



fengtao612的专栏

RSS订阅

CODING



个人资料



fengtao612

关注

原创
14粉丝
22喜欢
1评论
7

等级： 博客 4

访问量：8万+

积分：1018

排名：5万+



0



收藏



评论



微信



微博



QQ


0

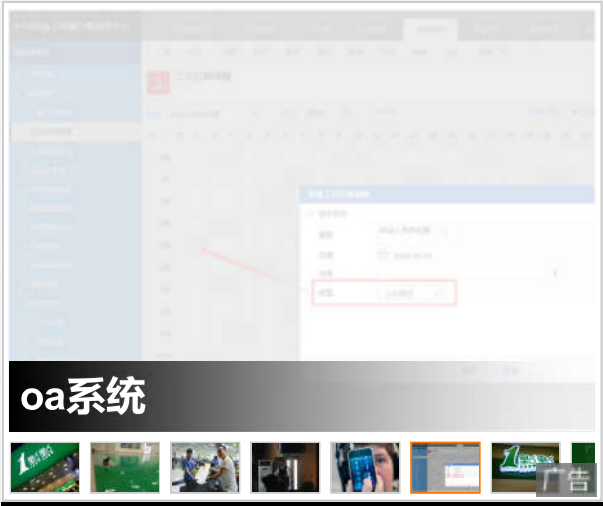

收藏


评论


微信


微博


QQ



最新文章

linux下 .bz2文件的解压缩（为了安装 insigh
t-6.1a.tar.bz2 ）

ARM_LIB_STACKHEAP关键词

如何在单片机上使用printf函数（ printf ）（ a
vr ）（ stm ）（ lpc ）（ 单片机 ）

指针的应用（ zlg ）

函数指针转换

个人分类

ARM	13篇
编程	7篇
通信	1篇


0


收藏


评论


微信


微博


QQ

归档

2012年3月	2篇
2011年8月	1篇
2011年6月	2篇
2011年2月	1篇
2011年1月	1篇

展开

热门文章

- 有关分散加载文件scatter的理解 关键是加载域 加载地址与执行域地址的 处理
阅读量：7275
- 如何在单片机上使用printf函数（ printf ）（ a
vr ）（ stm ）(lpc)（ 单片机 ）
阅读量：7018
- DSP CMD文件 连接器命令文件
阅读量：5457
- C2000系CMD文件的配置理解
阅读量：4975
- JTAG 接口
阅读量：4691
- 2812上电引导过程
阅读量：4634
- 关于F2812片内Flash烧写的问题
阅读量：4242



0



收藏



评论



微信



微博



QQ

NANDFLASH调试 (二)

阅读量 : 3942

ARM启动方式 特别是关于中断向量的讲解
解决了我的迷惑 reset vector的重定向

阅读量 : 3294

void EXTI_Configuration (void)

阅读量 : 2801

最新评论

#ifdef __cplusplus...

u010613585 : 不错, 很有用

DSP CMD文件 连接器命令文件

u010613585 : Good。

2812上电引导过程

zhouhangyuan : 学习了。谢谢分享！

(*(volatile unsig...

davee007 : 是否可以从强制类型转换的角度去看

void EXTI_Configu...

qinliu88 : [e01]

DSP CMD文件 连接器命令文件

匿名用户 : 谢谢, 很有用啊！

C2000系CMD文件的配置理解

fengtao612 : DSP的存储器的地址范围, CMD是
主要是根据那个来编的。CMD 它是用来分配ROM
和RAM空间...



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

关于 · 招聘 · 广告服务 · 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心



有关分散加载文件scatter的理解 关键是加载域 加载地址与执行域地址的 处理

2010年10月20日 22:34:00

scatter基本点：

1. 编译后输出的映像文件中各段是首尾相连的，中间没有空闲的区域，它们的先后关系是根据链接时参数的先后次序决定的 armlinker -file1.o file2.o

2. scatter用于将编译后的映像文件中的特定段加载到多个分散的指定内存区域

3. 有2类域region：执行域（execution region，一般是ram区域）和加载域(load region，一般是rom区域)

4. 加载域：就是编译之后得到的二进制文件烧写到rom中的这一段区域，所有的代码RO、预定义变量RW、堆栈之类清不清空无关紧要的大片内存区域ZI都包括在其中

5. 执行域：就是把映像进行‘解压缩’后的样子。比如：RO没有变动还是在ROM中，RW被移到了SRAM中，而ZI被放置在SDRAM中

6. scatter本身并不像实现‘解压缩’，编译器读入scatter文件之后会根据其中的各种地址生成启动代码，实现对映像的加载，而这一段代码就是boot

Sections)它是__main部分。这就是在汇编启动代码的最后跳转到__main() 而不是跳向main()的原因之一。 7. 起始地址与加载域重合的执行域成为root region，*(InRoot

Sections)必须放在执行域中，否则链接的时候会报错。*(+RO)包含了*(InRoot

Sections)，

scatter语法：

ROM_LOAD 0x00000000

{

ROM 0x00000000 0x003FFFFFFF

{

vectors.o (+RO,+FIRST)

```

* (InRoot$$Sections)      ; All library sections that must be in a root region
*(+RO)
}

```

```

SRAM 0x00400000 0x003FFFFF
{
    * (+RW,+ZI)
}

```

```

SDRAM1 0x41C00000 UNINIT

```

```

{
    stack.o (+ZI)
}

```

stack.s中定义了top_of_stack为长度为1的space，指定栈顶地址

```

SDRAM2 +0 UNINIT
{
    heap.o (+ZI)
}

```

```

}

```

注解：

1. ROM_LOAD是加载域，这里只有一个，也可以有多个（rom地址不连续的情况）
2. ROM、SRAM、SDRAM1、SDRAM2是执行域，有多个。第一个执行域必须和加载域地址重合，因为ARM的复位地址就是加载域的起始地址（有bootstrap的话加载域址就是bootstrap执行完后的跳转地址）
3. vectors.o (+RO, +FIRST) 中断向量表放在最开头
4. ROM 0x00000000 0x003FFFFF; 加载域名 起始地址 最大允许长度；‘最大允许长度’也可以省略，但缺点是编译器不会检查段是否溢出和别的段重叠了。‘起始地址’ = +0表示紧接着上一段开始的连续地址。

5. * (InRoot\$\$Sections)是复制代码的代码

6. UNINT关键字表示不进行初始化清零

值得注意的是：在一个scatter文件中，同一个.o文件不能出现2次，即使是在2个不同的加载域中也不可以，否则会报错：Ambiguous selectors found for *.o，错误的例子：

```
LOAD1 0x00000000
```

```
{  
  EXE1  
  {  
    Init.o  
  }  
}
```



0



收藏

```
LOAD2 0xFFFF0000
```

```
{  
  EXE2  
  {  
    Init.o  
  }  
}
```



评论



微信



微博



QQ

想起了中学里哲学老师让解释为什么人不能两次踏入同一条河流，当年稀里糊涂的写的答案，老师批了个大差，回去有没有补上，今天居然在这里遇到了老问题。。。推测是编译器自动生成的scatter载入代码InRoot\$\$Sections不支持把同一obj搬移2次。

这就带来一个问题：如果希望把同一段代码（如中断跳转表）载入2份拷贝到不同的地址，咋整？一个笨办法是自己写一段代码搬移程序来代替编译器自动生成的搬移代码，但前提是需要搞懂映像文件的组织，增加了工作量。投机一点的方法是在makefile中把一个.o文件复制并重新起一个名字，然后把它们传递给 armlink。另外，猜测scatter语法可能包含诸如+duplicate之类的关键字来允许同一段的多个副本

分散加载的实现：

由于现在的嵌入式技术发展比较快，各类存储器也层出不穷，但是它们在容量、成本和速度上有所差异，嵌入式系统又对成本比较敏感，那么合理的选择存储器和充分的利用存储器资源成为一个必要解决的问题。咱们工程师最喜欢的就是发掘问题，然后解决问题，基于嵌入式系统对存储敏感，那么要合理的利用存储器资源，就必须找到一种合理的方式。工程师们发现，可以把运行的程序放在不同成本的存储器中，这个成本的支点，比如把没有运行的但是较为庞大的程序放在容量大、成本低、速度也较低的FLASH存储器中，要用的时候再读。但是，这里面又有一个问题，嵌入式本身就对信号的处理速度有较高的要求，这点在实时操作系统的定义上有所体现。所以那些要用的程序段如果要保证其高速的运行那么就得放在一个在高速的存储器中，不过这是有代价的：较高成本，小容量。但是，相信技术的发展这个问题终将被解决，到时候寻找平衡点的问题也就不存在了。好了，说了多了点。切入正题。

程序总有两种：运行态和静止态。当系统掉电的时候程序需要被保存在非易失性的存储器中，且这个时候程序的排放是按照地址依次放的，换了我才懒得管它怎么放，只要不掉就行。当系统上电后，CPU就要跑起来了，CPU属于高速器件，存储器总是不怎么能跟得上，跟不上那么我们就尽量缩短它们之间的差距，那留下一条路，那就是尽量提高存储器的读取速度，存储器类型决定其速度的水平，量放在速度高的存储器就成为首选解决方案。那么我们就把要执行的程序暂时拿到速度较快的RAM中。那么拿的过程就牵涉到程序加载了。这就是要解决的问题。

一个映像文件 (region)、输出段(output sections)和输入段 (input sections) 组成。不要想得太复杂，其实他们之间就是包含与被包含的。具体关系如下：

映像文件>域>输出段>输入段

输入段：输入段就是我们写的应用程序代码+初始化的数据+应该被初始化为0的数据+没有初始化的数据。用英文表示就是:RO(READ ONLY),RW(READWRITE),ZI(ZERO Initialized), NOINIT(Not Initialized).ARM连接器根据各个输入段不同的属性把相同的再一起组合一下就成了输出段。（注 有关链接器的知识 补充。。。。。。。）

输出端：为了简化编译过程和更容易取得各种段的地址，那么把多个同属性的输入段按照一定的规律组合在一起，当然这个输出段的属性就和它包含的输入段的属性一样。输入段的排放规律就是：最先排放RO属性的输入段，然后是RW属性段，最后是ZI 或NOINIT

域：为什么还要加一层域，我的理解是由于代码的功能不同，那么我们有必要把不同功能的代码分类放。我们可以把需要高速执行的代码放在一起、把执行效率高的放在一起 低的放在一起。

```

LOAD_ROM1  C 00000  ;从火车上取出来时的地址（如：成都站）
{
    EXEC_ROM 0x40000000
    {
        PRO 1.O ( +RO ) ;把品牌RO的货物发给0x40000000去
        RAM 0x80000000
        {
            PRO 1.O(+RW,+ZI) ;把品牌RW,ZI的货物依次发给0x80000000
        }
    }
    .....
}

```

其他的段也可以这样依葫芦画瓢。

scatter的原理就介绍这样，其中的语法和规则要多写多把代码的地址拖出来看才能体会。不过都是很简单的，生活中的小常识就能解决这些问题。为什么？因为设计这些规则的工程师的灵感就是源自生活。嘿嘿...享受把代码随处放的乐趣吧，...enjoy...



0



收藏

文章标签：

存储

编



评论

嵌入式

makefile

exe

library



想对作者说点什么



微信



微博

轻松应战PostgreSQL & Greenplum非法/歧义字符加载入库

1. 内核字符集转换实现 2. 加载时常见问题 3. 非法字符库外工具、库内C语言容错的两种解决方案 4. 多分割符外部表自定义C语言扩展解决数据歧义 5. 如何从其它数据库导出数据，将问题消灭在源头


学院 2017年04月25日 20:39

keil mini2440 分散加载文件scatter中(InRoot\$\$Sections)的理解

查阅一些资料终于认识了 (InRoot

Sections)实现对映像的加载，而这一段代码就是 * (InRoot

Sections) 它是__main()的一部分。从启动代码说起 启动代码 1...

 weixiao2015 2016-05-02 12:59:45 阅读数：1573

对于程序员来说，英语到底多重要

不背单词和语法，一个公式学好英语

广告



0

编译提示warni 6314W: No section matches pattern *(InRoot\$\$Sections).解决方法

禁用所有具有指定标识符的消息。此选项需要以逗号分隔的诊断消息编号列表，用于指定要禁止显示的消息。例如，要禁止显示编号为L6314W和L6305W的警告消息，请使用命令：armlink-diag...

 li1367356 2017-07-19 19:38:18 阅读数：1945

Scatter文件的分析

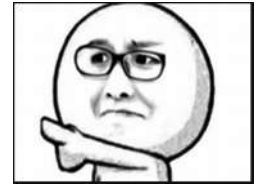
今天拿了被同事扔一边没看的ARM培训资料翻阅，读至scatter一节，发现写得甚是精辟。之前看的很多国人写得文章，未免有简单问题复杂化之嫌。而ARM的RVCT手册又偏冗长，让人立刻看到重点。今归纳如下...

 caz28 2012-06-01 10:30:15 阅读数：776

ARM启动方式 特别是关于中断向量的讲解解决了我的迷惑 reset vector的重定向

严格的说，ARM内核真正支持的启动方式只有一个：通过操作CP15来实现对reset vector的重定向，比如：硬件复位时指向ROM区域0xffff0000，而在启动代码中再重定向到R...

 fengtao612 2010-10-20 21:39:00 阅读数：3294



ARM_LIB_HEAP 和 ARM_LIB_STACK 分散装入描述

FLASH_LOAD 0x0000 0x00200000 { VECTORS +0 0x400 { * (:gdef:__vectab_stack_and_reset, +FIRST) ; ...

JRENSKY 2015-02-27 12:10:11 阅读数 : 1516

Unity3D程序加密，可有效防止反编译

无需手动加密Assembly.DLL代码，自动编译mono，防止反编译



0



收藏



评论

S3C2440开发板与Keilview MDK4.22之库的使用

一。与c库会强制链接。如果你写了一个c程序，必然会和c库链接，尽管你没有直接使用c库函数。这是因为编译器为了改进程序，可能隐含的产生了对c库函数调用。即便你的程序没有调用main()函数，也只是说c...

sam1430 2016-07-16 16:26:17 阅读数 : 2867

ARM 必须知道



微信

寄存器和工作模式：ARM模式： fiq/irq/abt/und/sys/usr/svc。通过"MSR cpsr_c,#0xdx"切换。上电时进入svc模式。 svc和usr的区别是：sv...

xjbclz 2016-07-09 09:15 阅读数 : 1667



微博

ARM_LIB_STACK 和 ARM_LIB_HEAP 关键词



QQ

__user_initial_stackheap 函数翻译(转) from: <http://blog.cechina.cn/metalmax/181841/message.aspx> 今...

fengtao612 2012-03-05 17:12:56 阅读数 : 2157

转载_ARM的分散加载文件(scatter)介绍

在keil中编译的程序通过了，但是debug的时候会出现一些错误： *** error 65: access violation at 0x4C000018 : no 'write' permi...

 williamwanglei 2012-11-26 16:12:28 阅读数：7832

对ARM加载域和运行域的理解

一般而言，一个程序包括只读的代码段和可读写的数据段。在ARM的集成开发环境中，只读的代码段和常量被称作RO段(ReadOnly)；可读写的全局变量和静态变量被称作RW段(ReadWrite)；RW段中...

 Fybon 2014-01-12 19:34:15 阅读数：1147

对ARM加载域和运行域的理解（ARM程序是怎么运行的）

对ARM加载域和运行域的理解 一般而言，一个程序包括只读的代码段和可读写的数据段。在ARM的集成开发环境中，只读的代码段和常量被称作RO段(ReadOnly)；可读写的全局变量和静态变量被称作RW段...

 lindabell 2013-04-23 23:04:12 阅读数：6397


试图搞懂MDK下载到Flash(一)--Nand Flash启动流程，加载域和运行域

NAND FLASH启动流程 这里我先以TQ2440的启动代码分析，因为手上有本书，反正Nand Flash启动流程都是一样的对于mini2440和TQ2440来说。TQ2440的启动代码...

 niepangu 2014-01-11 11:02:51 阅读数：595

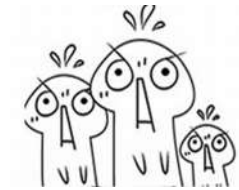
分散加载文件及分区应用（重点必看）

分散加载能够将加载域和运行域存储在存储器中的代码和数据描述在被称为分散加载描述文件的一个文本描述文件中，以供连接时使用。（1）分散加载区分散加载区域分为两类：• 加载域，含应用程序复位和加载时的代码和数据。...

 zsfqiuyigui 2011-03-24 23:33:00 阅读数：2340

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



armlink之分散加载文件和 __attribute__((section()))

背景： 因为平时自己有空也在研究uboot的代码，发现在uboot中shell的命令体系很与众不同，因为自己平时理解的命令体系大概分为两种，一种是将shell命令定义在一段数组中，另一种则是用链...

sinat_33611142



2016-10-08 22:29:22

阅读数：1119

0

Espresso 自动测试（九） - inRoot 使用

在日常的测试中，我们经常会碰到许许多多的测试点，如Toast内容的验证，AutoCompleteText的选择等。这些测试项都有一个共同的特点。即不在主UI布局的结构(layout,及其...

qq744746842



评论

2016-02-16 09:17:57

阅读数：1364

Xcode及模拟器下载

如果你嫌在 App Store 上下载 Xcode 太慢，你也可以选择从网络上下载：Xcode 下载（Beta版打的包是不能提交到App Store上的）绝对官方源！！！绝对官方源！！！绝对官方源！！...

zhangao0086



微信



微博



QQ

2016-08-11 13:39:57

阅读数：142131

【代码笔记】iOS-一个tableView,两个section

一，效果图。二，工程图。三，代码。RootViewController.h #import @interface RootViewController : UIViewCo...

fanqingtulv

2016-07-14 09:22:42

阅读数：438

解决编译kernel出现WARNING:Section mismatch(es)

在编译内核时出现section mismatch警告 原因分析 解决方法在编译内核时出现section mismatch警告：警告如下：WARNING: modpost: Found 3 sect...

 flfihpv259 2016-07-12 16:45:42 阅读数：2003

2018新款扫地机!今日398元!仅限3天!


扫地机器人!德国品质,398元限时促销中!家用首选!



百度广告 ∨

非常规使用 UITableView, reloadSections时动画效果异常

使用UITableView时遇到一个UI问题及相应的解决方法

 w_z_z_1991 2018-07-25 13:25:38 阅读数：1866



0



收藏



评论



微信



微博



QQ