

## Linux I/O复用技术实现极致性能（linuxio复用）

---

Linux I/O复用技术是目前Linux操作系统的一个非常重要的特性，它使得程序可以同时监视多个I/O请求，有效地处理大量事件，而不占用大量处理资源，这就是I/O复用技术能够实现极致性能。下面让我们一起来看看Linux I/O复用技术中实现极致性能的一些具体方法。

首先，Linux I/O复用技术能够有效率地实现极佳的资源利用率。这是通过引入事件驱动架构来实现的，它可以充分利用操作系统在读取和写入I/O请求上面的一次性操作，从而可以非常有效地实现各种复杂的I/O任务，而不必不断的占用操作系统的处理资源，从而可以实现非常高的性能。

其次，Linux I/O复用技术能够更加有效地实现任务的调度和分配，这是通过使用一些高级的功能，如`epoll()`和`poll()`。`epoll()`能够有效地监听I/O事件，而`poll()`则能够有效地进行数据通信。这样一来，两个框架就可以有效组合起来实现I/O任务的有效调度和分配，提升系统的I/O效率，达到极佳的性能。

最后，还需要强调的是Linux I/O复用技术支持并发编程，这对于实现极致性能是非常重要的。使用Linux I/O复用技术，程序可以同时监控多个I/O请求，通过引入多线程技术将I/O任务并发完成，从而可以有效降低系统资源占用，实现更佳性能。

因此，Linux I/O复用技术能够实现极高的性能，来应对系统中不断上升的I/O要求。下面是一个示例代码，它用来展示Linux I/O复用技术在实现有效率的I/O控制和分配的方法。

```
#include

#define MAXEVENTS 64

int main(int argc, char *argv[])

{

int epfd, nfds, fd;

struct epoll_event ev, events[MAXEVENTS];

int i;

epfd = epoll_create1 (0);

if (epfd == -1) {

perror ("epoll_create");

exit (EXIT_FAILURE);

}

ev.events = EPOLLIN;

ev.data.fd = fd;
```

```
if (epoll_ctl (epfd, EPOLL_CTL_ADD, fd, &ev) == -1) {

    perror ("epoll_ctl");

    exit (EXIT_FAILURE);

}

for (;;) {

    nfds = epoll_wait (epfd, events, MAXEVENTS, -1);

    if (nfds == -1) {

        perror ("epoll_wait");

        exit (EXIT_FAILURE);

    }

    for (i=0; i

        if (events[i].events & EPOLLIN) {

            /* 执行处理I/O任务 */

        }

    }

}
```

```
close (epfd);
```

```
return 0;
```

```
}
```

通过上面的介绍可以看出，Linux I/O复用技术是一种非常有效率的操作系统特性，它能够通过简单而又高效的方式来实现极佳的性能，而不必耗费大量的处理资源。因此，掌握Linux I/O复用技术可以帮助开发者实现极致性能的程序，缩短程序的发布时间，更有效地服务于用户。

[Linux](#)[Linux教程](#)[Linux资讯](#)

## ■ 我想要获取技术服务或软件

服务范围：MySQL、ORACLE、SQLSERVER、  
MongoDB、PostgreSQL、程序问题

服务方式：远程服务、电话支持、现场服务，沟通指定方式服务

技术标签：数据恢复、安装配置、数据迁移、集群容灾、异常处理、其它问题

沟通购买：[👤QQ咨询](#)    [淘宝咨询](#)    [微信咨询](#)    [淘宝店铺](#)

## ■ 版权申明及联系

本站部分文章参考或来源于网络，如有侵权请联系站长。本站提供相关远程技术服务，有需要可联系[👤QQ](#)

数据库远程运维 » [Linux I/O复用技术实现极致性能（linuxio复用）](#)

分享到：



上一篇

查询优化Oracle数据库的慢查询（oracle慢）

下一篇

解析Redis：快速入门Redis列表数据结构（listredis）

## 相关推荐



thumbnail

configLinux 下使用 ifconfig 管理网络连接（linux中的if）



thumbnail

探究MP4在Linux操作系统下的应用及优势（mp4linux）



thumbnail

Linux下如何重新启动Samba服务？（Linux重启samba）



thumbnail

Linux虚拟机设置静态IP地址指南（linux虚拟机静态ip）



thumbnail

“Linux掌控全局：一款高效的服务器连接软件” (linux服务器连接软件)

Linux JDK轻松卸载指南 (linuxjdk卸载)



妙用Linux：分区顺序指南 (linux分区顺序)

Linux下架设WebDAV服务的步骤 (linuxwebdav)

## 随机文章



Linux双向管道: 进入新世界的快乐之门 (linux双向管道)



深入了解Linux .m2文件夹及其重要性 (linux.m2)



的优势利用LKM Linux实现功能的更大优势 (lkmlinux)



at更新捷径：Linux系统快速升级实践 (linuxupd)



环境搭建极速搭建LinuxPHP企业级网站环境 (linuxphp网站)

thumbnail

98 (mysql教程15-17)



thumbnail

## Linux 上时间同步设置的指导步骤 (linux时间同步设置)

### 标签

[Linux](#)

[Linux教程](#)

[Linux资讯](#)

[MacOS](#)

[MacOS教程](#)

[MacOS资讯](#)

[MongoDB](#)

[MongoDB教程](#)

[MongoDB资讯](#)

[MSSQL](#)

[MSSQL错误](#)

[MySQL](#)

[mysql教程](#)

[MySQL维护](#)

[MySQL资讯](#)

[Neo4j](#)

[Neo4j教程](#)

[Neo4j资讯](#)

[ORACLE](#)

[Oracle优化](#)

[oracle内部视图](#)

[oracle参数](#)

[oracle开发](#)

[oracle异常修复](#)

[oracle故障处理](#)

[oracle教程](#)

[oracle维护](#)

[oracle视图](#)

[ORACLE资讯](#)

[oracle远程维护](#)

[ORA错误码](#)

[Redis](#)

[Redis教程](#)

[Redis资讯](#)

[SQLServer](#)

[SQLServer报错](#)

[SQLServer教程](#)

[SQLServer资讯](#)

[SQL修复](#)

[SQL异常](#)

[SQL远程处理](#)

[Windows](#)

[技术文档](#)

[操作系统](#)

[数据库](#)

本站部分资源来自互联网收集,仅供用于学习和交流,本站一切资源不代表本站立场,如有侵权、后门、不妥请联系本站站长删除

© 2020 Theme by - 数据服务 . All rights reserved 苏ICP备15021567号