Busybox支持中文的解决办法

2014-03-28 原文

转载：http://blog.csdn.net/wavemcu/article/details/7202908

****

作者：EasyWave **时间：2012.01.15**

类别：linux驱动开发 **声明：转载，请保留链接**

****

在嵌入式linux系统中，busybox是最常见的用来构建文件系统的。可是从busybox1.17.0以上之后，对ls命令不做修改是无法显示中文的。就算是内核设置了支持中文的话，在shell下用ls命令也是无法显示中文的，这是因为busybox1.17.0以后版本对中文的支持进行了限制。现在就来讲讲如何修改让busybox1.17.0以上版本支持中文，要想让busybox1.17.0以上支持中文，需要修改两个文

件：printable_string.c以及unicode.c

。下面来分析，为什么ls命令无法显示中文。请看printable_string.c未修改过的代码：

```
const char* FAST_FUNC printable_string(uni_stat_t *stats, const char *str)
{
    static char *saved[];
    static unsigned cur_saved; /* = 0 */

    char *dst;
    const char *s;

    s = str;
    while () {
        unsigned char c = *s;
        if (c == '\0') {
            /* 99+% of inputs do not need conversion */
            if (stats) {
                stats->byte_count = (s - str);
                stats->unicode_count = (s - str);
                stats->unicode_width = (s - str);
            }
            return str;
        }
        if (c < ' ')
            break;
        if (c >= 0x7f)
            break;
        s++;
    }

#if ENABLE_UNICODE_SUPPORT
    dst = unicode_conv_to_printable(stats, str);
#else
    {
        char *d = dst = xstrdup(str);
        while () {
            unsigned char c = *d;
            if (c == '\0')
```

```
                break;
                if (c < ' ' || c >= 0x7f)
                    *d = '?';
            d++;
        }
        if (stats) {
            stats->byte_count = (d - dst);
            stats->unicode_count = (d - dst);
            stats->unicode_width = (d - dst);
        }
    }
#endif

    free(saved[cur_saved]);
    saved[cur_saved] = dst;
    cur_saved = (cur_saved + ) & (ARRAY_SIZE(saved)-);

    return dst;
}
```

从上面代码23和24行以及37和38行可以看出：大于0x7F的字符直接被break掉，或者直接被“？”代替了。所以就算是linux内核设置了支持中文，也是无法显示出来的，被“？”代替了。修改红色加粗的代码如下：

```
const char* FAST_FUNC printable_string(uni_stat_t *stats, const char *str)
{
    static char *saved[];
    static unsigned cur_saved; /* = 0 */

    char *dst;
    const char *s;

    s = str;
    while () {
        unsigned char c = *s;
        if (c == '\0') {
            /* 99+% of inputs do not need conversion */
            if (stats) {
```

```
                stats->byte_count = (s - str);
                stats->unicode_count = (s - str);
                stats->unicode_width = (s - str);
            }
            return str;
        }
        if (c < ' ')
            break;
    /*
        if (c >= 0x7f)
            break;
    */
        s++;
    }

#if ENABLE_UNICODE_SUPPORT
    dst = unicode_conv_to_printable(stats, str);
#else
    {
        char *d = dst = xstrdup(str);
        while (1) {
            unsigned char c = *d;
            if (c == '\0')
                break;
            if (c < ' ' /*|| c >= 0x7f */)
                *d = '?';
            d++;
        }
        if (stats) {
            stats->byte_count = (d - dst);
            stats->unicode_count = (d - dst);
            stats->unicode_width = (d - dst);
        }
    }
#endif

    free(saved[cur_saved]);
    saved[cur_saved] = dst;
    cur_saved = (cur_saved + 1) & (ARRAY_SIZE(saved)-1);
```

```
54.
55.     return dst;
56. }
```


经过以上的修改之后，同时busybox1.17.0配置的时候没有选中[] **Support Unicode**的话，那么采用ls命令是可以看到中文的，这个我自己已经亲自测试过的。可是还有一种情况：busybox1.17.0在配置的时候选中了：[*] Support Unicode，见下：

```
1. 在配置里，有Support Unicode选上的：
2. Busybox Settings->General Configuration->
3.     | |[ ] Enable locale support (system needs locale for this to work)     | |
4.     | |[*] Support Unicode                                                  | |
5.     | |[*] Support for --long-options                                       | |
6.
```

那么这样还需要修改一个文件，这个文件就是：unicode.c。如果不修改这个文件，ls命令也是无法显示出中文的。见下未修改的代码：

```
1. static char* FAST_FUNC unicode_conv_to_printable2(uni_stat_t *stats, const char *src, unsigned width,
   int flags)
2. {
3.     char *dst;
4.     unsigned dst_len;
```

```
    unsigned uni_count;
    unsigned uni_width;

    if (unicode_status != UNICODE_ON) {
        char *d;
        if (flags & UNI_FLAG_PAD) {
            d = dst = xmalloc(width + );
            while ((int)--width >= ) {
                unsigned char c = *src;
                if (c == '\0') {
                    do
                        *d++ = ' ';
                    while ((int)--width >= );
                    break;
                }
                *d++ = (c >= ' ' && c < 0x7f) ? c : '?';
                src++;
            }
            *d = '\0';
        } else {
            d = dst = xstrndup(src, width);
            while (*d) {
                unsigned char c = *d;
                if (c < ' ' || c >= 0x7f)
                    *d = '?';
                d++;
            }
        }
        if (stats) {
            stats->byte_count = (d - dst);
            stats->unicode_count = (d - dst);
            stats->unicode_width = (d - dst);
        }
        return dst;
    }

    dst = NULL;
    uni_count = uni_width = ;
    dst_len = ;
```

```
    while () {
        int w;
        wchar_t wc;

#if ENABLE_UNICODE_USING_LOCALE
        {
            mbstate_t mbst = {  };
            ssize_t rc = mbsrtowcs(&wc, &src, , &mbst);
            /* If invalid sequence is seen: -1 is returned,
             * src points to the invalid sequence, errno = EILSEQ.
             * Else number of wchars (excluding terminating L'\0')
             * written to dest is returned.
             * If len (here: 1) non-L'\0' wchars stored at dest,
             * src points to the next char to be converted.
             * If string is completely converted: src = NULL.
             */
            if (rc == ) /* end-of-string */
                break;
            if (rc < ) { /* error */
                src++;
                goto subst;
            }
            if (!iswprint(wc))
                goto subst;
        }
#else
        src = mbstowc_internal(&wc, src);
        /* src is advanced to next mb char
         * wc == ERROR_WCHAR: invalid sequence is seen
         * else: wc is set
         */
        if (wc == ERROR_WCHAR) /* error */
            goto subst;
        if (wc == ) /* end-of-string */
            break;
#endif
        if (CONFIG_LAST_SUPPORTED_WCHAR && wc > CONFIG_LAST_SUPPORTED_WCHAR)
            goto subst;
        w = wcwidth(wc);
```

```
            if ((ENABLE_UNICODE_COMBINING_WCHARS && w < ) /* non-printable wchar */
             || (!ENABLE_UNICODE_COMBINING_WCHARS && w <= )
             || (!ENABLE_UNICODE_WIDE_WCHARS && w > )
            ) {
 subst:
                wc = CONFIG_SUBST_WCHAR;
                w = ;
            }
            width -= w;
            /* Note: if width == 0, we still may add more chars,
             * they may be zero-width or combining ones */
            if ((int)width < ) {
                /* can't add this wc, string would become longer than width */
                width += w;
                break;
            }

            uni_count++;
            uni_width += w;
            dst = xrealloc(dst, dst_len + MB_CUR_MAX);
#if ENABLE_UNICODE_USING_LOCALE
            {
                mbstate_t mbst = {  };
                dst_len += wcrtomb(&dst[dst_len], wc, &mbst);
            }
#else
            dst_len += wcrtomb_internal(&dst[dst_len], wc);
#endif
        }

        /* Pad to remaining width */
        if (flags & UNI_FLAG_PAD) {
            dst = xrealloc(dst, dst_len + width + );
            uni_count += width;
            uni_width += width;
            while ((int)--width >= ) {
                dst[dst_len++] = ' ';
            }
        }
```

```
122.      dst[dst_len] = '\0';
123.      if (stats) {
124.          stats->byte_count = dst_len;
125.          stats->unicode_count = uni_count;
126.          stats->unicode_width = uni_width;
127.      }
128.
129.      return dst;
130.  }
```

见上面20行和28行，需要修改一下，修改后的代码见下：

```
1.  static char* FAST_FUNC unicode_conv_to_printable2(uni_stat_t *stats, const char *src, unsigned width,
    int flags)
2.  {
3.      char *dst;
4.      unsigned dst_len;
5.      unsigned uni_count;
6.      unsigned uni_width;
7.
8.      if (unicode_status != UNICODE_ON) {
9.          char *d;
10.         if (flags & UNI_FLAG_PAD) {
11.             d = dst = xmalloc(width + );
12.             while ((int)--width >= ) {
13.                 unsigned char c = *src;
14.                 if (c == '\0') {
15.                     do
16.                         *d++ = ' ';
17.                     while ((int)--width >= );
18.                     break;
19.                 }
20.                 *d++ = (c >= ' '/* && c < 0x7f */) ? c : '?';
21.                 src++;
22.             }
23.             *d = '\0';
24.         } else {
25.             d = dst = xstrndup(src, width);
```

```
            while (*d) {
                unsigned char c = *d;
                if (c < ' '/* || c >= 0x7f */)
                    *d = '?';
                d++;
            }
        }
        if (stats) {
            stats->byte_count = (d - dst);
            stats->unicode_count = (d - dst);
            stats->unicode_width = (d - dst);
        }
        return dst;
    }

    dst = NULL;
    uni_count = uni_width = ;
    dst_len = ;
    while () {
        int w;
        wchar_t wc;

#if ENABLE_UNICODE_USING_LOCALE
        {
            mbstate_t mbst = {  };
            ssize_t rc = mbsrtowcs(&wc, &src, , &mbst);
            /* If invalid sequence is seen: -1 is returned,
             * src points to the invalid sequence, errno = EILSEQ.
             * Else number of wchars (excluding terminating L'\0')
             * written to dest is returned.
             * If len (here: 1) non-L'\0' wchars stored at dest,
             * src points to the next char to be converted.
             * If string is completely converted: src = NULL.
             */
            if (rc == ) /* end-of-string */
                break;
            if (rc < ) { /* error */
                src++;
                goto subst;
```

```
65.                 }
66.                 if (!iswprint(wc))
67.                     goto subst;
68.             }
69.     #else
70.             src = mbstowc_internal(&wc, src);
71.             /* src is advanced to next mb char
72.              * wc == ERROR_WCHAR: invalid sequence is seen
73.              * else: wc is set
74.              */
75.             if (wc == ERROR_WCHAR) /* error */
76.                 goto subst;
77.             if (wc == ) /* end-of-string */
78.                 break;
79.     #endif
80.             if (CONFIG_LAST_SUPPORTED_WCHAR && wc > CONFIG_LAST_SUPPORTED_WCHAR)
81.                 goto subst;
82.             w = wcwidth(wc);
83.             if ((ENABLE_UNICODE_COMBINING_WCHARS && w < ) /* non-printable wchar */
84.              || (!ENABLE_UNICODE_COMBINING_WCHARS && w <= )
85.              || (!ENABLE_UNICODE_WIDE_WCHARS && w > )
86.             ) {
87.      subst:
88.                 wc = CONFIG_SUBST_WCHAR;
89.                 w = ;
90.             }
91.             width -= w;
92.             /* Note: if width == 0, we still may add more chars,
93.              * they may be zero-width or combining ones */
94.             if ((int)width < ) {
95.                 /* can't add this wc, string would become longer than width */
96.                 width += w;
97.                 break;
98.             }
99.
100.            uni_count++;
101.            uni_width += w;
102.            dst = xrealloc(dst, dst_len + MB_CUR_MAX);
103.    #if ENABLE_UNICODE_USING_LOCALE
```

```
            {
                mbstate_t mbst = {  };
                dst_len += wcrtomb(&dst[dst_len], wc, &mbst);
            }
#else
            dst_len += wcrtomb_internal(&dst[dst_len], wc);
#endif
        }

        /* Pad to remaining width */
        if (flags & UNI_FLAG_PAD) {
            dst = xrealloc(dst, dst_len + width + );
            uni_count += width;
            uni_width += width;
            while ((int)--width >= ) {
                dst[dst_len++] = ' ';
            }
        }
        dst[dst_len] = '\0';
        if (stats) {
            stats->byte_count = dst_len;
            stats->unicode_count = uni_count;
            stats->unicode_width = uni_width;
        }

        return dst;
}
```

经过以上修改之后，就算配置支持Unicode，ls命令也是可以支持中文的。同时也可以进入中文目录可以文件夹。

Busybox支持中文的解决办法的更多相关文章

1. mac中matplotlib不支持中文的解决办法
   参考:https://blog.csdn.net/kaizei_pao/article/details/80795377 首先查看matplotlib已加载的字体: import matplotlib ...
2. python---不支持中文注释解决办法
   很神奇的一件事儿,pycharm不支持中文注释,具体解决办法: #-*- coding: utf- -*- 具体使用:
3. JqueryQrcode生成二维码不支持中文的解决办法
   JqueryQrcode.js有一个小小的缺点,就是默认不支持中文. 这跟js的机制有关系,jquery-qrcode这个库是采用 charCodeAt() 这个方式进行编码转换的,而这个方法默认会 ...
4. 使用iTextSharp 解析html生成pdf，xmlworker不支持中文的解决办法
   http://www.micmiu.com/opensource/expdoc/itext-xml-worker-cn/ 参考上面的文章,虽然是java的,但是和.net是对应的. 下载 html ...
5. IDLE3.6.3 Mac版不支持中文输入解决办法
   最近安装了IDLE 3.6.3版本 但是在IDLE中要输入中文注释时发现虽然输入法切换到了中文,但输入的还是英文.然后我在IDLE外试了下,输入中文没问题,于是就确认应该是IDLE的问题. 网上查询到 ...
6. koala不支持中文的解决办法（问题出现在使用中文字体时报错）
   C:\Program Files\Koala\rubygems\gems\sass-3.4.9\lib\sass 这是我的koala的安装路径,在sass文件夹下打开engine.rb(文本文档打开即 ...
7. Ubantu里面的Sublime Text3不支持中文的解决办法
   参考的大佬链接:https://github.com/lyfeyaj/sublime-text-imfix 更新然后将系统升级到最新版本,在linux终端输入 sudo apt-get update ...

8. [Linux] - CentOS中文乱码解决办法

   CentOS 7 终端中文乱码解决办法: 1.使用vim编辑locale.config文件: vim /etc/locale.conf 2.将LANG="en_US.UTF-8"修 ...

9. Oracle导入中文乱码解决办法

   Oracle导入中文乱码解决办法 一.确保各个客户端字符集的编码同服务器字符集编码一致 1-　　确定sqlplus字符集编码,如果是windows设置环境变量. 2-　　确保Sec ...

随机推荐

1. Codeforces 364

   A 第一题明显统计,注意0和long long(我WA,RE好几次) /* * Problem: A. Matrix * Author: Shun Yao */ #include <string ...

2. 【HDOJ】1058 Humble Numbers

   简单题,注意打表,以及输出格式.这里使用了可变参数. #include <stdio.h> #define MAXNUM 5845 #define ANS 2000000000 int b ...

3. [Qt] fontawesome图标

   fontawesome图标 fontawesome是一个图标的集合,里面有好多的图标,使用起来也还是非常方便的. 图标信息可以到官网去查:http://fontawesome.io/cheatshee ...

4. iOS开发中NSDate时间戳的转换--

   NSTimeInterval time =(NSTimeInterval)[model.day floatValue]; NSDate *date = [NSDate dateWithTimeInt ...

5. xBIM WeXplorer 设置模型颜色

   目录 基础 xBIM WeXplorer 简要介绍 xBIM WeXplorer xViewer 基本应用 xBIM WeXplorer xViewer 浏览器检查 xBIM WeXplorer xV ...

6. C++ 中vector的使用方法（转）

   原地址:http://blog.csdn.net/duan19920101/article/details/50617190/ 在c++中,vector是一个十分有用的容器. 作用:它能够像容器一样存 ...

7. Ubuntukylin 14.04 系统语言改成中文[转]

   1.在左侧点击"system setting" 2.按在图中方法设置 3.重启系统　参考地址:http://hi.baidu.com/thj2080/item/ae8e5dce ...

8. java 环境搭建

   一.安装jdk 下载jdk http://www.oracle.com/technetwork/java/javase/downloads 将下载的jdk文件放到 /opt 下解压 $sudo cp ...

9. 网页设计之字体和 CSS 调整

   调整 CSS 首先,我们先来看看问题的源头.CSS 的出现曾是技术的一大进步.你可以用一个集中式的样式表来装饰多个网页.如今很多 Web 开发者都会使用 Bootstrap 这样的框架. 这些框架当然 ...

10. PyQt5 简易计算器

    剩下计算函数(self.calculator)未实现,有兴趣的朋友可以实现它 [知识点] 1.利用循环添加按钮部件,及给每个按钮设置信号/槽 2.给按钮设置固定大小:button.setFixedSi ...