitimer(ITIMER_REAL, &new_value, &old_value);
24.
25.     for(;;);
26.
27.     return 0;
28. }
```

