[**千里之行，始于足下**](http://www.cnblogs.com/wang_yb/)

[**内核工具 – Sparse 简介**](http://www.cnblogs.com/wang_yb/p/3575039.html)

Sparse是内核代码静态分析工具, 能够帮助我们找出代码中的隐患.

**主要内容：**

* Sparse 介绍
* Sparse 使用方法
* Sparse 在编译内核中的使用
* 补充

**1. Sparse 介绍**

Sparse 诞生于 2004 年, 是由linux之父开发的, 目的就是提供一个静态检查代码的工具, 从而减少linux内核的隐患.

其实在Sparse之前, 已经有了一个不错的代码静态检查工具("SWAT"), 只不过这个工具不是免费软件, 使用上有一些限制.

所以 linus 还是自己开发了一个静态检查工具.

具体可以参考这篇文章(2004年的文章了): [Finding kernel problems automatically](http://lwn.net/Articles/87538/)

Sparse相关的资料非常少, 关于它的使用方法我也是网上查找+自己实验得出来的.

内核代码中还有一个简略的关于 Sparse的说明文件: Documentation/sparse.txt

Sparse通过 gcc 的扩展属性 \_\_attribute\_\_ 以及自己定义的 \_\_context\_\_ 来对代码进行静态检查.

这些属性如下(尽量整理的,可能还有些不全的地方):

|  |  |  |
| --- | --- | --- |
| **宏名称** | **宏定义** | **检查点** |
| \_\_bitwise | \_\_attribute\_\_((bitwise)) | 确保变量是相同的位方式(比如 bit-endian, little-endiandeng) |
| \_\_user | \_\_attribute\_\_((noderef, address\_space(1))) | 指针地址必须在用户地址空间 |
| \_\_kernel | \_\_attribute\_\_((noderef, address\_space(0))) | 指针地址必须在内核地址空间 |
| \_\_iomem | \_\_attribute\_\_((noderef, address\_space(2))) | 指针地址必须在设备地址空间 |
| \_\_safe | \_\_attribute\_\_((safe)) | 变量可以为空 |
| \_\_force | \_\_attribute\_\_((force)) | 变量可以进行强制转换 |
| \_\_nocast | \_\_attribute\_\_((nocast)) | 参数类型与实际参数类型必须一致 |
| \_\_acquires(x) | \_\_attribute\_\_((context(x, 0, 1))) | 参数x 在执行前引用计数必须是0,执行后,引用计数必须为1 |
| \_\_releases(x) | \_\_attribute\_\_((context(x, 1, 0))) | 与 \_\_acquires(x) 相反 |
| \_\_acquire(x) | \_\_context\_\_(x, 1) | 参数x 的引用计数 + 1 |
| \_\_release(x) | \_\_context\_\_(x, -1) | 与 \_\_acquire(x) 相反 |
| \_\_cond\_lock(x,c) | ((c) ? ({ \_\_acquire(x); 1; }) : 0) | 参数c 不为0时,引用计数 + 1, 并返回1 |

其中 \_\_acquires(x) 和 \_\_releases(x), \_\_acquire(x) 和 \_\_release(x) 必须配对使用, 否则 Sparse 会给出警告

**注**： 在Fedora系统中通过 rpm 安装的 sparse 存在一个小bug.

即使用时会报出 error: unable to open ’stddef.h’ 的错误, 最好从自己源码编译安装 sparse.

参考: <http://wangcong.org/blog/archives/504>

**2. Sparse 使用方法**

**2.1 \_\_bitwise 的使用**

主要作用就是确保内核使用的整数是在同样的位方式下.

在内核代码根目录下 grep -r '\_\_bitwise', 会发现内核代码中很多地方都使用了这个宏.

对于使用了这个宏的变量, Sparse 会检查这个变量是否一直在同一种位方式(big-endian, little-endian或其他)下被使用,

如果此变量在多个位方式下被使用了, Sparse 会给出警告.

内核代码中的例子:

/\* 内核版本:v2.6.32.61 file:include/sound/core.h 51行 \*/

typedef int \_\_bitwise snd\_device\_type\_t;

**2.2 \_\_user 的使用**

如果使用了 \_\_user 宏的指针不在用户地址空间初始化, 或者指向内核地址空间, 设备地址空间等等, Sparse会给出警告.

内核代码中的例子:

/\* 内核版本:v2.6.32.61 file:arch/score/kernel/signal.c 45行 \*/

static int setup\_sigcontext(struct pt\_regs \*regs, struct sigcontext \_\_user \*sc)

**2.3 \_\_kernel 的使用**

如果使用了 \_\_kernel 宏的指针不在内核地址空间初始化, 或者指向用户地址空间, 设备地址空间等等, Sparse会给出警告.

内核代码中的例子:

/\* 内核版本:v2.6.32.61 file:arch/s390/lib/uaccess\_pt.c 180行 \*/

memcpy(to, (void \_\_kernel \_\_force \*) from, n);

**2.4 \_\_iomem 的使用**

如果使用了 \_\_iomem 宏的指针不在设备地址空间初始化, 或者指向用户地址空间, 内核地址空间等等, Sparse会给出警告.

内核代码中的例子:

/\* 内核版本:v2.6.32.61 file:arch/microblaze/include/asm/io.h 22行 \*/

static inline unsigned char \_\_raw\_readb(const volatile void \_\_iomem \*addr)

**2.5 \_\_safe 的使用**

使用了 \_\_safe修饰的变量在使用前没有判断它是否为空(null), Sparse会给出警告.

我参考的内核版本(v2.6.32.61) 中的所有内核代码都没有使用 \_\_safe, 估计可能是由于随着gcc版本的更新,

gcc已经会对这种情况给出警告, 所以没有必要用Sparse去检查了.

**2.6 \_\_force 的使用**

使用了\_\_force修饰的变量可以进行强制类型转换, 没有使用 \_\_force修饰的变量进行强制类型转换时, Sparse会给出警告.

内核代码中的例子:

/\* 内核版本:v2.6.32.61 file:arch/s390/lib/uaccess\_pt.c 180行 \*/

memcpy(to, (void \_\_kernel \_\_force \*) from, n);

**2.7 \_\_nocast 的使用**

使用了\_\_nocast修饰的参数的类型必须和实际传入的参数类型一致才行，否则Sparse会给出警告.

内核代码中的例子:

/\* 内核版本:v2.6.32.61 file:fs/xfs/support/ktrace.c 55行 \*/

ktrace\_alloc(int nentries, unsigned int \_\_nocast sleep)

**2.8 \_\_acquires \_\_releases \_\_acquire \_\_release的使用**

这4个宏都是和锁有关的, \_\_acquires 和 \_\_releases 必须成对使用, \_\_acquire 和 \_\_release 必须成对使用, 否则Sparse会给出警告.

**2.9 \_\_cond\_lock 的使用**

这个宏有点特别, 因为没有 \_\_cond\_unlock 之类的宏和它对应.

之所以有这个宏的原因可以参见: <http://yarchive.net/comp/linux/sparse.html> 最后一段.

这个宏的来源清楚了, 但是为什么这个宏里面还要调用一次 \_\_acquire(x)? 我也不是很清楚, 在网上找了好久也没找到, **谁能指教的话非常感谢!!!**

**3. Sparse 在编译内核中的使用**

用 Sparse 对内核进行静态分析非常简单.

# 检查所有内核代码

make C=1 检查所有重新编译的代码

make C=2 检查所有代码, 不管是不是被重新编译

**4. 补充**

Sparse除了能够用在内核代码的静态分析上, 其实也可以用在一般的C语言程序中.

比如下面的小例子:

[复制代码](javascript:void(0);)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @file : sparse\_test.c

\* @author : wangyubin

\* @date : Fri Feb 28 16:33:34 2014

\*

\* @brief : 测试 sparse 的各个检查点

\* history : init

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define \_\_acquire(x) \_\_context\_\_(x,1)

#define \_\_release(x) \_\_context\_\_(x,-1)

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

{

int lock = 1;

\_\_acquire(lock);

/\* TODO something \*/

\_\_release(lock); /\* 注释掉这一句 sparse 就会报错 \*/

return 0;

}

[复制代码](javascript:void(0);)

如果安装了 Sparse, 执行静态检查的命令如下:

$ sparse -a sparse\_test.c

sparse\_test.c:15:5: warning: context imbalance in 'main' - wrong count at exit

Sparse相关资料可以参考wiki: [Sparse wiki](http://en.wikipedia.org/wiki/Sparse)

标签: [linux-kernel](http://www.cnblogs.com/wang_yb/tag/linux-kernel/)

[**好文要顶**](javascript:void(0);) [**关注我**](javascript:void(0);) [**收藏该文**](javascript:void(0);)**[http://common.cnblogs.com/images/icon_weibo_24.png](javascript:void(0);)** **[http://common.cnblogs.com/images/wechat.png](javascript:void(0);)**

[http://pic.cnblogs.com/face/83005/20140701082644.png](http://home.cnblogs.com/u/wang_yb/)

[wang\_yb](http://home.cnblogs.com/u/wang_yb/)  
[关注 - 7](http://home.cnblogs.com/u/wang_yb/followees)  
[粉丝 - 204](http://home.cnblogs.com/u/wang_yb/followers)

[+加关注](javascript:void(0);)

1

0

[«](http://www.cnblogs.com/wang_yb/p/3532349.html)上一篇：[Linux 内核Coding Style整理](http://www.cnblogs.com/wang_yb/p/3532349.html)  
[»](http://www.cnblogs.com/wang_yb/p/3699410.html)下一篇：[fedora 配置使用点滴](http://www.cnblogs.com/wang_yb/p/3699410.html)

posted @ 2014-03-01 14:18 [wang\_yb](http://www.cnblogs.com/wang_yb/) 阅读(3772) 评论(0) [编辑](https://i.cnblogs.com/EditPosts.aspx?postid=3575039) [收藏](http://www.cnblogs.com/wang_yb/p/3575039.html)

[刷新评论](javascript:void(0);)[刷新页面](http://www.cnblogs.com/wang_yb/p/3575039.html)[返回顶部](http://www.cnblogs.com/wang_yb/p/3575039.html#top)

**注册用户登录后才能发表评论，请**[**登录**](javascript:void(0);)**或**[**注册**](javascript:void(0);)**，**[**访问**](http://www.cnblogs.com/)**网站首页。**

[【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库](http://www.ucancode.com/index.htm)  
[【链接】融云阿里云大咖面对面 ：教你玩转直播风口](http://www.rongcloud.cn/active/live)  
[【推荐】报表开发有捷径：快速设计轻松集成，数据可视化和交互](http://www.gcpowertools.com.cn/products/activereports_overview.htm?utm_source=cnblogs&utm_medium=blogpage&utm_term=bottom&utm_content=AR&utm_campaign=community)  
[【推荐】一个月仅用630元赚取15000元，学会投资](http://cn.iforex.com/landing_pages/br1/zhns/Testimonials8/?content=Testimonials8&SID=445058)  
[【推荐】网易云信-一天开发一个微信，独创1对1技术顾问让开发加速](http://netease.im/?&hmsr=bky8&from=bky8)

[](http://gold.xitu.io/welcome/?utm_source=cnblogs&utm_medium=banner&utm_content=gaoshou&utm_campaign=q3_website)

**最新IT新闻**:  
· [陌陌布局视频社交：继直播之后再推限时视频分享“时刻”](http://news.cnblogs.com/n/551969/)  
· [滴滴进军在线租车领域 可免费送车上门](http://news.cnblogs.com/n/551968/)  
· [徐小平：我不满足，我很焦虑](http://news.cnblogs.com/n/551967/)  
· [NASA发布新360度图片 跟好奇号一起探索火星](http://news.cnblogs.com/n/551966/)  
· [Linux kernel 4.8-rc3发布下载](http://news.cnblogs.com/n/551965/)  
» [更多新闻...](http://news.cnblogs.com/)

[http://images2015.cnblogs.com/news/24442/201608/24442-20160803181056012-1919505484.jpg](https://www.jiguang.cn/devservice/?from=cnblogs01)

**最新知识库文章**:

· [程序猿媳妇儿注意事项](http://kb.cnblogs.com/page/550625/)  
· [可是姑娘，你为什么要编程呢？](http://kb.cnblogs.com/page/540529/)  
· [知其所以然（以算法学习为例）](http://kb.cnblogs.com/page/549631/)  
· [如何给变量取个简短且无歧义的名字](http://kb.cnblogs.com/page/548394/)  
· [编程的智慧](http://kb.cnblogs.com/page/549080/)

» [更多知识库文章...](http://kb.cnblogs.com/)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | [<](javascript:void(0);) | 2016年8月 | [>](javascript:void(0);) | | | | | | | |
| **日** | **一** | **二** | **三** | **四** | **五** | **六** |
| 31 | 1 | 2 | 3 | 4 | [5](http://www.cnblogs.com/wang_yb/archive/2016/08/05.html) | 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 | 31 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |

**搜索**



**常用链接**

* [我的随笔](http://www.cnblogs.com/wang_yb/p/)
* [我的评论](http://www.cnblogs.com/wang_yb/MyComments.html)
* [我的参与](http://www.cnblogs.com/wang_yb/OtherPosts.html)
* [最新评论](http://www.cnblogs.com/wang_yb/RecentComments.html)
* [我的标签](http://www.cnblogs.com/wang_yb/tag/)
* [更多链接](http://www.cnblogs.com/wang_yb/p/3575039.html)

**我的标签**

* [linux](http://www.cnblogs.com/wang_yb/tag/linux/)(63)
* [linux-kernel](http://www.cnblogs.com/wang_yb/tag/linux-kernel/)(30)
* [CLR via C#笔记](http://www.cnblogs.com/wang_yb/tag/CLR%20via%20C%23%E7%AC%94%E8%AE%B0/)(24)
* [C#学习](http://www.cnblogs.com/wang_yb/tag/C%23%E5%AD%A6%E4%B9%A0/)(9)
* [puppet](http://www.cnblogs.com/wang_yb/tag/puppet/)(8)
* [elixir](http://www.cnblogs.com/wang_yb/tag/elixir/)(8)
* [C#疑难杂症](http://www.cnblogs.com/wang_yb/tag/C%23%E7%96%91%E9%9A%BE%E6%9D%82%E7%97%87/)(7)
* [javascript学习](http://www.cnblogs.com/wang_yb/tag/javascript%E5%AD%A6%E4%B9%A0/)(6)
* [golang](http://www.cnblogs.com/wang_yb/tag/golang/)(4)
* [C学习](http://www.cnblogs.com/wang_yb/tag/C%E5%AD%A6%E4%B9%A0/)(4)
* [更多](http://www.cnblogs.com/wang_yb/tag/)

**随笔档案**(180)

* [2016年8月 (1)](http://www.cnblogs.com/wang_yb/archive/2016/08.html)
* [2016年7月 (1)](http://www.cnblogs.com/wang_yb/archive/2016/07.html)
* [2016年6月 (3)](http://www.cnblogs.com/wang_yb/archive/2016/06.html)
* [2016年5月 (3)](http://www.cnblogs.com/wang_yb/archive/2016/05.html)
* [2016年4月 (1)](http://www.cnblogs.com/wang_yb/archive/2016/04.html)
* [2016年3月 (1)](http://www.cnblogs.com/wang_yb/archive/2016/03.html)
* [2016年2月 (1)](http://www.cnblogs.com/wang_yb/archive/2016/02.html)
* [2015年10月 (2)](http://www.cnblogs.com/wang_yb/archive/2015/10.html)
* [2015年3月 (2)](http://www.cnblogs.com/wang_yb/archive/2015/03.html)
* [2015年2月 (2)](http://www.cnblogs.com/wang_yb/archive/2015/02.html)
* [2015年1月 (7)](http://www.cnblogs.com/wang_yb/archive/2015/01.html)
* [2014年11月 (1)](http://www.cnblogs.com/wang_yb/archive/2014/11.html)
* [2014年10月 (1)](http://www.cnblogs.com/wang_yb/archive/2014/10.html)
* [2014年9月 (4)](http://www.cnblogs.com/wang_yb/archive/2014/09.html)
* [2014年8月 (3)](http://www.cnblogs.com/wang_yb/archive/2014/08.html)
* [2014年7月 (11)](http://www.cnblogs.com/wang_yb/archive/2014/07.html)
* [2014年6月 (40)](http://www.cnblogs.com/wang_yb/archive/2014/06.html)
* [2014年5月 (7)](http://www.cnblogs.com/wang_yb/archive/2014/05.html)
* [2014年4月 (1)](http://www.cnblogs.com/wang_yb/archive/2014/04.html)
* [2014年3月 (1)](http://www.cnblogs.com/wang_yb/archive/2014/03.html)
* [2014年1月 (6)](http://www.cnblogs.com/wang_yb/archive/2014/01.html)
* [2013年12月 (1)](http://www.cnblogs.com/wang_yb/archive/2013/12.html)
* [2013年11月 (1)](http://www.cnblogs.com/wang_yb/archive/2013/11.html)
* [2013年10月 (1)](http://www.cnblogs.com/wang_yb/archive/2013/10.html)
* [2013年9月 (1)](http://www.cnblogs.com/wang_yb/archive/2013/09.html)
* [2013年6月 (1)](http://www.cnblogs.com/wang_yb/archive/2013/06.html)
* [2013年5月 (3)](http://www.cnblogs.com/wang_yb/archive/2013/05.html)
* [2013年4月 (6)](http://www.cnblogs.com/wang_yb/archive/2013/04.html)
* [2012年9月 (2)](http://www.cnblogs.com/wang_yb/archive/2012/09.html)
* [2012年8月 (4)](http://www.cnblogs.com/wang_yb/archive/2012/08.html)
* [2012年7月 (3)](http://www.cnblogs.com/wang_yb/archive/2012/07.html)
* [2012年6月 (1)](http://www.cnblogs.com/wang_yb/archive/2012/06.html)
* [2012年1月 (1)](http://www.cnblogs.com/wang_yb/archive/2012/01.html)
* [2011年12月 (2)](http://www.cnblogs.com/wang_yb/archive/2011/12.html)
* [2011年11月 (4)](http://www.cnblogs.com/wang_yb/archive/2011/11.html)
* [2011年10月 (2)](http://www.cnblogs.com/wang_yb/archive/2011/10.html)
* [2011年9月 (1)](http://www.cnblogs.com/wang_yb/archive/2011/09.html)
* [2011年8月 (2)](http://www.cnblogs.com/wang_yb/archive/2011/08.html)
* [2011年7月 (9)](http://www.cnblogs.com/wang_yb/archive/2011/07.html)
* [2011年6月 (5)](http://www.cnblogs.com/wang_yb/archive/2011/06.html)
* [2011年5月 (3)](http://www.cnblogs.com/wang_yb/archive/2011/05.html)
* [2011年4月 (9)](http://www.cnblogs.com/wang_yb/archive/2011/04.html)
* [2010年6月 (7)](http://www.cnblogs.com/wang_yb/archive/2010/06.html)
* [2010年5月 (9)](http://www.cnblogs.com/wang_yb/archive/2010/05.html)
* [2010年4月 (3)](http://www.cnblogs.com/wang_yb/archive/2010/04.html)

**积分与排名**

* 积分 - 163252
* 排名 - 1122

**最新评论**

* [1. Re:GIT的使用](http://www.cnblogs.com/wang_yb/p/3867221.html#3490324)
* @痴心的汪汪本地仓库的url就是 仓库所在的路径： 比如用户 abc 要自已的用户目录（/home/abc）下 创建了一个仓库 xyz.git，那么这个仓库的路径就是 （~/xyz.git 或者......
* --wang\_yb
* [2. Re:GIT的使用](http://www.cnblogs.com/wang_yb/p/3867221.html#3490318)
* git的本地仓库的url怎么得到啊？在线等，拜托拜托
* --痴心的汪汪
* [3. Re:Makefile 使用总结](http://www.cnblogs.com/wang_yb/p/3990952.html#3488553)
* 简单明了
* --亚煞极
* [4. Re:《Linux内核设计与实现》读书笔记（十五）- 进程地址空间(kernel 2.6.32.60)](http://www.cnblogs.com/wang_yb/p/3351599.html#3485475)
* 三点补充说明很赞，解决了我看书时候留下的疑惑
* --hbzhouce
* [5. Re:《Linux内核设计与实现》读书笔记（十二）- 内存管理](http://www.cnblogs.com/wang_yb/archive/2013/05/23/3095907.html#3407619)
* @wang\_yb这里说的不分页是指内核所在的page不会被换出去，同2.4.3 小节说的是一回事。书中有些地方翻译得稍显生硬，还有些笔误什么的。...
* --puppypyb

**阅读排行榜**

* [1. C#操作Excel总结(49731)](http://www.cnblogs.com/wang_yb/archive/2010/06/03/1750419.html)
* [2. Makefile 使用总结(39204)](http://www.cnblogs.com/wang_yb/p/3990952.html)
* [3. GIT的使用(35450)](http://www.cnblogs.com/wang_yb/p/3867221.html)
* [4. IIS 403错误详细原因(21544)](http://www.cnblogs.com/wang_yb/archive/2010/05/10/1732093.html)
* [5. C# 连接Oracle（利用ODP.net，不安装oracle客户端）(20556)](http://www.cnblogs.com/wang_yb/archive/2011/07/11/2103368.html)

**评论排行榜**

* [1. emacs org-mode文件转html文件(22)](http://www.cnblogs.com/wang_yb/p/3519221.html)
* [2. C# 连接Oracle（利用ODP.net，不安装oracle客户端）(15)](http://www.cnblogs.com/wang_yb/archive/2011/07/11/2103368.html)
* [3. 《Linux内核设计与实现》读书笔记（十二）- 内存管理(15)](http://www.cnblogs.com/wang_yb/archive/2013/05/23/3095907.html)
* [4. Petapoco 连接oracle11g 自动生成poco时遇到的问题(12)](http://www.cnblogs.com/wang_yb/archive/2011/07/12/2104788.html)
* [5. Makefile 使用总结(12)](http://www.cnblogs.com/wang_yb/p/3990952.html)

**Copyright ©2016 wang\_yb**