

原创

华为云开发者社区

2021-08-28 14:57:44

 378

 收藏 5

版权

分类专栏：

技术交流

文章标签：


linux

python

gpio

sysfs

value

 技术交流 专栏收录该内容

95 订阅

2314 篇文章

订阅专栏

摘要：本文介绍GPIO的读写，介绍基本原理，以及不同读写方式的性能。

本文分享自华为云社区《[Linux 基于sysfs的GPIO读写操作](#)》，作者：一颗小树x。

前言

最近接触到 [Linux](#) 系统中的GPIO开发，这里做个小总结，也分享一下；本文会介绍GPIO的读写，介绍基本原理，以及不同读写方式的性能。

一、GPIO sysfs interface 基本原理

在Linux中，最常见的读写GPIO方式就是用GPIO sysfs interface，是通过操作/sys/class/gpio目录下的export、unexport、gpio{N}/direction、gpio{N}/value（用实际引脚号替代{N}）等文件实现的，经常出现shell脚本里面。

首先声明GPIO口，比如GPIO258：（这个命令需要root权限）

```
echo 258 > /sys/class/gpio/export
```

然后在/sys/class/gpio，会生成一个新目录，其名字叫gpio258

比如：GPIO258，N对应是258；进入其目录：cd /sys/class/gpio/gpio258

依次能看到：active_low、device、direction、edge、power、subsystem、uevent、value

其中比较常用的是value文件，它存放GPIO的值；范围是：0或1。我们可以直接对它读写，达到读写GPIO的效果。

二、定义GPIO为输入

比如定义GPIO258为输入，命令如下：（这个命令需要root权限）

```
echo in > /sys/class/gpio/gpio258/direction
```

三、定义GPIO为输出

比如定义GPIO258为输出，命令如下：（这个命令需要root权限）

```
echo out > /sys/class/gpio/gpio258/direction
```

四、读GPIO的值

我们可以直接对/sys/class/gpio/gpio258 目录下的value文件（存放GPIO的值），进行读取。

可以用cat 查看读GPIO的值，比如查看GPIO258的值：

```
cat /sys/class/gpio/gpio258/value
```

目录

前言

- 一、GPIO sysfs interface 基本原理
- 二、定义GPIO为输入
- 三、定义GPIO为输出
- 四、读GPIO的值
- 五、写GPIO的值
- 六、小案例——设置GPIO为输入，并读...
 - 方式一：纯shell命令
 - 方式二：shell命令 + Python程序（ ...
- 七、小案例——设置GPIO为输出，并读...
 - 方式一：纯shell命令

分类专栏

	程序员	369篇
	云图说	17篇
	DevCloud · 敏捷智库	44篇
	华为助力企业上云	398篇
	项目管理	23篇
	官方活动	152篇
	技术交流	2314篇
	开源社区	61篇
	华为故事	60篇

value 只是一个文件，可以通过其他方式读取；写一个python程序，实现读取GPIO的值：

```
1 | # 定义一个函数，用于读取GPIO258的值。
2 | def read_258():
3 |     with open('/sys/class/gpio/gpio258/value', 'r') as f:
4 |         io_258 = int(f.read())
5 |         print("read_258:%d"%io_258)
6 |
7 | # 调用函数
8 | read_258()
```

五、写GPIO的值

我们可以直接对/sys/class/gpio/gpio258 目录下的value文件（存放GPIO的值），进行写值。

可以用ehco 写GPIO的值，比如写GPIO258的值：

```
1 | echo 1 > /sys/class/gpio/gpio258/value           # output logic 1 level
2 | echo 0 > /sys/class/gpio/gpio258/value           # output logic 0 level
```

写一个python程序，实现写GPIO的值：

```
1 | # 定义一个函数，用于写GPIO258的值。
2 | def write_258(io_str):
3 |     with open('/sys/class/gpio/gpio258/value', 'w+') as f:
4 |         f.write(io_str)
5 |         print("write_258:%s"%(io_str))
6 |
7 | # 调用函数
8 | write_258()
```

经过测试，程序进行一次写操作，耗时0.6ms左右；ehco方式就比较久了，10ms左右。

六、小案例——设置GPIO为输入，并读取IO值

方式一：纯shell命令

```
1 | # 设置GPIO20为输入
2 | echo 20 > /sys/class/gpio/export
3 | echo in > /sys/class/gpio/gpio20/direction
4 |
5 | # 读取IO值
6 | cat /sys/class/gpio/gpio20/value
```

方式二：shell命令 + Python程序（效率更高）

```
1 | # 设置GPIO20为输入
2 | echo 20 > /sys/class/gpio/export
3 | echo in > /sys/class/gpio/gpio20/direction
```

读取IO值：

```
1 # 定义一个函数，用于读取GPIO258的值。
2 def read_20():
3     with open('/sys/class/gpio/gpio20/value', 'r') as f:
4         io_20 = int(f.read())
5         print("read_20:%d"%io_20)
6
7 # 调用函数
8 read_20()
```

七、小案例——设置GPIO为输出，并读写IO值

方式一：纯shell命令

```
1 # 设置GPIO40为输出
2 echo 40 > /sys/class/gpio/export
3 echo out > /sys/class/gpio/gpio40/direction
4
5 # 写IO值，高电平
6 echo 1 > /sys/class/gpio/gpio40/value
7
8 # 写IO值，低电平
9 echo 0 > /sys/class/gpio/gpio40/value
```

方式二：shell命令 + Python程序（效率更高）

```
1 # 设置GPIO40为输出
2 echo 40 > /sys/class/gpio/export
3 echo out > /sys/class/gpio/gpio40/direction
```

读写IO值：

```
1 import time
2
3 def read_40():
4     with open('/sys/class/gpio/gpio40/value', 'r') as f:
5         io_40 = int(f.read())
6         print("read_40:%d"%io_40)
7
8 def write_40(io_str):
9     with open('/sys/class/gpio/gpio40/value', 'w') as f:
10        f.write(io_str)
11        print("write_40:%s"%(io_str))
12
13 start = time.time()
14 read_40()
15 write_40("1")
16 read_40()
17 end = time.time()
18 print("读写IO的时间", end-start)
```

效果：耗时0.6ms。

```
read_40:0
write_40:1
read_40:1
读写IO的时间 0.0006775856018066406
```



华为云开发者...



2804

243

21

601万+



原创

周排名

总排名

访问

等级

8万+

4万+

1万+

3809

4万+

积分

粉丝

获赞

评论

收藏

私信

关注

搜博主文章

热门文章

100 个网络基础知识普及，看完成半个网络高手 173477

周杰伦新歌《说好不哭》上线，程序员哭了.....【华为云分享】 94510

挑战10个最难的Java面试题（附答案）【上】 77989

35岁+的大龄程序员：面对“中年危机”，这份书单或许能帮到你 75468

终于知道为啥网页不让我复制粘贴了！ 38041

最新评论

下班约会时来了新需求，咋办？
OuuuJun: 知道了，请问女朋友哪里领？
你真的懂Redis的5种基本数据结构吗？
Datrilla: 还没有用过 完锤子 我还是先用我的多哨兵队列。我的数据没法唯一key 还...
小熊派开发实践 | 小熊派+合宙 Cat.1接入...
可是我不是石头: 您好 附件在哪儿呀 我没有找到 不好意思
Java编程中忽略这些细节，Bug肯定找上你
我叫晨曦啊: 第三个讲解倒数第三行代码，应该是这一句吧 System.out.println(sb.le...
性能提升1400+倍，快来看MySQL Volca...
weixin_38329873: 有点东西

您愿意向朋友推荐“博客详情页”吗？

强烈不推荐

不推荐

一般般

推荐

强烈推荐

最新文章

存算分离架构的高斯Redis，用强一致提供可靠保障

带你了解AKG正反向算子注册+关联流程

软件开发除了23种设计模式，还有7个开发原

点击关注，第一时间了解华为云新鲜技术~

华为云开发者社区

关注

0

1

5

专栏目录

请发表有价值的评论， 博客评论欢迎灌水，良好的社区氛围需大家一起维护。

评论

宇宙爆肝锦标赛冠军：受益匪浅！ 2 月前 回复 ...

性能提升1400+倍，快来看MySQL Volcano模型迭代器的谓词位置优化详解 1万+

摘要：性能提升1400+倍，快来看MySQL Volcano模型迭代器的谓词位置优化详解。

新生代农民工的十八般武艺，你都了解吗 1万+

摘要：作为新生代农民工，行走江湖，必身怀秘技，“码农”的十八般武艺，你都了解吗？

想知道你未来宝宝长什么样？ 1万+

摘要：本案例可根据一张父亲和母亲的正脸照片，生成孩子的照片，并且可以调节参数，看看不同性别和年龄孩子的长相。

终于知道为啥网页不让我复制粘贴了！ 3万+

摘要：我们在项目中，通常为了保证数据安全性和保护用户个人信息，个别页面需要禁用右键、复制、粘贴以及输入框自动填充的功能。所以在此补充和...

存算分离架构的高斯Redis，用强一致提供可靠保障 8

摘要：其实开源Redis的弱一致性已经不能满足很多应用场景的诉求。怎么，不信？

带你了解AKG正反向算子注册+关联流程 21

摘要：简要介绍一下akg正反向算子的注册和关联流程。

软件开发除了23种设计模式，还有7个开发原则需要了解 45

摘要：使用设计模式能够使我们开发的程序，易维护、可拓展，可复用。但是在23个设计模式的背后，还有7个开发原则去支撑着设计模式，保证23个设...

Sechunter移动应用隐私合规检测详解 108

摘要：本文简要介绍Sechunter移动应用隐私合规检测的方法步骤，以及目标检测技术在其中的应用。

数仓如何限制临时数据文件下盘量 34

摘要：查询的中间结果集如果太大导致落盘生成的临时数据文件，本文提供两种限制临时数据文件下盘数据量的方案，以防影响正常业务运行。

填坑总结：python内存泄漏排查小技巧 122

摘要：最近服务遇到了内存泄漏问题，运维同学紧急呼叫解决，于是在解决问题之余也系统记录了下内存泄漏问题的常见解决思路。

关于HTTPS认证，这里解决你所有疑惑 47

摘要：从签发证书到数据加密交互，按流程的进展讲解HTTPS认证过程内容和原理。

分析师机构发布中国低代码平台现状分析报告，华为云AppCube为数字化转型加码 554

摘要：Forrester指出，中国企业数字化转型过程中，有58%的决策者正在采用低代码工具进行软件构建，另有16%的决策者计划采用低代码。

云小课 | SA基线检查—给云服务的一次全面“体检” 46

摘要：华为云态势感知（Situation Awareness，SA）是华为云安全管理与态势分析平台，能够检测出超过20大类的云上安全风险，利用大数据分析技术，...

并发高？可能是编译优化引发有序性问题 269

摘要：CPU为了对程序进行优化，会对程序的指令进行重排序，此时程序的执行顺序和代码的编写顺序不一定一致，这就可能会引起有序性问题。

不止承上启下，带你了解工业物联网关 473

摘要：本文将介绍工业领域当中国网场景下网关的技术细节。

面试只要问到分布式，必问分布式锁 182

摘要：分布式知识是考验一个程序员知识面广度和深度很好的度量标准，而分布式锁又是其中非常重要的一个知识点。

你真的懂Redis的5种基本数据结构吗？ 1116

摘要：你真的懂Redis的5种基本数据结构吗？这些知识点或许你还需要看看。



则需要了解

2021			
11月 71篇	10月 73篇	09月 94篇	08月 97篇
07月 102篇	06月 104篇	05月 83篇	04月 103篇
03月 100篇	02月 70篇	01月 93篇	
2020年 931篇		2019年 1031篇	

从原理带你掌握Spring MVC拦截处理器知识

173

摘要：SpringWebMVC的处理器拦截器，类似于Servlet开发中的过滤器Filter，用于处理器进行预处理和后处理。

Java编程中忽略这些细节，Bug肯定找上你

568

摘要：在Java语言的日常编程中，也存在着容易被忽略的细节，这些细节可能会导致程序出现各种Bug。

论文解读二十七：文本行识别模型的再思考

1099

摘要：本文研究了两个解码器（CTC[1]和Transformer[2]）和三个编码器模块（双向LSTM[3]、Self-Attention[4]和GRCL[5]），通过大量实验在广泛使用的...

关于我们 招贤纳士 广告服务 开发助手 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 网络110报警服务 中国互联网举报中心 家长监护 Chrome商店下载 ©1999-2021北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照

