数据管理

张海宁

贵州大学

hnzhang1@gzu.edu.cn

April 26, 2018

Overview

内存管理

内存管理概述

在所有的计算机系统中，内存都是一种稀缺资源。而且无论有多少内存，似乎总是不够用，这就对内存的管理提出了挑战。
联想一下电脑和手机的内存增长史，应用程序所需要占用的内存也在随着内存容量的增长而增加。

内存分配 I

使用 c 语言标准库中的 malloc 来分配内存

```
#include<stdlib.h>
void *malloc(size_t size);
```

内存分配 II

使用 c 语言标准库中的 malloc 来分配内存

```
#include<stdlib.h>
void *malloc(size_t size);
```

文件锁定

文件锁定

文件锁定是多用户多任务操作系统中一个非常重要的组成部分。如果两个程序同时访问一个文件，一个读一个写，或者都在写，那么非常有可能出现文件内容不一致。Linux 使用文件锁定来解决这个问题。实现文件锁定有两种形式：

1. 创建锁文件
2. 锁定区域

创建锁文件 I

许多应用程序只要能够针对某个资源创建一个锁文件即可，然后其他的程序就可以通过检查这个文件的状态来判断它们自己是否被允许访问这个资源。
为了创建一个用途锁指示器的文件，使用 fcntl.h 头文件中定义的带 O_CREAT 和 O_EXECL 标志的 open 系统调用。这样能保证以一个原子操作同时完成两项工作：确定文件不存在，然后创建它。

创建锁文件 II

创建锁文件部分代码

```
int main(){
 int file_desc, save_errno;
  file_desc=open("LCK.test",O_RDWR|O_CREAT|O_EXCL,
     S_IRUSR);
  if(file_desc==-1){
   save_errno=errno;
   printf("Open failed with error code %d.\n",
     save_errno);
  }else{
   printf("Open succeeded.\n");
  }
 exit(EXIT_SUCCESS);
}
```

创建锁文件 III

执行第10页代码

```
$ ./lock1
Open succeeded.
$ ./lock1
Open failed with error code 17.
```

可以看到，第一次运行程序，会成功打开文件，第二次运行就会打开失败，其实不管再运行几次，都会失败。Why?
关于 errno 的更多信息可以查看第19页。

创建锁文件 IV

临界区：如果某个程序在执行的时候需要独占某个资源进行某操作，这部分操作被称为临界区，程序在进入临界区之前需要创建锁文件，在退出临界区之前需要删除（unlink）锁文件。

临界区代码

```
void writeAfile(){
 FILE *out = fopen("lcktest2","a");
 char st='a', end='z', c;
 for(c=st;c<end;c++){
  fputc(c,out);  fflush(out);
 }
 fputc('\n',out); fclose(out);
 printf("write done.\n");
}
```

创建锁文件 V

普通模式访问代码

```
int main(){
  int file_desc, tries=2;
  while(tries--){
    writeAfile();
  }
  exit(EXIT_SUCCESS);
}
```

创建锁文件 VI

使用锁文件方式

```
int main(){
 int file_desc, tries=2;
 while(tries--){
  file_desc=open(lock_file,O_RDWR|O_CREAT|O_EXCL,
    S_IRUSR);
  if(file_desc==-1){
   printf("%d : Lock already present.\n",getpid());
   sleep(3);   tries++;
  }else{
   printf("%d : Got access.\n",getpid());
   writeAfile(); close(file_desc);
   unlink(lock_file); sleep(1);
  } } exit(EXIT_SUCCESS);}
```

创建锁文件 VII

查看结果

普通方式执行程序

```
$./lock2n & ./lock2n
aabbccddeeffgghhiijjkkllmmnnooppqqrrssttuuvvwwxxyy

abcdeafbcdhefgiijklmnopqjrksltmnopqrstuvwxy
uvwxy
```

创建锁文件方式执行程序

```
$./lock2 & ./lock2
abcdefghijklmnopqrstuvwxy
abcdefghijklmnopqrstuvwxy
abcdefghijklmnopqrstuvwxy
abcdefghijklmnopqrstuvwxy
```

创建锁文件 VIII

通过第??页，可以看出，通过创建锁文件的方式，达到了一个程序单独占用某个文件的目的，确保了同一时间段只能有一个程序来独占一个文件进行操作。

数据库

The End

Appendix

errno

关于文件操作相关的错误代码所代表的错误信息可以查看:

```
/usr/include/sys/errno.h

\#define EEXIST          17
/* File exists */
```

本课程相关资源下载

1. ppt
   https://github.com/gmsft/ppt/tree/master/linux
2. 实验指导书
   https://github.com/gmsft/ppt/tree/master/book/linux

about man page

The manual is generally split into eight numbered sections, organized as follows (on Research Unix, BSD, macOS and Linux):

section	description
1	General commands
2	System calls
3	Library function(C standard library)
4	Special files(devices) and drivers
5	File formats and conventions
6	Games and screensavers
7	Miscellanea
8	System administration commands and daemons

Table: man page

在终端中运行 man read 与 man 2 read ， 观察其输出的区别。