Android实现USB打印指导手册

实验室工作日志 `Android` `USB` `打印` `GhostScript` `foo2zjs`

简介：本手册通过指导将Linux系统上一些优秀的开源打印工具（如`GhostScript`、`foo2zjs`等）移植到Android平台，结合Android应用层的API接口（如`IText`、`Canvas`、`PDFDocument`等），实现Android设备控制USB打印机进行打印的功能。
撰写人：闫隆鑫(ylx601@gmail.com)，周星宇
时间：2017年4月

Version：1.0

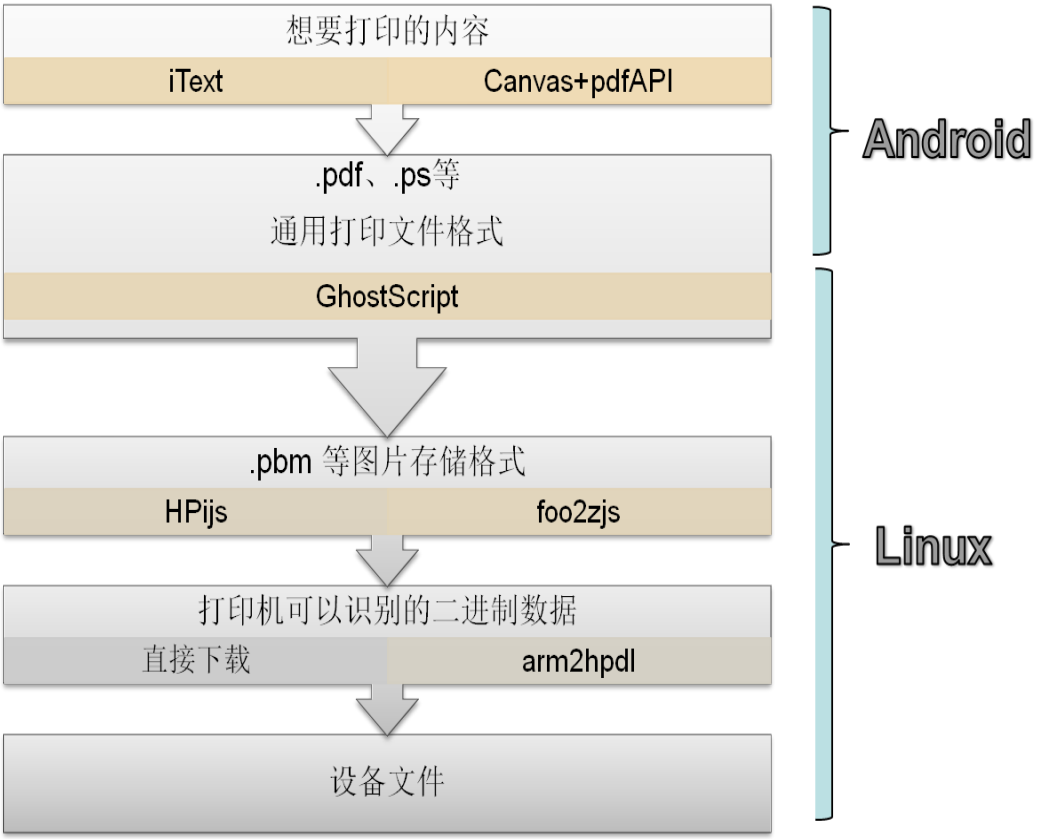
1.介绍

Android设备实现USB打印功能，是Android系统相对空白的一块。尽管Google在`Android4.4`版本之后推出了打印框架，但该框架支持的是有WiFi无线打印功能的打印机，对于市场上大量充斥的只支持USB打印的老式打印机仍然是爱莫能助。
本手册通过指导将Linux系统上一些优秀的开源打印工具（如`GhostScript`、`foo2zjs`等）移植到Android平台，结合Android应用层的API接口（如`IText`、`Canvas`、`PDFDocument`等），实现Android设备控制USB打印机进行打印的功能。

2.整体工作流程

如下图所示，整个方案的工作流程可以分为两大部分：

1. **Android应用层**：通过`iText`或者`Canvas+PDFDocumentAPI`等Java类库，将用户想要打印的内容转化为pdf格式文档。
2. **Linux内核层**：利用移植到Android内核的一些开源打印工具将pdf文档打印出来，具体分为两步：
    1. 通过`GhostScript`将pdf文档转换为.pbm图片存储格式的文件。
    2. 通过`foo2zjs`或者`HPijs`将pbm文件转化为打印机可以识别的二进制数据，输出到打印机。
    3. **前提条件**：打印机需要加载对应打印机型号的驱动文件。



Android设备打印流程

2.1.各模块介绍

- **iText**
    - IText API a pdf library
    - 一个Java类库，可根据要打印的文本内容，直接生成pdf文件的工具
- **Canvas**

    - Android提供的绘图工具，可在制定区域内绘制图形和文字

- **pdfDocument**

    - Android 提供的可根据Android系统内容（包括Canvas绘制的内容）生成pdf文件的类
    - pdfDocument-doc

- **GhostScript**

    - Ghostscript是一套建基于Adobe、PostScript及可移植文档格式（PDF）的页面描述语言等而编译成的开源**免费软件**。
    - 可以将pdf文件转化为.pbm、.ppm、.pgm等图片文件格式
        - 这三种格式的语法很接近，分别对就黑白图片、灰度图片和彩色图片
        - PBM支持单色图（1个像素位）
        - PGM支持灰度图形，能够读PBM图形和PGM图形，输出PGM图形
        - PPM支持真彩色图形，可以读上面所有格式，输出PPM图形

    - Ghostscript-use-doc
    - GhostScript本身也支持一些内置驱动和可作为输出设备直接被GhostScript调用的驱动（如：ijs驱动和Rinkj驱动），具体支持的型号及使用方法可以参考`doc/Android printer指导.docx`

- **HPijs**

    - HPijs是依赖于GNU Ghostscript的一款打印机驱动，此驱动支持惠普大部分打印机型号,主要支持较老型号的喷墨打印机。
    - 具体支持型号见`doc/打印机驱动支持.xlsx`
    - 由于该驱动支持的打印机型号较为老旧，本方案暂不支持该驱动。具体使用方法可以参考`doc/Android printer指导.docx`

- **foo2zjs**

    - foo2zjs是Foo2系列Linux打印机驱动的一种
    - Foo2系列支持多种打印机输出协议格式，每种协议对应一个驱动文件，如`foo2zjs/foo2xqx/foo2hp`，每种驱动文件支持的打印机型号见`doc/打印机驱动支持.xlsx`，编译和使用方法见本手册对应章节。
    - foo2zjs官方网站

- **arm2hpdl**

    - Foo2系列工具中的一个，可以将打印机的img文件转换为打印机在arm平台上的驱动文件。
    - 具体使用方法见本手册相应章节。

3.环境搭建

3.1.编译环境搭建

- Linux系统：`Ubuntu 12.04.2 64bit`系统
- 交叉编译器：`arm-linux-gcc-4.5.1`，编译器源码见`src/arm-linux-gcc-4.5.1-v6-vfp-20120301.tgz`
- `Ubuntu` 和 `arm-linux-gcc-4.5.1`的安装和配置，请参考`doc/Tiny4412用户手册2015.pdf`的3.2节

3.2.Android系统ROOT

因为本方案需要在Android应用层对内核和文件系统进行一系列操作，所以**必须**为Android设备`获取ROOT权限`。

3.3.配置Android内核

- 要实现Android设备的USB打印功能，**前提条件**是*Android*内核提供*USB*打印机支持。
- **验证方法**：将打印机的USB线插在Android设备的USB接口上，看是否有设备号是**180**的设备文件`/dev/usb/lp0`。若有，则说明此设备的内核支持USB打印机；若无，则需要配置内核。
- **配置内核方法**：
  - **条件**：需要有*Android*内核源码和相应的下载工具。
  - 重新编译Android内核：进入内核的目录后，在终端输入命令`# make menuconfig` 可打开图形化内核配置界面。
  - 依次选择`Device Drivers -->USB Support -->USB Printer Support`。保存退出后，重新编译内核并更新到板子上。

4.GhostScript的移植和使用

> 本手册使用的是GhostScript最新版9.21版本，移植方法与9.04及以前的版本不同。9.04及以前版本的移植可以参考Android 上移植 ghostscript-9.04 静态编译。

4.1.移植 GhostScript-9.21

4.1.1.源码下载

下载地址：http://downloads.ghostscript.com/public/

若没有网络可以使用已经下载好的源码，`src/ghostscript-9.21.tar.gz`为下载好的9.21源码。

建议： 下载时最好使用`迅雷`等下载工具进行下载，否则下载的文件很可能不完整或者损坏。

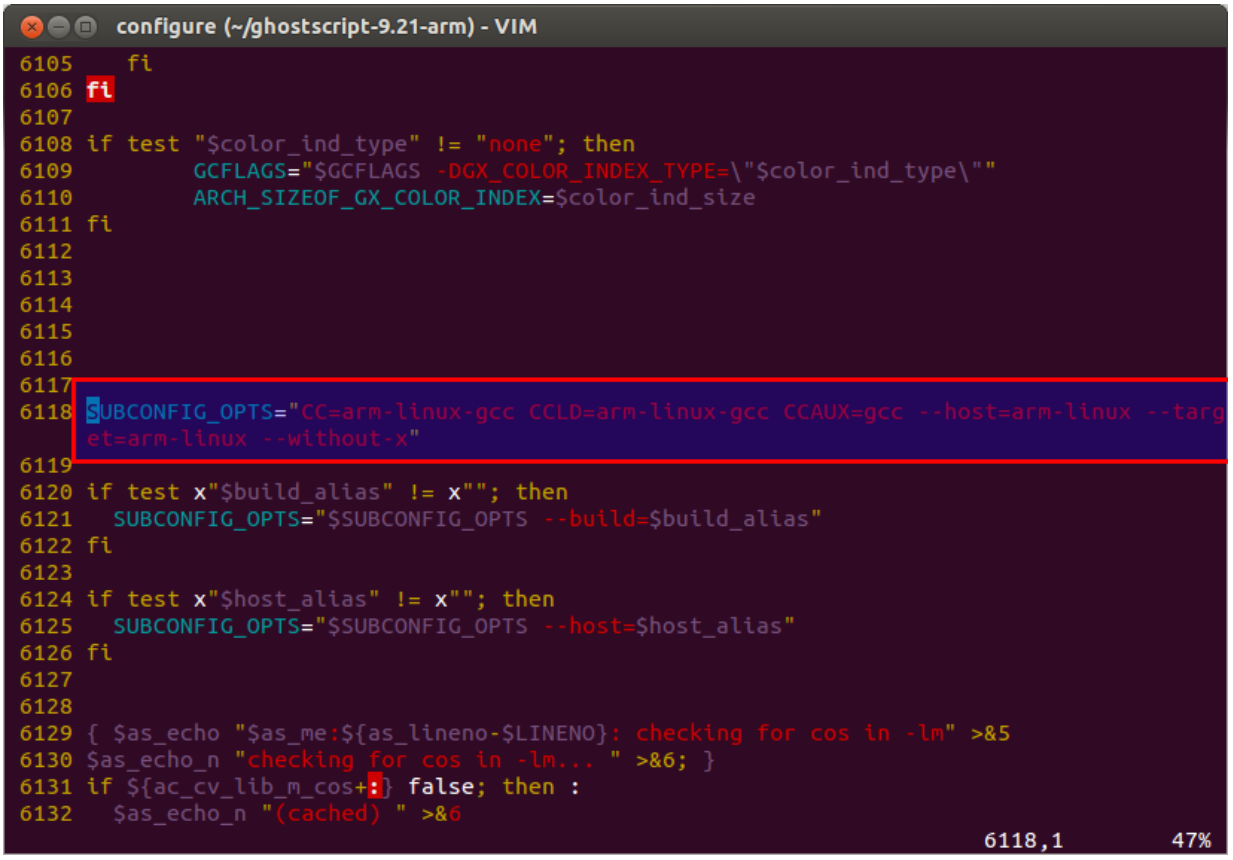
4.1.2.静态编译GhostScript 9.21 for ARM

> GhostScript 自**9.05**版本及以后，源码的架构发生了较大的变化，网络上流传的在Android 上移植ghostscript静态编译的方案都是基于9.04以前的版本，对9.04以后的版本均不再适用。笔者总结了GhostScript官网给出的9.21版本的使用手册How to build Ghostscript from source code，以及参考了Stack Overflow上类似的贴子Cross Compile GhostPDL for ARM9，总结出以下的方法。

- 解压源码文件
    - 打开终端，在放置源码压缩包的路径下执行：

```
tar zxvf ghostscript-9.21.tar.gz
cd ghostscript-9.21/
```

- 修改`configure`文件
    - 用编辑器打开源码主目录下的`configure`文件
    - 如下图所示，替换第6118行的`SUBCONFIG_OPTS`为：`CC=arm-linux-gcc CCLD=arm-linux-gcc CCAUX=gcc --host=arm-linux --target=arm-linux --without-x`
    - 关闭文件并保存

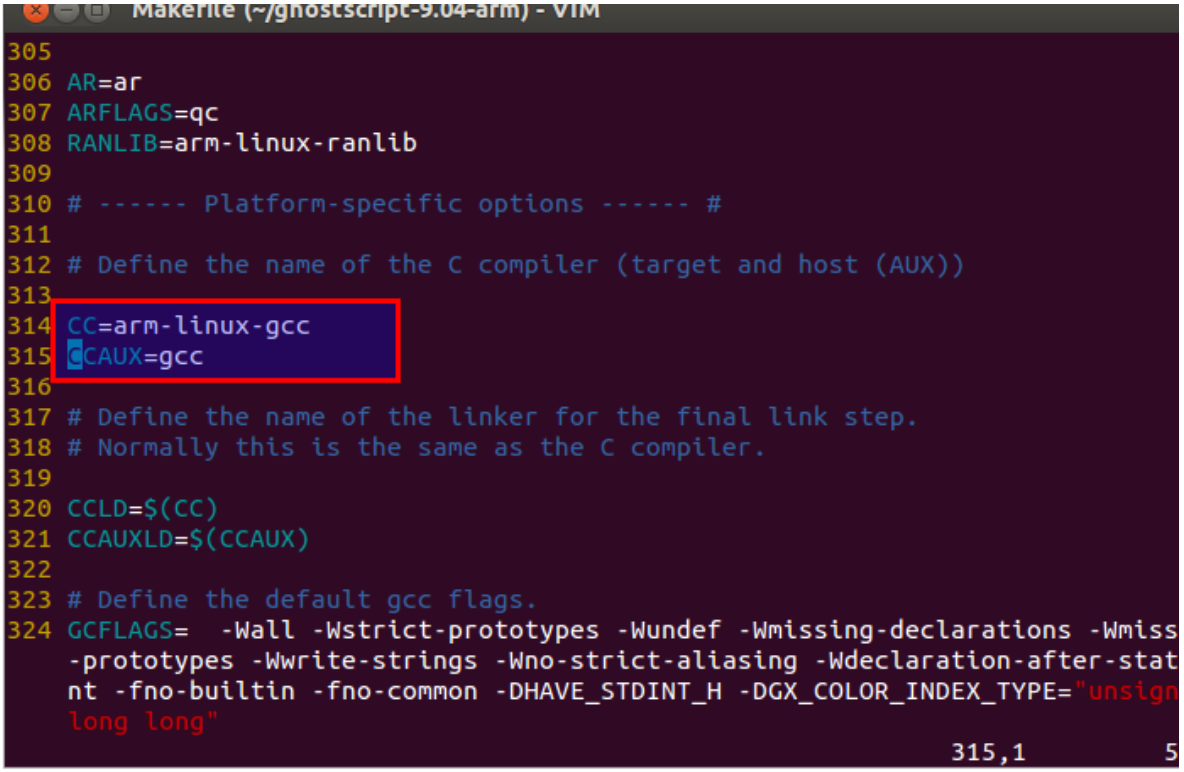


修改configure文件

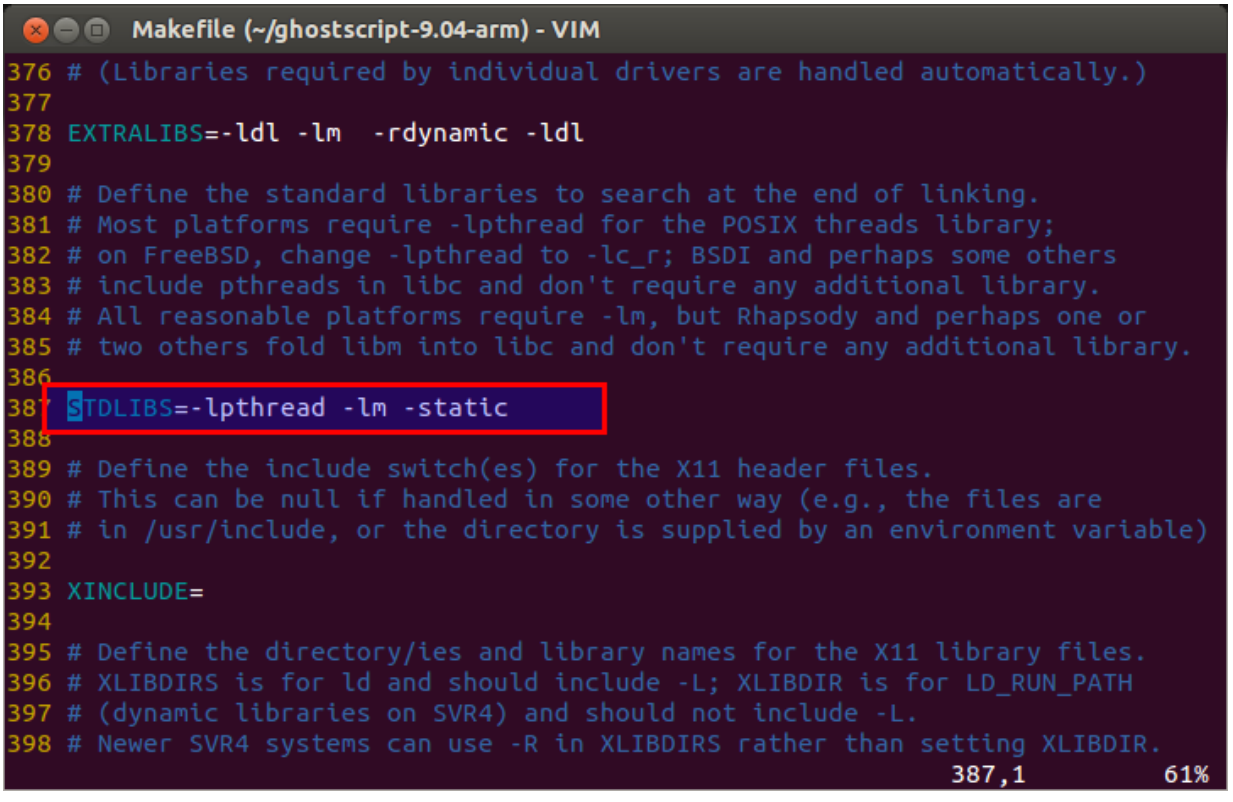
- 生成`Makefile`
  - 打开终端，在源码的主目录下执行:

```
1. ./configure CC=arm-linux-gcc CCLD=arm-linux-gcc CCAUX=gcc --hos
   t=arm-linux --target=arm-linux --without-x
```

- 修改`Makefile`
  - 执行完毕后，会在目录下生成Makefile
  - 用编辑器打开`Makefile`，看314、315行是否为下图所示



- 添加静态编译选项
  - 找到`Makefile`的第387行，在`STDLIBS`变量后添加`-static`，如下图所示



> 添加静态编译的目的是为了：使编译器在编译可执行文件的时候，将可执行文件需要调用的对应动态链接库(.so或.lib)中的部分提取出来，链接到可执行文件中去，使可执行文件在运行的时候不依赖于动态链接库。

- 编译GhostScript

- 关闭并保存`Makefile`
- 打开终端，在源码的目录下执行命令：

```
make CC=arm-linux-gcc CCLD=arm-linux-gcc CCAUX=gcc
```

- 待编译完成后，`ghostscript-9.21/bin`目录下的`gs`文件，即为ARM平台下的GhostScript 9.21 可执行文件

- [ ] 具体的安装和使用GhostScript的方法见**下一节**。
- [ ] 编译好的gs在目录`tools/gs/9.21`下。

4.3.GhostScript使用说明

4.3.1.准备GS需要的资源文件

如果在电脑上安装了GhostScript，在命令行输入`gs -h`会返回GhostScript的相关信息，在输出信息的最下面几行找到如下几行。**GhostScript运行时在这些路径中查找初始化文件和字体等的资源文件。**

```
Search path:
   %rom%Resource/Init/ : %rom%lib/ :
   /usr/local/share/ghostscript/9.21/Resource/Init :
   /usr/local/share/ghostscript/9.21/lib :
   /usr/local/share/ghostscript/9.21/Resource/Font :
   /usr/local/share/ghostscript/fonts :
   /usr/local/share/fonts/default/ghostscript :
   /usr/local/share/fonts/default/Type1 :
   /usr/local/share/fonts/default/TrueType : /usr/lib/DPS/outline/base :
   /usr/openwin/lib/X11/fonts/Type1 : /usr/openwin/lib/X11/fonts/TrueType
Initialization files are compiled into the executable.
```

gs 的搜索路径

GhostScript运行时以**如下流程**在系统中找到它需要的字体

1. GhostScrip在`lib/Fontmap.GS`文件中查找字体对应。
   例如，Fontmap.GS中如下语句将**URWBookmanL-DemiBold**字体对应到**b018015l.pfb**文件。

```
/URWBookmanL-DemiBold   (b018015l.pfb)  ;
```

2. GhostScript在字体路径这种查找对应的pfb文件。GhostScript查找的字体路径可以使用`gs -h`命令查看。例如：`/usr/local/share/ghostscript/fonts`

刚刚编译好的GhostScript不包含字体文件，我们需要把字体文件从编译GhostScript的电脑中拷贝到GhostScript字体目录（例如：我使用的Ubuntu 12.04.2 64bit系统中字体对应路径为：`/usr/share/fonts/type1/gsfonts/`）。

我们将gs9.21源码目录下的/lib、/Resource以及对应系统中字体目录拷贝到新建的`/usr/local/share/ghostscript`目录下，压缩打包为一个tar.gz包，之后压缩到Android设备的根目录下备用。
可以参考我已经将打包好的压缩包，放在`tools/gs/9.21/gs.tar.gz`

4.3.2.在Android上安装GS

在交叉编译完成GhostScript后，将生成的**gs**文件拷贝到开发板中`/system/bin/`目录下，修改使用权限，并配置所需的资源文件。具体步骤为：

1. 确保设备可进行`adb`调试
2. 为设备安装busybox，便于操作做(若系统自带，则忽略)。 busybox的二进制文件在`tools/busybox`。安装步骤参考网址：Android安装BusyBox —— 完整的bash shell
3. 使用adb push命令将包含gs放在任一目录
4. 在adb shell内用cd命令进入上述目录，输入如下命令。

```
1. mount -o remount,rw /system #重新挂载文件目录，获得读写权限
2. mount -o remount,rw /
3. busybox cp ./gs /system/bin #将gs拷贝至/system/bin下
4. busybox chmod a+x /system/bin/gs /system/bin/hpijs #给gs执行权限
5. busybox tar xvzf ./gs.tar.gz -C / #将gs所需的资源文件解压到根目录下
6. chmod 0666 /dev/usb/lp0  #修改打印机权限
```

4. 在`adb shell`里执行命令`gs -v`，若返回如下，就说明GhostScript 9.21 在Android平台上安装成功了。

```
1. GPL Ghostscript 9.21 (2017-03-16)
2. Copyright (C) 2017 Artifex Software, Inc.  All rights reserved.
```

4.3.3.使用说明

* 命令行说明*
在确保安装无误之后即可在命令行运行GhostScript。
首先我们可以通过一个最简单的应用来了解GhostScript使用方法。

```
1. gs -sDEVICE=epson -sOutputFile=ABC.xyz myfile.ps
```

此命令指定**设备模型**为epson,**输入文件**为myfile.ps,**输出文件**为ABC.xyz

另外，gs命令和输入文件之间还可加入一些命令参数选项。（具体请参考GhostScript官网GhostScript9.21-use-doc ）

其中常用的命令行参数如下：

```
-sOutputFile 指定输出文件
-r<number1>x<number2> 指定分辨率
-dBATCH 当处理完所有文件之后退出GhostScript
-dNOPAUSE 每一页装换之间没有停顿
-sDEVICE 设定输出文件
-dFirstPage 第一页
-dLastPage 最后一页
-sPAPERSIZE 纸张大小
-g<number1>x<number2> 指定图片像素（一般不指定，使用默认值）
-q 忽略部分命令行输出信息（报错、警告信息）
```

使用命令行参数方法可参考如下实例：

```
gs -q -dBATCH -dSAFER -dNOPAUSE -sPAPERSIZE=a4 -r1200x600 -sDEVICE=pbmraw -sOutputFile=chinese.pbm chinese.pdf
```

Android应用内调用

在Android的APP中需要使用**java输出流**的形式执行**shell脚本**来调用GhostScript相关功能例如：

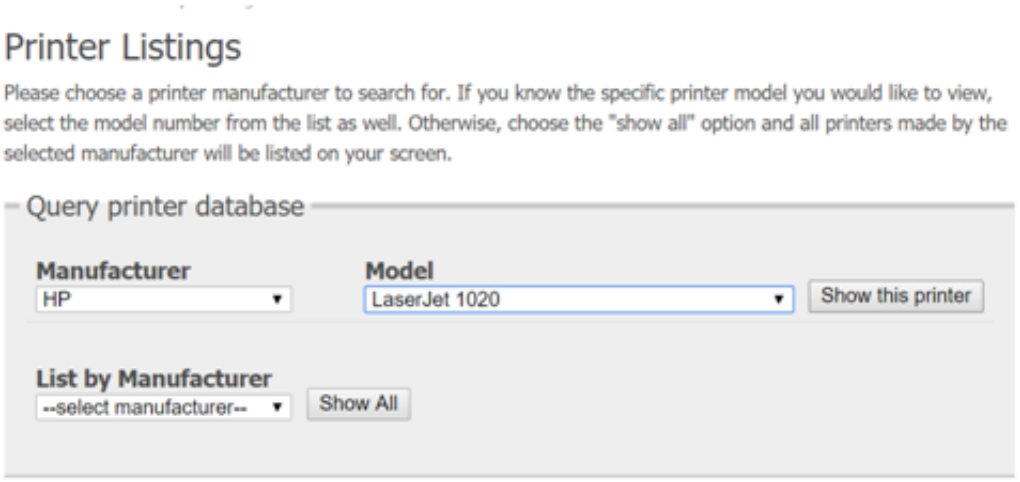
```
    Process process = null;
    DataOutputStream os = null;
    try {
        process = Runtime.getRuntime().exec("su"); //使用root权限
        os = new DataOutputStream(process.getOutputStream());
        os.writeBytes("gs -q -dBATCH -dSAFER -dNOPAUSE -sPAPERSIZE=a4 -r1200x600 -sDEVICE=pbmraw -sOutputFile=chinese.pbm chinese.pdf \n");
        os.flush();
        process.waitFor();
    } catch (Exception e) {
        return false;
    } finally {
        try {
            if (os != null) {
                os.close();
            }
            process.destroy();
            } catch (Exception e) {
            }
    }
```

此段程序在android应用程序中使用java输出流的形式执行上面提到的GhostScript命令。

5.foo2zjs工具及打印机驱动的移植与使用

5.1.打印机驱动下载

1. 在OpenPrinting网站上查找相应打印机型号所使用的驱动并下载。
   下图搜索的型号为HP laserjet 1020

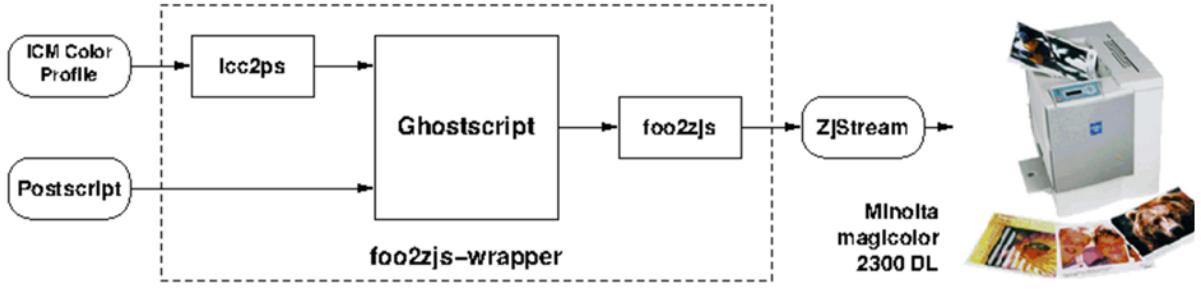


OpenPrinting下载驱动

2. 本章的5.2.4节介绍了另外一种获取打印机驱动的方法。
3. 常见的惠普打印机驱动，放置在`/drivers`目录下。

5.2.foo2zjs工具的移植

Foo2系列支持多种打印机输出协议格式，每种协议对应一个驱动文件，如foo2zjs/foo2xqx/foo2hp…….
以`foo2zjs`为例进行介绍:foo2zjs将GhostScript生成的pbm文件转化成符合ZJStream协议字节流，从而驱动打印机进行打印。（如下图所示）



foo2zjs打印流程

具体参考官网http://foo2zjs.rkkda.com/

5.2.1.下载并解压源码

在Ubuntu的命令行里输入：

```
wget -O foo2zjs.tar.gz http://foo2zjs.rkkda.com/foo2zjs.tar.gz
tar zxf foo2zjs.tar.gz
cd foo2zjs
```

（若没有网络可以使用已经下载好的源码，附于文件夹中 `src/foo2zjs.tar.gz` ）

5.2.2.安装编译所需的工具

（此步骤较容易被忽视）
`Ubuntu` 系统请安装如下工具

```
$ sudo apt-get install build-essential tix foomatic-filters groff
dc
```

参考foo2zjs安装官方说明—— `foo22zjs/INSTALL` 第249行

```
249 UBUNTU NOTES
250 ------------
251     Install build-essential, tix, foomatic-filters, groff, dc FIRST:
252     $ sudo apt-get install build-essential tix foomatic-filters groff dc
253
```

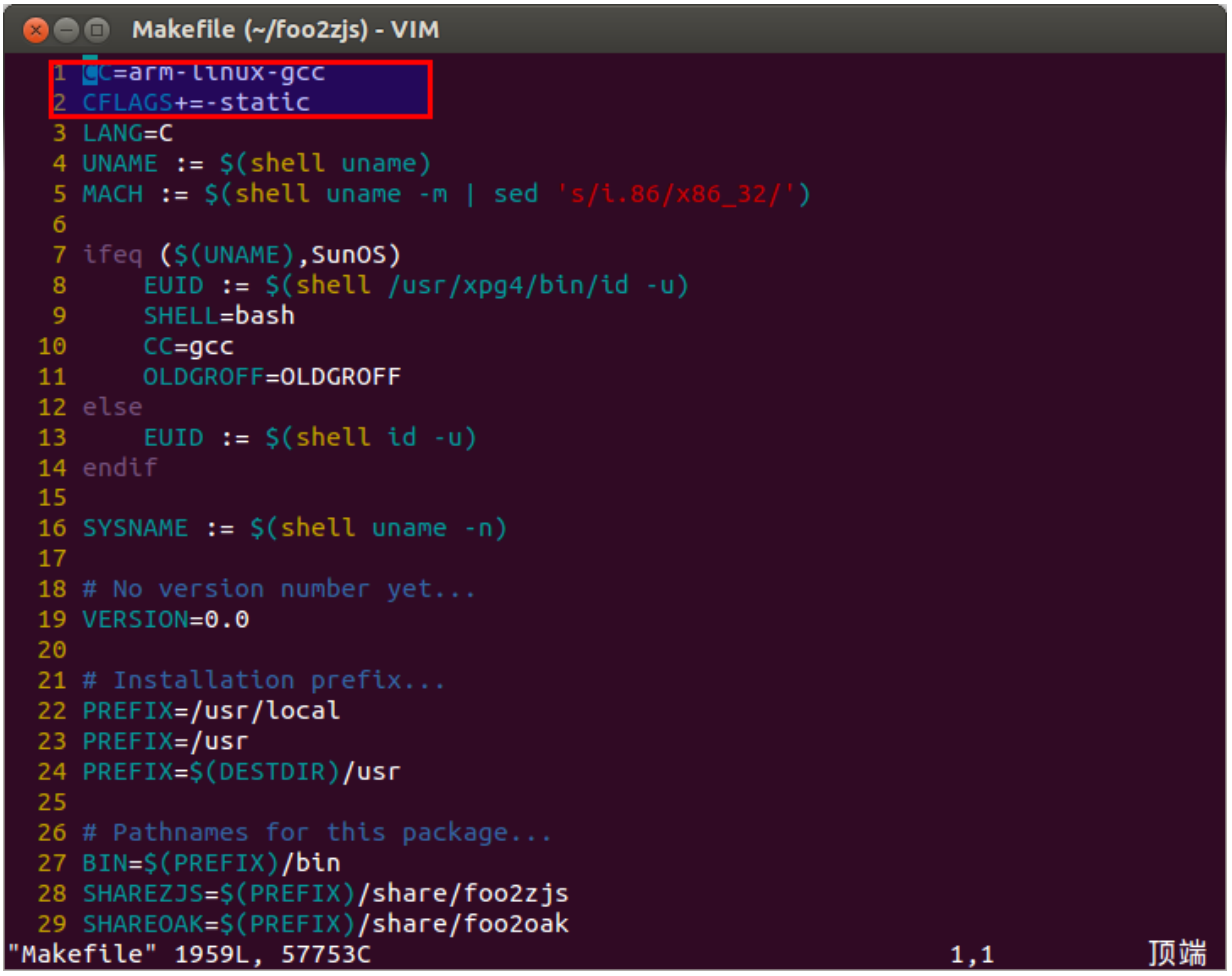
Ubuntu安装foo2zjs环境配置

注意：不同型号的linux系统安装的工具可能不同，请查阅 `foo22zjs/INSTALL` 的229~512行

5.2.3. 编译foo2zjs

1. 以gcc直接 `make` ，生成的 `arm2hpdl` 文件要保存到其他文件夹，以备后续使用。
2. 修改Makefile，在Makefile开头加上两行：（如下图所示）

```
CC=arm-linux-gcc    #使用交叉工具链
CFLAGS+= -static     #静态编译
```



修改Makefile

3. 执行make
   编译成功后，foo2zjs根目录会出现foo2zjs/foo2xqx/foo2hp……等一系列驱动程序。驱动程序大小均为600kb左右（大小太小可能编译有问题，请确保以上步骤执行无误）。这些程序就是在ARM平台上可以运行的foo2系列驱动程序

`<tools/foo2zjs>里`包含了所有生成的ARM平台的foo2zjs工具，可以直接使用

5.2.4.利用arm2hpdl获取打印驱动

以HP laserjet 1020型号打印机为例

1. （这一步需确保联网）在foo2zjs目录下执行：

```
1. ./getweb 1020
```

这一步将得到**sihp1020.img**文件。

2. 用**5.1.3节第一步**保存下来的arm2hpdl拷贝到当前文件夹下，覆盖新生成的arm2hpdl。并执行：**./arm2hpdl sihp1020.img > sihp1020.dl**、
3. 生成的 **sihp1020.dl**即为在ARM-Linux平台上的hp1020打印机驱动文件
   所有可获取的驱动型号见`foo22zjs/INSTALL`第45~110行。下图列出部分内容

```
Get extra files from the web, such as .ICM profiles (for color correction)
and firmware.  Select the model number for your printer:
    $ ./getweb 1025     # Get HP LaserJet Pro CP1025nw .ICM files
    $ ./getweb 1215 # Get HP Color LaserJet CP1215 .ICM files
    $ ./getweb 1500 # Get HP Color LaserJet 1500 .ICM files
    $ ./getweb 1600 # Get HP Color LaserJet 1600 .ICM files
    $ ./getweb 2600n    # Get HP Color LaserJet 2600n .ICM files

    $ ./getweb 1600w    # Get Konica Minolta magicolor 1600W .ICM files
    $ ./getweb 1680     # Get Konica Minolta magicolor 1680MF .ICM files
    $ ./getweb 1690     # Get Konica Minolta magicolor 1690MF .ICM files
    $ ./getweb 2480 # Get Konica Minolta magicolor 2480 MF .ICM files
    $ ./getweb 2490 # Get Konica Minolta magicolor 2490 MF .ICM files
    $ ./getweb 2530 # Get Konica Minolta magicolor 2530 DL .ICM files
    $ ./getweb 4690     # Get Konica Minolta magicolor 4690MF .ICM files
    $ ./getweb 110      # Get Oki C110 .ICM files
    $ ./getweb 6115 # Get Xerox Phaser 6115MFP .ICM files
    $ ./getweb 6121     # Get Xerox Phaser 6121MFP.ICM files

    $ ./getweb cpwl # Get Minolta Color PageWorks/Pro L .ICM files
    $ ./getweb 2200 # Get Minolta/QMS magicolor 2200 DL .ICM files
    $ ./getweb 2300 # Get Minolta/QMS magicolor 2300 DL .ICM files
    $ ./getweb 2430 # Get Konica Minolta magicolor 2430 DL .ICM files

    $ ./getweb 300      # Get Samsung CLP-300 .ICM files
    $ ./getweb 315      # Get Samsung CLP-315 .ICM files
    $ ./getweb 325      # Get Samsung CLP-325 .ICM files
    $ ./getweb 360      # Get Samsung CLP-360 .ICM files
    $ ./getweb 365      # Get Samsung CLP-365 .ICM files
    $ ./getweb 600      # Get Samsung CLP-600 .ICM files
    $ ./getweb 610      # Get Samsung CLP-610 .ICM files
    $ ./getweb 2160     # Get Samsung CLX-2160 .ICM files
    $ ./getweb 3160     # Get Samsung CLX-3160 .ICM files
    $ ./getweb 3175     # Get Samsung CLX-3175 .ICM files
    $ ./getweb 3185     # Get Samsung CLX-3185 .ICM files
    $ ./getweb 6110     # Get Xerox Phaser 6110 and 6110MFP .ICM files
```

foo2zjs可获取的驱动型号

5.3.foo2zjs的安装与驱动的加载

以HP LaserJet 1020打印机为例

1. 将**sihp1020.dl**拷贝到平板的任意文件夹中，将编译生成的**foo2zjs**和**usb_printerid**拷贝到平板的/system/bin/目录下。
2. 在**adb shell**里执行：

```
busybox chmod a+x /system/bin/foo2zjs
busybox chmod a+x /system/bin/usb_printerid
```

把上述文件修改成可执行的权限

3. **检查foo2zjs编译和安装是否成功**
   在命令行（android系统使用adb工具）输入 foo2zjs –V，输出如下信息，证明foo2zjs编译和安装成功。（记得V是大写的）

```
1|root@RAGE12:/ # foo2zjs -V
foo2zjs -V
$Id: foo2zjs.c,v 1.109 2011/11/12 16:39:53 rick Exp $
```

foo2zjs -V

4. **加载打印机驱动**
   插上打印机，检查设备文件，等待`/dev/usb/lp0`创建完成。执行

```
1. cp sihp1020.dl /dev/usb/lp0
```

这一步加载打印机固件，如果固件加载正常的话，会听到打印机空转的声音。
注意！！有些型号打印机（*HP_LASERJET_1000/1005/1018/1020, P1005/P1006/P1007/P1008/P1505*）需要使用特定语句加载驱动.否则会加载失败
参考`foo22zjs/INSTALL`的539行

```
539  HP LASERJET 1000/1005/1018/1020, P1005/P1006/P1007/P1008/P1505 NOTES
540  ----------------------------------------------------------------------
541      These printers need their firmware downloaded to them every time they
542      are powered up.
543
544      On Linux with USB connected printer:
545          If you are running Linux and the printer is connected via USB, you
546          can arrange for the firmware to be automatically downloaded to the
547          printer by performing one more installation step:
548
549          # make install-hotplug
550
551          Power off then on the printer. Light should flash orange for
552          ~5 seconds as the firmware is getting downloaded.
553
554      On another OS or with a parallel port connected printer:
555
556          You must send a firmware file to the printer each time you power it
557          up.  If you downloaded the extra files for the HP above, a typical
558          command line to load the firmware would be ONE of these:
559
560          # cat /usr/share/foo2zjs/firmware/sihp1000.dl > /dev/usb/lp0
561          # cat /usr/share/foo2zjs/firmware/sihp1005.dl > /dev/usb/lp0
562          # cat /usr/share/foo2zjs/firmware/sihp1018.dl > /dev/usb/lp0
563          # cat /usr/share/foo2zjs/firmware/sihp1020.dl > /dev/usb/lp0
564          # cat /usr/share/foo2xqx/firmware/sihpP1005.dl > /dev/usb/lp0
565          # cat /usr/share/foo2xqx/firmware/sihpP1006.dl > /dev/usb/lp0
566          # cat /usr/share/foo2xqx/firmware/sihpP1505.dl > /dev/usb/lp0
567
568          Light should flash orange for ~5 seconds as the firmware is
569          getting downloaded.
570
571      On Mac OS X:
```

加载打印机驱动

5.4.foo2zjs的使用

此处需满足两个前提条件：**1. GhostScript已安装**; **2. 打印机的驱动已经加载** 。

1. 将pdf文件转化成打印机可识别的pbm文件格式

（假设想要打印的pdf文件为`/sdcard/SimPr/files/mytest1.pdf`）

```
gs -q -dBATCH -dSAFER -dNOPAUSE -sPAPERSIZE=a4 -r1200x600 -sDEVICE=pbmraw -sOutputFile=mytest1.pbm /sdcard/SimPr/files/mytest1.pdf
```

具体参数含义请参考GS使用手册

- -r1200x600根据foo2驱动设置的分辨率，具体参考`doc/foo2zjs.pdf`
- -sDEVICE=pbmraw 输出数据指定为pbm格式
- -sOutputFile=chinese.pbm 输出文件

2.将pbm文件推送到打印机中进行打印

```
foo2zjs -z1 -p9 -r1200x600 mytest1.pbm > /dev/usb/lp0
```

- foo2zjs不同驱动见openPrinting官网
- -z1 打印机所使用的的设备模型，具体参考`<doc/foo2zjs.pdf>`
- –p9 使用A4纸。（!!不是所有驱动都是p9, 具体参考`<doc/foo2zjs.pdf>`）
- -r1200x600 指定分辨率，一般情况可以空着使用默认值
- /dev/usb/lp0 打印机字节流输出设备，这个不能改

3.在Android APP里的调用与`4.3.3`节类似，使用Java输出流来执行。

5.5.打印机型号的识别

> 如果我们已经有了很多型号的打印机驱动，如何正确识别出打印机的型号，然后加载正确的打印机驱动文件呢？
> 答案是使用foo2zjs系列工具——usbprinter_id

在adb shell里执行：

```
usb_printerid /dev/usb/lp0
```

能够得到类似下面的信息，这段信息里包含了打印机名称HP LaserJet 1020：

```
GET_DEVICE_ID string:MFG:Hewlett-Packard; MDL:HP LaserJet 1020; CMD:HBS,PJL,ACL; CLS:PRINTER;DES:HP LaserJet 1020; FWVER:20090916;
```

注意：linux自动生成的/dev/usb/lp0节点**即使不加载驱动**也可以让usb_printerid显示上述信息。此时lp0节点大小为0Byte。当加载驱动失败时lp0节点大于0Byte，但是调用usb_printerid工具会出现报错。
因此我们可以在加载打印机驱动前，通过使用该工具实现识别打印机型号的功能，根据打印机型号再加载合适的驱动文件。

在Android的APP里，可以通过获取**JAVA输入流** `DataInputStream` 的方式获取上面的返回信息，然后截取对应打印机型号的字符串，示例代码如下。

```java
public String Identify() {
    String printerId = "";
    Process process = null;
    DataInputStream in = null;
    DataOutputStream os = null;
    try {
        process = Runtime.getRuntime().exec("su"); //切换到root帐号
        os = new DataOutputStream(process.getOutputStream());
        in = new DataInputStream(process.getInputStream());
        os.writeBytes("usb_printerid /dev/usb/lp0\n");
        os.flush();

        Log.d("Identify", "write");

        printerId = in.readLine();//获取输入流
        printerId = printerId + in.readLine();

        Log.d("Identify", printerId);

        os.writeBytes("exit\n");
        os.flush();

        process.waitFor();
    } catch (Exception e) {
        return "error";
    } finally {
        try {
            if (os != null) {
                os.close();
            }
            if (in != null) {
                in.close();
            }
            process.destroy();
        } catch (Exception e) {
        }
    }
    int begin = printerId.indexOf("DES:");
    if (begin > -1) {
        int end = printerId.indexOf(";", begin);
        printerId = printerId.substring(begin + 4, end);
        return printerId;
    }
    return "error";
}
```

6.Itext使用指南

6..1 itext使用

将itext以jar包的形式包含到java或android工程中即可使用。
Android studio中导入jar包方法参考
http://jingyan.baidu.com/article/e6c8503c7190b7e54f1a1893.html

个人感觉使用itext简单方便的方法是在itext官方例程上进行修改，从而插入自己的PDF内容。
快速上手：`docs/iText中文教程.pdf`
官网iText例程：http://developers.itextpdf.com/content/itext-5-examples
官网FAQ：http://developers.itextpdf.com/content/best-itext-questions-stackoverview/tables

6.2 输出中文

itext中加入中文有两种方法：

1. **使用itext-assian.jar**
   itext-assian.jar是itext官方提供的亚洲字体支持包。使用此支持包需要
   1) 将`itext-assian.jar`和`itext.jar`一同包含入java/android工程。注意：itext-assian.jar包中CMAP的路径要和itext.jar包中字体CMAP路径一致，否则会报错提示字体未找到。
   2) 导入`itext-assian.jar`成功之后使用如下代码加载中文字体。

```
1. BaseFont bfChinese = BaseFont.createFont("STSong-Light", "UniGB-UCS2-H", BaseFont.NOT_EMBEDDED);
2. Font FontChinese = new Font(bfChinese, 12, Font.NORMAL);
```

注意！！使用itext-assian.jar生成的PDF在做为GhostScript输入文件时会报错：找不到资源。所以建议使用方法2

2. 使用字库
   1)在windows电脑C：/windows/Fonts/中选择字体（最好选择比较常用的字体，不要选择微软字体）
   2)将字体放入android平台的SD卡中（如/storage/sdcard0/printer_tools/fonts）
   3)在itext中调用字体文件。例：黑体字体文件 black.ttf

```
1. BaseFont bfChinese = BaseFont.createFont(PrinterParam.ToolDir+"bl
   ack.ttf",BaseFont.IDENTITY_H,BaseFont.NOT_EMBEDDED);
2. Font FontChinese = new Font(bfChinese, 30, Font.NORMAL);
```