

转

共享内存映射之mmap()函数详解

2014年10月05日 20:29:11

阅读数：3337

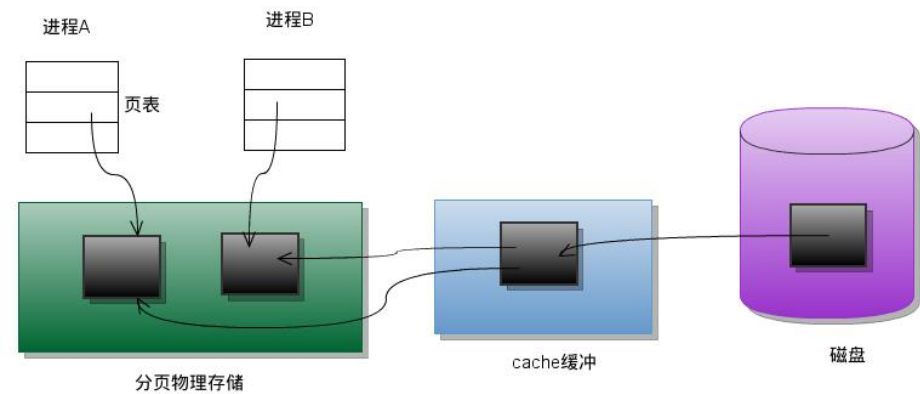
本文转自：<http://blog.csdn.net/chdhust/article/details/8159397>

共享内存映射之mmap()函数详解

共享内存可以说是最有用的进程间通信方式，也是最快的IPC形式,因为进程可以直接读写内存，而不需要任何数据的拷贝。对于像管道和消息队列等通信方式，则需要在内核和用户空间进行四次的数据拷贝，而共享内存则只拷贝两次数据: 一次从输入文件到共享内存区，另一次从共享内存区到输出文件。实际上，进程之间在共享内存时，并不总是读写少量数据后就解除映射，有新的通信时，再重新建立共享内存区域。而是保持共享区域，直到通信完毕为止，这样，数据内容一直保存在共享内存中，并没有写回文件。共享内存中的内容往往是在解除映射时才写回文件的。因此，采用共享内存的通信方式效率是非常高的。

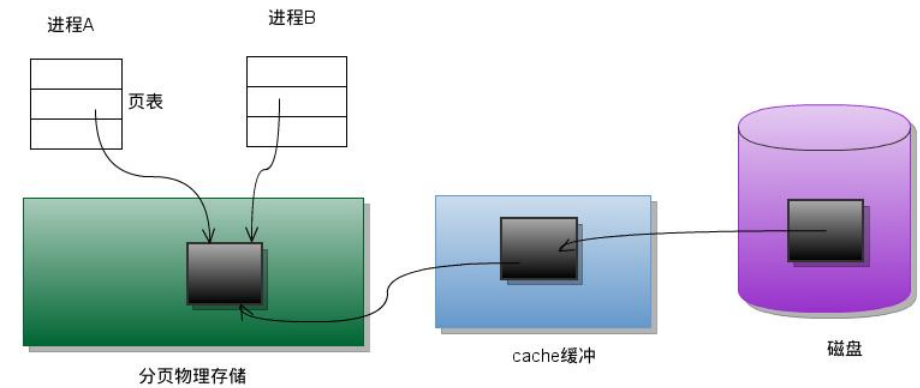
一. 传统文件访问

UNIX访问文件的传统方法是用open打开它们, 如果有多个进程访问同一个文件, 则每一个进程在自己的地址空间都包含有该文件的副本,这不必要地浪费了存储空间. 下图说明了两个进程同时读一个文件的同一页的情形. 系统要将该页从磁盘读到高速缓冲区中, 每个进程再执行一个存储体内的复制操作将数据从高速缓冲区读到自己的地址空间.



二. 共享存储映射

现在考虑另一种处理方法: 进程A和进程B都将该页映射到自己的地址空间, 当进程A第一次访问该页中的数据时, 它生成一个缺页中断. 内核此时读入这一页到内存并更新页表使之指向它.以后, 当进程B访问同一页面而出现缺页中断时, 该页已经在内存, 内核只需要将进程B的页表登记项指向次页即可. 如下图所示:



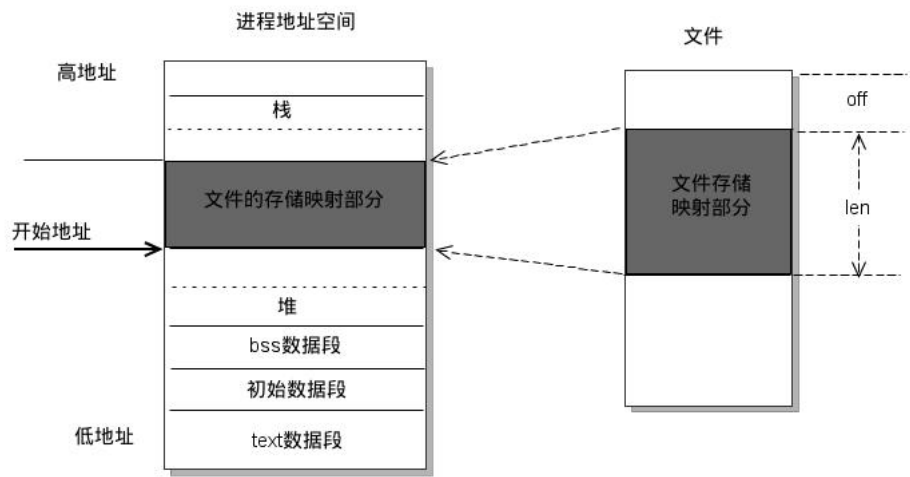
三、mmap()及其相关系统调用

mmap()系统调用使得进程之间通过映射同一个普通文件实现共享内存。普通文件被映射到进程地址空间后，进程可以向访问普通内存一样对文件进行访问，不必再调用read(), write () 等操作。

mmap()系统调用形式如下：

```
void* mmap ( void * addr , size_t len , int prot , int flags , int fd , off_t offset )
```

mmap的作用是映射文件描述符fd指定文件的 [off,off + len]区域至调用进程的[addr, addr + len]的内存区域, 如下图所示:



参数fd为即将映射到进程空间的文件描述字，一般由open()返回，同时，fd可以指定为-1，此时须指定flags参数中的MAP_ANON，表明进行的是匿名映射（不涉及具体的文件名，避免了文件的创建及打开，很显然只能用于具有亲缘关系的进程间通信）。

len是映射到调用进程地址空间的字节数，它从被映射文件开头offset个字节开始算起。

prot 参数指定共享内存的访问权限。可取如下几个值的或：PROT_READ（可读），PROT_WRITE（可写），PROT_EXEC（可执行），PROT_NONE（不可访问）。

flags由以下几个常值指定：MAP_SHARED，MAP_PRIVATE，MAP_FIXED，其中，MAP_SHARED，MAP_PRIVATE必选其一，而MAP_FIXED则不推荐使用。

offset参数一般设为0，表示从文件头开始映射。

参数addr指定文件应被映射到进程空间的起始地址，一般被指定一个空指针，此时选择起始地址的任务留给内核来完成。函数的返回值为最后文件映射到进程空间的地址，进程可直接操作起始地址为该值的有效地址。

四. mmap的两个例子

范例中使用的测试文件 data.txt:

Xml代码

```
01. aaaaaaaaaa
02. bbbbbbbbbb
03. ccccccccc
04. dddddddddd
```

1 通过共享映射的方式修改文件

C代码

```
01. #include <sys/mman.h>
02. #include <sys/stat.h>
03. #include <fcntl.h>
04. #include <stdio.h>
05. #include <stdlib.h>
06. #include <unistd.h>
07. #include <error.h>
08.
09. #define BUF_SIZE 100
10.
11. int main(int argc, char **argv)
12. {
13.     int fd, nread, i;
14.     struct stat sb;
15.     char *mapped, buf[BUF_SIZE];
16.
17.     for (i = 0; i < BUF_SIZE; i++) {
18.         buf[i] = '#';
19.     }
20.
21.     /* 打开文件 */
22.     if ((fd = open(argv[1], O_RDWR)) < 0) {
23.         perror("open");
24.     }
```

```
25.
26.  /* 获取文件的属性 */
27.  if ((fstat(fd, &sb)) == -1) {
28.      perror("fstat");
29.  }
30.
31.  /* 将文件映射至进程的地址空间 */
32.  if ((mapped = (char *)mmap(NULL, sb.st_size, PROT_READ |
33.      PROT_WRITE, MAP_SHARED, fd, 0)) == (void *)-1) {
34.      perror("mmap");
35.  }
36.
37.  /* 映射完后, 关闭文件也可以操纵内存 */
38.  close(fd);
39.
40.  printf("%s", mapped);
41.
42.  /* 修改一个字符,同步到磁盘文件 */
43.  mapped[20] = '9';
44.  if ((msync((void *)mapped, sb.st_size, MS_SYNC)) == -1) {
45.      perror("msync");
46.  }
47.
48.  /* 释放存储映射区 */
49.  if ((munmap((void *)mapped, sb.st_size)) == -1) {
50.      perror("munmap");
51.  }
52.
53.  return 0;
54. }
```

2 私有映射无法修改文件

```
/* 将文件映射至进程的地址空间 */
if ((mapped = (char *)mmap(NULL, sb.st_size, PROT_READ |
    PROT_WRITE, MAP_PRIVATE, fd, 0)) == (void *)-1) {
    perror("mmap");
}
```

五. 使用共享映射实现两个进程之间的通信

两个程序映射同一个文件到自己的地址空间, 进程A先运行, 每隔两秒读取映射区域, 看是否发生变化.

进程B后运行, 它修改映射区域, 然后推出, 此时进程A能够观察到存储映射区的变化

进程A的代码:

C代码

```
01. #include <sys/mman.h>
02. #include <sys/stat.h>
03. #include <fcntl.h>
04. #include <stdio.h>
05. #include <stdlib.h>
06. #include <unistd.h>
07. #include <error.h>
08.
09. #define BUF_SIZE 100
10.
11. int main(int argc, char **argv)
12. {
13.     int fd, nread, i;
14.     struct stat sb;
15.     char *mapped, buf[BUF_SIZE];
16.
17.     for (i = 0; i < BUF_SIZE; i++) {
18.         buf[i] = '#';
19.     }
20.
21.     /* 打开文件 */
22.     if ((fd = open(argv[1], O_RDWR)) < 0) {
23.         perror("open");
24.     }
25.
26.     /* 获取文件的属性 */
27.     if ((fstat(fd, &sb)) == -1) {
28.         perror("fstat");
```



0



写评论



收藏



微信



微博



QQ

```
29.     }
30.
31.     /* 将文件映射至进程的地址空间 */
32.     if ((mapped = (char *)mmap(NULL, sb.st_size, PROT_READ |
33.         PROT_WRITE, MAP_SHARED, fd, 0)) == (void *)-1) {
34.         perror("mmap");
35.     }
36.
37.     /* 文件已在内存, 关闭文件也可以操纵内存 */
38.     close(fd);
39.
40.     /* 每隔两秒查看存储映射区是否被修改 */
41.     while (1) {
42.         printf("%s\n", mapped);
43.         sleep(2);
44.     }
45.
46.     return 0;
47. }
```

进程B的代码:

C代码

```
01. #include <sys/mman.h>
02. #include <sys/stat.h>
03. #include <fcntl.h>
04. #include <stdio.h>
05. #include <stdlib.h>
06. #include <unistd.h>
07. #include <error.h>
08.
09. #define BUF_SIZE 100
10.
11. int main(int argc, char **argv)
12. {
13.     int fd, nread, i;
14.     struct stat sb;
15.     char *mapped, buf[BUF_SIZE];
16.
17.     for (i = 0; i < BUF_SIZE; i++) {
18.         buf[i] = '#';
19.     }
20.
21.     /* 打开文件 */
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录 注册 ×

```
25.
26.     /* 获取文件的属性 */
27.     if ((fstat(fd, &sb)) == -1) {
28.         perror("fstat");
29.     }
30.
31.     /* 私有文件映射将无法修改文件 */
32.     if ((mapped = (char *)mmap(NULL, sb.st_size, PROT_READ |
33.         PROT_WRITE, MAP_PRIVATE, fd, 0)) == (void *)-1) {
34.         perror("mmap");
35.     }
36.
37.     /* 映射完后, 关闭文件也可以操纵内存 */
38.     close(fd);
39.
40.     /* 修改一个字符 */
41.     mapped[20] = '9';
42.
43.     return 0;
44. }
```

0

写评论

收藏

微信

微博

QQ

六. 通过匿名映射实现父子进程通信

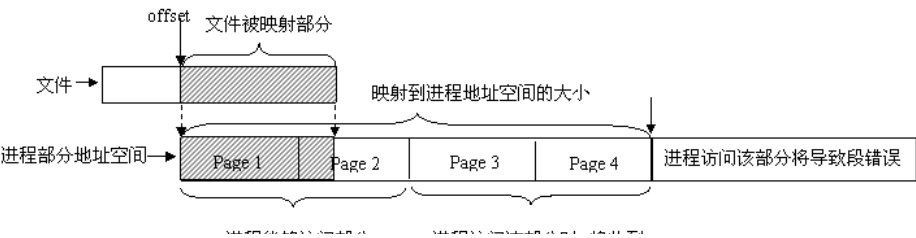
C代码

```
01. #include <sys/mman.h>
02. #include <stdio.h>
```

```
03. #include <stdlib.h>
04. #include <unistd.h>
05.
06. #define BUF_SIZE 100
07.
08. int main(int argc, char** argv)
09. {
10.     char *p_map;
11.
12.     /* 匿名映射,创建一块内存供父子进程通信 */
13.     p_map = (char *)mmap(NULL, BUF_SIZE, PROT_READ | PROT_WRITE,
14.         MAP_SHARED | MAP_ANONYMOUS, -1, 0);
15.
16.     if(fork() == 0) {
17.         sleep(1);
18.         printf("child got a message: %s\n", p_map);
19.         sprintf(p_map, "%s", "hi, dad, this is son");
20.         munmap(p_map, BUF_SIZE); //实际上, 进程终止时, 会自动解除映射。
21.         exit(0);
22.     }
23.
24.     sprintf(p_map, "%s", "hi, this is father");
25.     sleep(2);
26.     printf("parent got a message: %s\n", p_map);
27.
28.     return 0;
29. }
```

七. 对mmap()返回地址的访问

linux采用的是页式管理机制。对于用mmap()映射普通文件来说，进程会在自己的地址空间新增一块空间，空间大小由mmap()的len参数指定，注意，进程并不一定能够对全部新增空间都能进行有效访问。进程能够访问的有效地址大小取决于文件被映射部分的大小。简单的说，能够容纳文件被映射部分大小的最少页面个数决定了进程从mmap()返回的地址开始，能够有效访问的地址空间大小。超过这个空间大小，内核会根据超过的严重程度返回发送不同的信号给进程。可用如下图示说明：



加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录注册

进程能访问的大小。下面看一个实例：

0

写评论

收藏

微信

微博

QQ

```
01. #include <sys/mman.h>
02. #include <sys/types.h>
03. #include <sys/stat.h>
04. #include <fcntl.h>
05. #include <unistd.h>
06. #include <stdio.h>
07.
08. int main(int argc, char** argv)
09. {
10.     int fd,i;
11.     int pagesize,offset;
12.     char *p_map;
13.     struct stat sb;
14.
15.     /* 取得page size */
16.     pagesize = sysconf( SC_PAGESIZE);
17.     printf("pagesize is %d\n",pagesize);
18.
19.     /* 打开文件 */
20.     fd = open(argv[1], O_RDWR, 00777);
21.     fstat(fd, &sb);
```

```
22. printf("file size is %zd\n", (size_t)sb.st_size);
23.
24. offset = 0;
25. p_map = (char *)mmap(NULL, pagesize * 2, PROT_READ|PROT_WRITE,
26.     MAP_SHARED, fd, offset);
27. close(fd);
28.
29. p_map[sb.st_size] = '9'; /* 导致总线错误 */
30. p_map[pagesize] = '9'; /* 导致段错误 */
31.
32. munmap(p_map, pagesize * 2);
33.
34. return 0;
35. }
```

参考来源：
<http://wenku.baidu.com/view/e7d9b222bcd126fff7050b53.html>
<http://learn.akae.cn/media/ch28s09.html>
<http://blog.nosqlfan.com/html/2956.html>

文章标签： Linux 共享内存 mmap
个人分类： Linux 编程

老股民酒后无意说漏：20年炒股 坚持只看1指标

东正金融 · 顶新

想对作者说点什么？ 我来说两句

Linux 内存映射函数 mmap () 函数详解

mmap将一个文件或者其它对象映射进内存。文件被映射到多个页上，如果文件的大小不是所有页的大小之和，最后一个页不被使用的空间将会清零。mmap在用户空间映射调用系统中作用很大。头文件 函数原型...

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！ 登录 注册 ×

linux中mmap系统调用原理分析与实现

1、mmap系统调用（功能） void* mmap (void * size_t len , int prot , int flags ,int fd , off_t o...

yinjiabin 2012-05-17 11:48:42 阅读数：1

老中医说：男人多吃这个东西，时间

莫托 · 顶新

mmap函数用法及示例程序

UNIX网络编程第二卷进程间通信对mmap函数进行了用，这样用内存读写取代I/O读写，以获得较高的性能。该函数主要用途有三个： 1、将一个普通文件映射到内存中，通常在需要对文件进行频繁读写时使用

xiaohuima_dong 2015-03-26 22:38:11 阅 3654

共享内存:mmap函数实现

内存映射的应用: 以页面为单位, 将一个普通文件映射到内存中, 通常在需要对文件进行频繁读写时使用, 这样用内存读写取代I/O读写, 以获得较高的性能; 将特殊文件进行匿名内存映射, 可以为关联进程提供共享内存...

callinglove 2015-07-01 15:04:25 阅读数：3629

mmap()函数

mmap()函数将一个文件或者其它对象映射进内存。文件被映射到多个页上，如果文件的大小不是所有页的大小之和，最后一个页不被使用的空间将会清零。**mmap()**系统调用使得进程之间通过映射同一个普通文...

 leixiaolan

2013-11-25 14:25:00

阅读数：1371

内存映射mmap函数

函数**mmap()**将某个文件的指定内容映射到内存空间中，从而提供不同于一般的普通文件操作方式，进程可以像读写内存一样对普通文件进行操作。普通文件被映射到进程地址空间后，进程可以像访问普通内存一样对文件进...

 zhou8400

2014-11-06 11:35:27

阅读数：140

呼和浩特新出的赚钱方式，第一批人已经赚嗨翻了！

五洲七洋 · 顶新

mmap函数使用与实例详解

mmap 编辑 **mmap**将一个文件或者其它对象映射进内存。文件被映射到多个页上，如果文件的大小不是所有页的大小之和，最后一个页不被使用的空间将会清零。**mmap**在用户空间映射调用系统中作用...

 u013525455

2016-09-19 09:27:10

阅读数：3413

mmap的详细使用及函数参数意义

mmap 功能描述： **mmap**将一个文件或者其它对象映射进内存。文件被映射到多个页上，如果文件的大小不是所有页的大小之和，最后一个页不被使用的空间将会清零。**mmap**执行相反的操作，删除特定地...

 chary8088

2008-09-25 15:42:00

阅读数：6326

mmap函数使用

UNIX网络编程第二卷进程间通信对**mmap**函数进行了说明。该函数主要用途有三个：1、将一个普通文件映射到内存中，通常在对文件进行频繁读写时使用，这样用内存读写取代I/O读写，以获得较高的性能；2、...

 scorpio16

2008-01-22 17:03:00

阅读数：9870

TI C64X DSP中断向量表的配置（硬件中断）

1. 编写中断服务例程在.c源文件中编写ISR函数**c_intXX**，用于中断处理，如：interrupt void **c_intXX** (void){ ...;}注：对于硬件中断而言，XX ...

 henhen2002

2009-08-19 20:59:00

阅读数：2936

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

为64位CPU设计，性能提升10%。同时支持100MB/s的IO...


 sean_hy2009

2010-07-18 16:04:00

阅读数：690

linux 中**mmap**的用法

函数： void ***mmap**(void *start,size_t length,int prot,int fd,off_t offsize); 参数start：指向欲映...

 angelbosj

2012-12-08 16:27:17

阅读数：

驱动总结之**mmap**函数实现

原文地址：驱动总结之**mmap**函数实现 作者：gongp **mmap**作为struct file_operations的重要一个元素，**mmap**主要是实现物理内存到虚拟...

 bugaosuonia

2013-11-21 16:14:38


阅读数：9

对于程序员来说，英语到底多重要？

不背单词和语法，一个公式秒懂英语！

Mmap的实现原理和应用

很多文章分析了**mmap**的实现原理。从代码的逻辑来分析，总是觉没有把**mmap**后读写映射区域和普通的read/write联系起来。不得不产生疑问： 1，普通的read/write和**mmap**后的映射区域的...

 edwardlinux

2013-02-23 00:02:30

阅读数：12143

Linux的mmap内存映射机制解析

在讲述文件映射的概念时, 不可避免的要牵涉到虚存(SVR 4的VM). 实际上, 文件映射是虚存的中心概念, 文件映射一方面给用户提供了措施, 好似用户将文件映射到自己地址空间的某个部分, 使用...

zqxiao_09 2016-04-07 19:48:38 阅读数: 11391

计算机底层知识拾遗 (九) 深入理解内存映射mmap

内存映射mmap是Linux内核的一个重要机制, 它和虚拟内存管理以及文件IO都有直接的关系, 这篇细说一下mmap的一些要点。mmap和虚拟内存管理 先来看Linux内核的用户进程虚拟内存管...

ITer_ZC 2015-03-16 17:08:18 阅读数: 5022

mmap。

linux mmap 内存映射 mmap() vs read()/write()/lseek() 通过strace统计系统调用的时候, 经常可以看到mmap()与mmap2()。系统调用mmap()...

ryder001 2012-09-11 10:40:20 阅读数: 17675

mmap内存映射

linux mmap 内存映射 mmap() vs read()/write()/lseek() 通过strace统计系统调用的时候, 经常可以看到mmap()与mmap2()。系统调用mm...

vc0051127833 2012-05-15 15:25:11 阅读数: 5701

免费云主机试用一年

有哪些可以免费试用一年左右的云服务器



Linux共享内存 (mmap详解)

mmap()系统调用使得进程之间通过映射同一个普通文件实现共享内存。普通文件被映射到进程地址空间后, 进程可以向访问普通内存一样对文件进行访问, 不必再调用read(), write () 等操作。...

MJ813 2016-08-01 06:47:39 阅读数: 3832

进程间通信笔记 (8) —mmap详解

linux mmap详解

zhangxiao93 2016-09-22 10:53:25 阅读数: 1254

加入CSDN, 享受更精准的内容推荐, 与500万程序员共同成长!

登录

注册

×

于标识磁盘上你想用作内存映射文件的文件; 创建一个文件映...

go_Michael 2011-01-10 16:11:00 阅读数:

mmap详解

转载自: http://www.cnblogs.com/techdoc/archive/2012/1913521.html mmap 功能描述: mmap将一个文件或者其它对象映射进...

l534145398 2016-11-22 11:30:28 阅读数:

mmap - 用户空间与内核空间

mmap概述共享内存可以说是最有用的进程间通信方式, 是最快的IPC形式, 因为进程可以直接读写内存, 而不需要任何数据的拷贝。对于像管道和消息队列等通信方式, 则需要在内核和用户空间进行四次的数据拷贝...

luckywang1103 2016-02-02 20:06:20 阅读数: 411

50万码农评论: 英语对于程序员有多重要!

不背单词和语法, 老司机教你一个数学公式秒懂天下英语



Linux中的mmap的使用

利用ioremap访问硬件, 需要经过两次拷贝 1.明确: 不管是在用户空间还是在内核空间, 软件一律不能去直接访问设备的物理地址; 2.在内核驱动中如果要访问设备的物理地址, 需要利用ioremap...

qq_28090573

2016-03-26 16:36:57

阅读数：7158

mmap()

事例代码：//-----mmap.c-----...

duanbeibei

2007-07-25 21:25:00

阅读数：7004

Linux内存映射: mmap

Linux提供了内存映射函数mmap，它把文件内容映射到一段内存上(准确说是虚拟内存上)，通过对这段内存的读取和修改，实现对文件的读取和修改，先来看一下mmap的函数声明：头文...

dai_weitao

2007-07-25 17:09:00

阅读数：20558

mmap

mmap将一个文件或者其它对象映射进内存。文件被映射到多个页上，如果文件的大小不是所有页的大小之和，最后一个页不被使用的空间将会清零。mmap在用户空间映射调用系统中作用很大。 void* ...

u012199908

2017-11-05 21:20:21

阅读数：115

mmap 详解

谨以此文纪念过往的岁月 一.前言 mmap的具体实现以前在学习内核时学习过，但是对于其中的很多函数是一知半解的，有些只能根据其函数名来猜测其具体的功能，在本文中，一起来重新深入理解其具体的实现。 ...

prike

2017-06-24 13:50:52

阅读数：661

自己组装电脑配置清单四款

自己组装电脑



理解mmap

在接入日志组件xlog的工作中，对mmap内存映射加深了了解，分享一下学习心得。...

u012455213

2017-02-25 19:50:40

阅读数：386

内存映射mmap

Table of Contents 1. 什么是mmap 2. 使用方法2.1. mmap构造器的格式 2.2. 例子1 2.3. 例子2: 3. 其它 4. 参考资料什么是mmap通常在Unix系统...

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

mmap 实现原理和应用（派性）

在调试完达芬奇的视屏输入后记录一下调试心得。重点还是再次放在系统调用mmap后的内存分配上。做了一个mmap系统调用的笔记以及记录以备后用。大致介绍了linux中对于mmap的应用以及后续产生缺页异...

edwardlinux

2016-07-27 15:57:26

阅读数：055

Linux那些事儿之我是Sysfs(7)dentriode

我们在进程中要怎样去描述一个文件呢？我们用目录entry和索引节点(inode)。它们的定义如下:struct dentry {struct inode ...

fudan_abc

2007-09-06 22:09:00

阅读数：收藏

Python之mmap内存映射模块（大文理解）说明

http://www.cnblogs.com/zhoubinyi/p/6062907.html 通常在UNIX下面处理文本文件的方法是sed、awk等shell命令...

zdy0_2004

2016-11-17 15:06:06

阅读数：

50万码农评论：英语对于程序员有多重要！

不背单词和语法，老司机教你一个数学公式秒懂天下英语



python 、 mmap 实现内存数据共享

```
import mmap mmap_file = None ##从内存中读取信息， def read_mmap_info(): global mmap_file mmap_f...
```

 zhaohongyan6

2017-05-04 10:09:52

阅读数：2035

mmap映射区和shm共享内存的区别总结

linux中的两种共享内存。一种是我们的IPC通信System V版本的共享内存，另外的一种就是我们今天提到的存储映射I/O（mmap函数） 在说mmap之前我们先说一下普通的读写文件的原理，进...

 hj605635529

2017-06-13 11:45:24

阅读数：3981

Sphinx 的介绍和原理探索

What/Sphinx是什么 定义：Sphinx是一个全文搜索引擎。特性： 索引和性能优异易于集成SQL和XML数据源，并可使用SphinxAPI、SphinxQL或者Sphi...

 yk19851111

2016-06-01 09:01:24

阅读数：3272

coreseek搭建教程

说明 coreseek官网(<http://www.coreseek.cn/>)已经关闭。可参考手册：<https://my.oschina.net/alexwu/blog/71155> coreseek是...

 Time888

2016-09-20 17:42:04

阅读数：10325

mmap参数详解

mmap 功能描述： mmap将一个文件或者其它对象映射进内存。文件被映射到多个页上，如果文件的大小不是所有页的大小之和，最后一个页不被使用的空间将会清零。munmap执行相反的操作，删除...

 majiakun1

2013-01-31 10:57:13

阅读数：567

[传播公益品牌，支持慈善事业](#)
为公益机构提供免费推广服务
gongyi.baidu.com

mmap：是什么 为什么 怎么用

mmap基础概念 mmap是一种内存映射文件的方法，即将一个文件或者其它对象映射到进程的地址空间，实现文件磁盘地址和进程虚拟地址空间中一段虚拟地址的一一对映关系。实现这样的映射关系后，进程就可以...

 ebw123

2015-12-23 23:06:07

阅读数：990

mmap内核源码分析

对于mmap函数，我之前的理解太单一了。这几天好好复习了一下以前学过的知识，重新对该函数有了新的认识。之前我的认识是，mmap是用来映射内存的，它映射的内存来自磁盘上的文件，所以我们可以为mmap函数提供...

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

mmap的真正作用

最近看了一些关于linux内核源代码解析的说，发现从下： #include void...

 supenman_mwg

2013-09-28 21:15:21

阅读数：2510

open, ioctl, mmap函数解释

open("/dev/binder", O_RDWR); 打开"/dev/binder"设备文件。int ioctl(int fd, ind cmd, ...); ioctl是设备驱动程序中...

 qq_28077303

2016-01-15 18:56:28

阅读数：6

linux mmap()函数

<http://baike.baidu.com/view/1499209.htm> mmap 将一个文件或者其它对象映射进内存。文件被映射到多个页上...

 zxygww

2013-11-22 15:38:58

阅读数：11

50万码农评论：英语对于程序员有多重要？

不背单词和语法，一个公式学好英语



linux基础编程 共享内存 使用内存映射接口mmap系统调用 blog.csdn.net/ghostyu/article/details/7736733

linux基础编程 共享内存 使用内存映射接口mmap系统调用 分类： linux系统编程 2012-07-11 15:24 1607人阅读 评论(0) 收藏 举报 linu...

wangyin159 2015-09-15 16:21:17 阅读数：504

存储映射I/O（mmap函数）和共享内存

前言最近在一次电话面试的过程中，面试官问我相关的共享内存的知识，当时就认为是我在学习IPC进程间通信学习到的共享内存，所以就给面试官讲解了一下，然后，面试官说我是否知道在Linux下的存储映射I/O，...

qq_26768741 2017-03-28 15:26:14 阅读数：908

文件内存映射mmap解决大文件快速读写问题和进程间共享内存

mmap函数主要用途有三个： 1、将一个普通文件映射到内存中，通常在需要对文件进行频繁读写时使用，这样用内存读写取代I/O读写，以获得较高的性能； 2、将特殊文件进行匿名内存映射，可以为关...

xhu_eternalcc 2014-08-09 11:08:32 阅读数：3394

mmap映射文件至内存(实现 共享内存 与 文件的另类访问)

Linux下mmap函数的介绍 (2013-5-20 14:46) 标签： mmap函数, Linux mmap函数是unix/linux下的系统调用，mmap系统调用并不是完全为了用...

u011170660 2014-08-01 18:25:13 阅读数：2727

linux共享存储映射mmap

共享存储映射 文件进程间通信 使用文件也可以完成IPC，理论依据是，fork后，父子进程共享文件描述符。也就共享打开的文件。 练习：编程测试，父子进程共享打开的文件。借助文件进行进程间通信...

oguro 2016-12-23 17:59:24 阅读数：1150

python学习路线

初学python 请教学习路线



进程分配内存的两种方式--brk() 和mmap()（不设计共享内存）

如何查看进程发生缺页中断的次数？ 用ps -o majflt,minflt -C program命令查看。 majflt代表major fault，中文名叫大...

yusiguyuan 2014-09-23 11:31:56 阅读数：3554

进程与内存4-mmap实现1（nopage方法原理及实例）

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

...

0

写评论

收藏

微信

微博

QQ

个人资料

关注

原创	粉丝	喜欢	评论
6	36	22	7

等级： 博客 5

积分： 3226

访问： 27万+

排名： 1万+



最新文章

- 使用vagrant+VirtualBox搭建虚拟开发环境
- stratum协议原理
- 深度解析挖矿的逻辑和技术实现
- 【区块链开发指南】区块链基础之挖矿和矿池
- 【区块链开发指南】区块链基础之区块和交易

个人分类

Python	123篇
Linux	69篇
C语言	7篇
C++	14篇
算法	41篇

展开

归档

2018年5月	1篇
2018年4月	6篇
2018年3月	4篇
...	...

登录 注册 X

展开

热门文章

- 几种内存泄露检测工具的比较
阅读量：15667
- python安装模块时报异常：Compression requires the (missing) zlib module
阅读量：8872
- Python ctypes 模块
阅读量：8243
- virtualenv详解
阅读量：7938
- Python调用windows下DLL详解 - ctypes库的使用
阅读量：7297

最新评论

- python图形处理库PIL(Py...
Catherine981234： [python] view plain copyimport StringIO img = ...
- 十月百度，阿里巴巴，迅雷搜狗最新面...
qq_35407317： 可以整理发在博客吗

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！


0


写评论


收藏


微信


微博


QQ

- python安装模块时报异常：Co...
qq_35468007：感谢！
- 解决安装pycrypto时的错误： ...
Neptune1997：感谢博主
- 深入浅出 Python 装饰器： 1...
rainivey：希望改改错别字，有点影响阅读，写的倒是不错

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×


0


写评论


收藏


微信


微博


QQ