ARM（S3C2440 ）下解决的非法指令问题 (Illegal instruction)- 艾那的小强 - ChinaUnix 博客

> 最近在学习和做项目的时候需要搭建 s3c2440 的环境，遇到了一些问题（非法指令）和大家分享一下修正错误的过程。

最近在学习和做项目的时候需要搭建 s3c2440 的环境，遇到了一些问题（非法指令）和大家分享一下修正错误的过程。

一、我先介绍一下我们的实验环境：
内核版本: kernel-2.6.27-android_ok
编绎器: arm-2010q1-202-arm-none-linux-gnueabi
硬件:(S3C2440) 支持 armv4t 指令，
busybox 版本: busybox-1.17.0

我们这次实验的 kernel 相对来说版本不算老，编译可就是非常新的了。
他支持 ARMV7, 就是 cortex-A8,A9 系列的。
而我们的硬件 s3c2440 只有 armv4t 的指令集。

二、问题的出现：
我们用新的编译器编译出来的 uboot,kernel 均能正常执行。
当我们用它来编译 busybox，并且生成静态 busybox 的时候，有时候执行正常，
有时候执行就不正常。不正常的时候出现如下提示：
Illegal instruction
这个错误表明我们的程序执行了不正确的指令。
一般这种情况是因为我们编译起编译出了较高版本的 ARM 指令造成的。

三、排错
排错 1：让编译器编译出支持 s3c2440 的 armv4t, 在网络上查询得知增加 “-march=armv4t” 选项即可。

于是在 busybox make menuconfig 中编译选项中加入了上述参数，见下图

增加 -march=armv4t

残酷的事实验证后，结果发现还是不行。

这是为什么，我们该怎么做？有两个问题困扰着我：

1. 出现的非法指令是在哪，是什么指令？

2. 所加入的参数到底有没有起作用？

排错二：

现来寻找上述第一个问题的答案，

由于在交叉编译过程中产生了非法指令，故在执行过程中会产生异常，我们进入到内核的非法指令异常去看看。

于是到内核 (arch/arm/kernel/traps.c) 中找到相关的异常处理函数 (do_undefinstr)

其中部分代码为：

```
#ifdef CONFIG_DEBUG_USER
    if (user_debug & UDBG_UNDEFINED)
   {
       printk(KERN_INFO "%s (%d): undefined instruction:
pc=%p\n",    current->comm,
                task_pid_nr(current), pc);
       dump_instr(regs);
   }
#endif
```

通过上面的代码我们如果

CONFIG_DEBUG_USER 定义了，并且 user_debug 设置了参数，应该可以看到系统打印是哪个程序产生了异常。

通过 kernel 的. config 文件发现 CONFIG_DEBUG_USER 已经定义了。

user_debug 是一个命令行参数。

好了我们可以通过 uboot 打开：

打开的方法为在 U-boot bootargs 添加 user_debug=1;

（egs： setenv bootargs console=ttySAC0,115200 user_debug=1 saveenv   );

在 LINUX 再次执行程序，发现在原先的出现 "指令错误" 的上方多出了两条具体的错误信息。

打印出非法指令信息

以上图片中，是执行 mdev（mdev，是 busybox 的子程序）时出错。 pc=000ca8b4, 是指 mdev 的 ca8b4 处有个非法指令。

接下来从 busybox 中找出次指令：
进入 busybox 目录 ， 输入命令 (arm-none-linux-gnueabi-objdump -D busybox> bu.S 意思为将生的目标文件的汇编代码 dump 到 bu.S 文件中) ，
打开 bu.S(命令：vim bu.S) 文件，根据出现 "指令错误" 上面两条具体指令错误位置信息可以找到非法指令指置，我们查到该位置为 clz 指令：
在 ARM 官方网上息查询得知 clz 指令相关信息： （）

clz 指令体系结构：
此 ARM 指令可用于 ARMv5 及更高版本。
此 32 位 Thumb 指令可用于 ARMv6T2 及更高版本。
此指令无 16 位 Thumb 版本。

我们最终找到出现非法指令的位置，以及是什么指令。

好，接下来，我们需要第二步，回到 busybox 本身：

我们想把编译过程中的参数给打印出来，通过查看 busybox 中的 Makefile，得知由 quiet_决定，我们在编译 busybox 时加入选项参数 V=1（egs: make V=1);

通过编译过程所打印的信息知所加的选项参数在编译的时候起了作用，但在链接生成静态库的时候并没有起到作用。

虽然知道在 make menuconfig 加了相关参数（ - march=armv4t), 编译器并没有去执行这条语句，可能的原因是：我们所加的参数的格式不正确，或其它原因等）。于是我们就想找一种能解决此问题的方法。下面给出一种针对上述问题的解决的方案：
为了以防万一，我们手动将编译链接的地方都增加了 （-march=armv4t ）

我们可在 busybox 中的 Makefile.flags 中找到静态链接的位置以及编译的位置：

将：

ifeq ($(CONFIG_STATIC),y)CFLAGS_busybox += -static

endif 修改为：

ifeq ($(CONFIG_STATIC),y)CFLAGS_busybox += -static -march=armv4t

endif

将：

CPPFLAGS += $(call cc-option,-std=gnu99,)

修改为：CPPFLAGS += $(call cc-option,-std=gnu99 -march=armv4t,)

再进行编译就生成了我们所需指令版本的目标文件了。

相关软件下载地址：交叉编译器：arm-2010q1-202-arm-none-linux-gnueabi 下载地址：

具体为 Recommended Packagesbusybox-1.17.0 下载地址：

以上错误由第一小组共同完成，由 aka522 整理。E-mail:aka522@foxmail.com

参与人员有： lili,aka522,dubo,fengximing