

2021年11月						
日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

公告

昵称: DoubleLi
园龄: 11年10个月
粉丝: 2088
关注: 29
+加关注

搜索

- 常用链接
- [我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

- 随笔分类 (5195)
- [android\(2\)](#)
[ASP.NET\(30\)](#)
[ASP.NET MVC\(11\)](#)
[Boost\(118\)](#)
[c#\(10\)](#)
[C++/C\(785\)](#)
[c++11\(15\)](#)
[cmake/autotool\(66\)](#)
[com/ATL/Activex\(75\)](#)
[Css\(16\)](#)
[CImage\(12\)](#)
[darwin stream server\(3\)](#)
[DataBase\(32\)](#)
[DirectX\(16\)](#)
[Extjs\(13\)](#)
[更多](#)

- 随笔档案 (3894)
- [2021年11月\(27\)](#)
[2021年10月\(36\)](#)
[2021年9月\(4\)](#)
[2021年8月\(10\)](#)
[2021年7月\(43\)](#)

Linux 下操作gpio (两种方法 , 驱动和mmap)

目前我所知道的在linux下操作GPIO有两种方法:

1. 编写驱动,这当然要熟悉linux下驱动的编写方法和技巧,在驱动里可以使用ioremap函数获得GPIO物理基地址指针,然后使用这个指针根据ioctl命令进行GPIO寄存器的读写,并把结果回送到应用层。这里提供一点程序片断供大家参考:

```
1.  int  init_module(void){
2.
3.      printk(KERN_ALERT  "ioctl load.\r\n");
4.
5.      register_chrdev(254,"ioreg",&fops);
6.
7.      stb_gpio = (STBX25XX_GPIO_REG *)ioremap(GPIO_BASE,GPIO_LEN);
8.
9.      if(stb_gpio == NULL){
10.
11.          printk(KERN_ALERT  "can't get io base.\r\n");
12.
13.          return -1;
14.
15.      }
16.
17.      return 0;
18.
19.  }
20.
21.  int io_ioctl (struct inode *inode, struct file *filp,unsigned int cmd, unsigned long arg){
22.
23.      unsigned long uGpio;
24.
25.      printk(KERN_ALERT  "io_ioctl cmd=%04x,arg=%04x.\r\n",cmd,(int)arg);
26.
27.      switch(cmd){
28.
29.          case SET_IO_DIR:{
30.
31.              printk(KERN_ALERT  "SET_IO_DIR\r\n");
32.
33.              break;
34.
35.          }
36.
37.          case SET_IO_VALUE:{
38.
39.              printk(KERN_ALERT  "SET_IO_VALUE\r\n");
40.
41.              break;
42.
43.          }
```

2021年6月(1) 2021年5月(29) 2021年4月(15) 2021年3月(13) 2021年2月(96) 2021年1月(47) 2020年12月(2) 2020年11月(27) 2020年10月(44) 2020年9月(14) 更多	
文章分类 (2) SilverLight(1) sql server(1)	
参考博客 linux驱动 回忆未来-向东 Nginx模块开发与原理剖析 大坡3D软件开发 Dean Chen的专栏 Sloan 音视频FFmpeg等 opencv教程 个人开发历程知识库 关注DirectX chenyujing1234 morewindows 雷霄骅(leixiaohua1020)的专栏 ffmpeg参考 webrtc参考— 更多	
阅读排行榜 1. Nginx之location 匹配规则详解(243469) 2. cmake使用方法详解(183804) 3. RTMP、RTSP、HTTP视频协议详解（附：直播流地址、播放软件）(104535) 4. MinGW安装和使用(104336) 5. C语言字符串操作总结大全(超详细)(95672)	
评论排行榜 1. 非IE内核浏览器支持activex插件(38) 2. Nginx之location 匹配规则详解(19) 3. Javascript中定义类(15) 4. C++中的头文件和源文件(9) 5. RTSP协议详解(8)	
推荐排行榜 1. C++中的头文件和源文件(25) 2. Nginx之location 匹配规则详解(23) 3. Javascript中定义类(12) 4. JavaScript中typeof知多少？(11) 5. MinGW安装和使用(9)	
最新评论 1. Re:字符串中的转义字符与字符串的长度 不对打错了是3吧 --英语不过四级不改名 2. Re:字符串中的转义字符与字符串的长度 第三个那个是4吧，strlen（） --英语不过四级不改名 3. Re:Onvif开发之基础介绍篇 大佬，可以转载吗？ --江河湖海日月卿	

```
44.
45.         case GET_IO_VALUE:{
46.
47.             printk(KERN_ALERT  "GET_IO_VALUE\r\n");
48.
49.             uGpio = stb_gpio->GPI;
50.
51.             printk(KERN_ALERT  "GPIO = %08x",(int)uGpio);
52.
53.             copy_to_user((void *)arg,(const void *) &uGpio,sizeof(uGpio));
54.
55.             break;
56.
57.         }
58.
59.         case GET_IO_DIR:{
60.
61.             printk(KERN_ALERT  "GET_IO_DIR\r\n");
62.
63.             break;
64.
65.         }
66.
67.     }
68.
69.     return 0;
70.
71. }
72.
73.
```

2. 在应用层使用mmap函数在应用层获得GPIO物理基地址对应的虚拟地址指针,然后使用这个指针来读写GPIO寄存器,这里提供一点程序片断供大家参考:

```
1. char dev_name[] = "/dev/mem";
2.
3.     GPIO_REGISTER    *gpio_base;
4.
5.     fd = open(dev_name,O_RDWR);
6.
7.     if(fd<0){
8.
9.         printf("open %s is error\n",dev_name);
10.
11.         return -1 ;
12.
13.     }
14.
15.     gpio_base = (GPIO_REGISTER *)mmap( 0, 0x32, PROT_READ | PROT_WRITE, MAP_SHARED,fd, 0x40060000 );
16.
17.     if(gpio_base == NULL){
18.
19.         printf("gpio base mmap is error\n");
20.
21.         close(fd);
22.
23.         return -1;
24.
25.     }
26.
27.     gpio_base->or = (gpio_base->or & 0x7fffffff);
```

Re:非IE内核浏览器支持activex插件
如果想在高版本的谷歌Chrome、Firefox等浏览器中低延迟播放海康威视大华等摄像头的RTSP视频流可以采用猿大师的VLC播放程序，原理是借助猿大师提供的内嵌网页播放的专利技术，底层调用VLC的A...
--猿大师

5. Re:WebSocket兼容到低版本浏览器
亲测第三方websocket框架【GoEasy】的浏览器兼容性不错，对老版本浏览器兼容的很好，用起来稳定性和实时性都不错，还具备自动重连、心跳机制，是一个比较不错的解决方案。地址：...
--Xiqincai

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，删除特定地址区域的对象映射。
基于文件的映射，在mmap和munmap执行过程的任何时刻，被映射文件的st_atime可能被更新。如果st_atime字段在前述的情况下没有得到更新，首次对映射区第一个页索引时会更新该字段的值。用PROT_WRITE 和 MAP_SHARED标志建立起来的文件映射，其st_ctime 和 st_mtime在对映射区写入之后，但在msync()通过MS_SYNC 和 MS_ASYNC两个标志调用之前会被更新。

用法：
#include <sys/mman.h>
void *mmap(void *start, size_t length, int prot, int flags, int fd, off_t offset);
int munmap(void *start, size_t length);

参数：
start：映射区的开始地址。
length：映射区的长度。
prot：期望的内存保护标志，不能与文件的打开模式冲突。是以下的某个值，可以通过or运算合理地组合在一起
PROT_EXEC //页内容可以被执行
PROT_READ //页内容可以被读取
PROT_WRITE //页可以被写入
PROT_NONE //页不可访问
flags：指定映射对象的类型，映射选项和映射页是否可以共享。它的值可以是一个或者多个以下位的组合体
MAP_FIXED //使用指定的映射起始地址，如果由start和len参数指定的内存区重叠于现存的映射空间，重叠部分将会被丢弃。如果指定的起始地址不可用，操作将会失败。并且起始地址必须落在页的边界上。
MAP_SHARED //与其它所有映射这个对象的进程共享映射空间。对共享区的写入，相当于输出到文件。直到msync()或者munmap()被调用，文件实际上不会被更新。
MAP_PRIVATE //建立一个写入时拷贝的私有映射。内存区域的写入不会影响到原文件。这个标志和以上标志是互斥的，只能使用其中一个。
MAP_DENYWRITE //这个标志被忽略。
MAP_EXECUTABLE //同上
MAP_NORESERVE //不要为这个映射保留交换空间。当交换空间被保留，对映射区修改的可能会得到保证。当交换空间不被保留，同时内存不足，对映射区的修改会引起段违例信号。
MAP_LOCKED //锁定映射区的页面，从而防止页面被交换出内存。
MAP_GROWSDOWN //用于堆栈，告诉内核VM系统，映射区可以向下扩展。
MAP_ANONYMOUS //匿名映射，映射区不与任何文件关联。
MAP_ANON //MAP_ANONYMOUS的别称，不再被使用。
MAP_FILE //兼容标志，被忽略。
MAP_32BIT //将映射区放在进程地址空间的低2GB，MAP_FIXED指定时会被忽略。当前这个标志只在x86-64平台上得到支持。
MAP_POPULATE //为文件映射通过预读的方式准备好页表。随后对映射区的访问不会被页违例阻塞。
MAP_NONBLOCK //仅和MAP_POPULATE一起使用时才有意义。不执行预读，只为已存在于内存中的页面建立页表入口。
fd：有效的文件描述词。如果MAP_ANONYMOUS被设定，为了兼容问题，其值应为-1。
offset：被映射对象内容的起点。

返回说明：
成功执行时，mmap()返回被映射区的指针，munmap()返回0。失败时，mmap()返回MAP_FAILED[其值为(void *)-1]，munmap返回-1。errno被设为以下的某个值
EACCES：访问出错
EAGAIN：文件已被锁定，或者太多的内存已被锁定
EBADF：fd不是有效的文件描述词
EINVAL：一个或者多个参数无效
ENFILE：已达到系统对打开文件的限制
ENODEV：指定文件所在的文件系统不支持内存映射
ENOMEM：内存不足，或者进程已超出最大内存映射数量
EPERM：权限不足，操作不允许
ETXTBSY：已写的方式打开文件，同时指定MAP_DENYWRITE标志
SIGSEGV：试着向只读区写入
SIGBUS：试着访问不属于进程的内存区

分类: 嵌入式

好文要顶

关注我

收藏该文

DoubleLi

关注 - 29

粉丝 - 2088

+加关注

« 上一篇: Kernel 中的 GPIO 定义和控制
» 下一篇: linux下判断网络是否连接

posted on 2016-03-25 14:00 DoubleLi 阅读(18733) 评论(0) 编辑 收藏 举报

0

0

推荐

反对

编辑推荐：

- [理解ASP.NET Core - 错误处理\(Handle Errors\)](#)
- [一文分析 Android现状及发展前景](#)
- [Three.js 实现脸书元宇宙 3D 动态 Logo](#)
- [关于研发规范化的一些实践和思考](#)
- [2次心态变化和27个问题：机制落地的部分全貌与节奏控制](#)

最新新闻：

- [科学家开发能实时监测健康数据的超薄设备 可永久粘附在骨骼表面 \(2021-11-23 15:55 \)](#)
 - [Rust程序员一觉醒来都懵了：审核团队集体辞职，发生甚么事了？ \(2021-11-23 15:48 \)](#)
 - [微软赏金太抠门：安全研究人员怒而曝光Windows提权零日漏洞 \(2021-11-23 15:40 \)](#)
 - [小米 屏下摄像头专利公布 采用弹出式设计 \(2021-11-23 15:35 \)](#)
 - [Edge 96稳定版更新：内置Office文件查看器 持续改进PDF体验 \(2021-11-23 15:31 \)](#)
- » [更多新闻...](#)