Zedboard自己建立NFS实践

目录

[搭环境： 1](#_Toc434525522)

[一. 建立宿主服务器 2](#_Toc434525523)

[二：编译含NFS支持的zImage 3](#_Toc434525524)

[三．修改uboot传递参数bootargs 5](#_Toc434525525)

[工程所有文件备份： 7](#_Toc434525526)

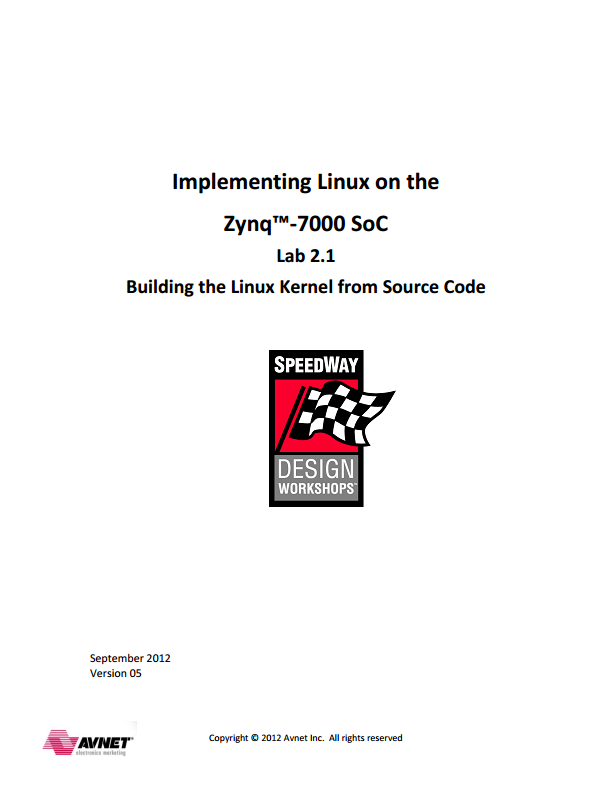
## 搭环境：

制作Uboot：

制作：Linux内核：



首先自己要有可用的Uboot、zImage、ramdisk、dts

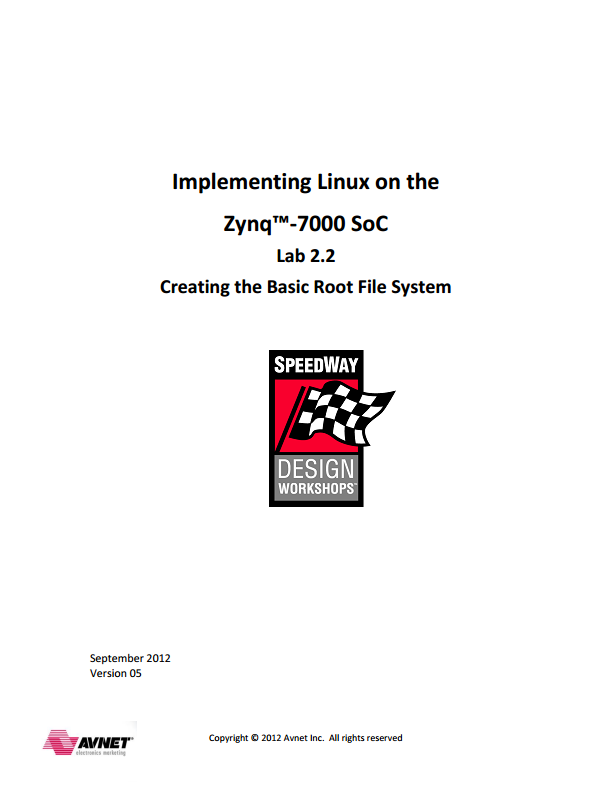
下载最新Linux内核 ：

$ git clone <https://github.com/Xilinx/linux-xlnx.git>

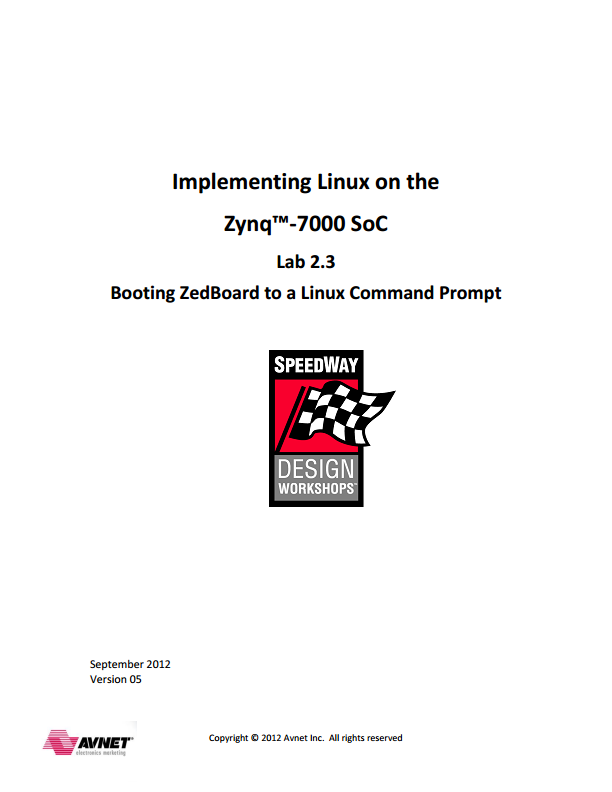
制作编译SSH、制作根文件系统：

编译SSH时候要注意采用新版的dropbear，不然会有错误编译不下去

可以参考文章：<http://blog.chinaunix.net/uid-23065002-id-3179295.html>



整个系统跑起来：



下面要使zedboard的文件系统跑的是NFS主要需要在三个地方修改：

## 一. 建立宿主服务器

我这里是Ubuntu 14.04LTS

**1、安装Ubuntu nfs**

Ubuntu上默认是没有安装Ubuntu nfs服务器的，因此我们首先安装Ubuntu nfs服务器端：

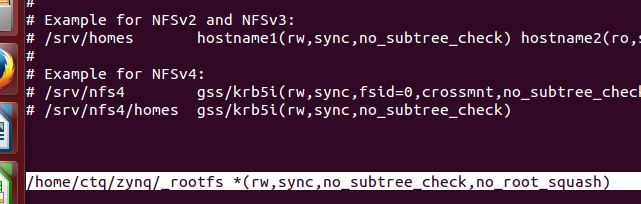
> apt-get install nfs-kernel-server

**2、配置/etc/exports**

Ubuntu nfs允许挂载的目录及权限在文件/etc/exports中进行了定义例如，我们要将根目录下的/nfs目录共享出来，那么我们需要在/etc/exports文件末尾添加如下一行：

>sudo nano /etc/exports

加上一行：/home/ctq/zynq/\_rootfs \*(rw,sync,no\_subtree\_check,no\_root\_squash)



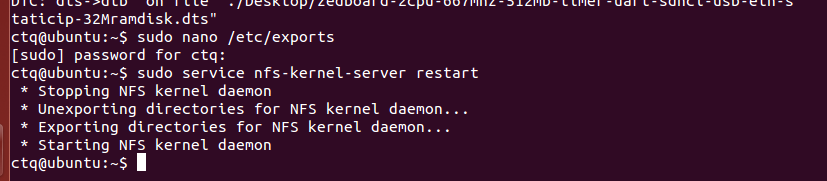
其中：/nfs是要共享的目录，\*代表允许所有的网络段访问，rw是可读写权限,sync是资料同步写入内存和硬盘，no\_root\_squash是Ubuntu nfs客户端分享目录使用者的权限，如果客户端使用的是root用户，那么对于该共享目录而言，该客户端就具有root权限，记住要mkdir  /nfs并在这个目录下新建一些文件帮助我们测试是否成功。

**3、关闭Ubuntu的防火墙**

> ufw disable   防火墙在系统启动时自动禁用  （启用就是enable）

**4、Ubuntu nfs重启服务**

$sudo service nfs-kernel-server restart



**5、测试Ubuntu nfs**

此时可以运行以下命令来显示一下共享出来的目录：

$showmount -e

现在可以在本机上试一下：

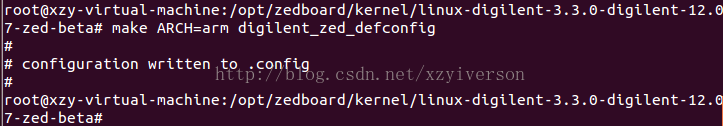
#sudo mount -t nfs localhost:/home/ctq/zynq/\_rootfs  /mnt

## 二：编译含NFS支持的zImage

1 .配置内核源码

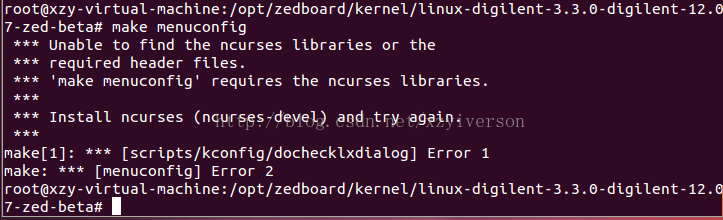
进入解压的目录，在arch/arm/configs/目录下有这个配置文件  digilent\_zed\_defconfig （不用我们设置，执行下面的命令他会自己寻找。）

make ARCH=arm xilinx\_zynq\_defconfig



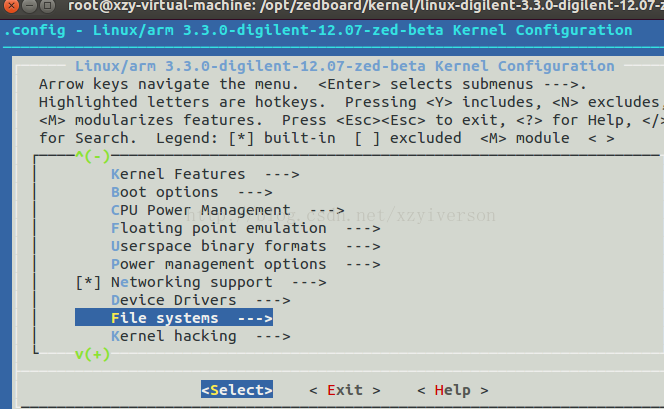
必须在这个文件的基础上进行修改，

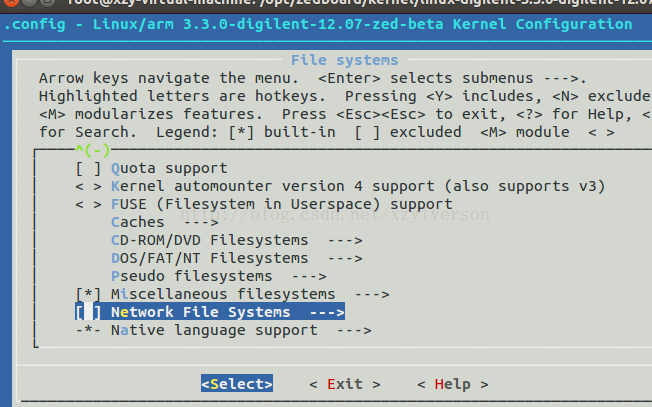
make menuconfig    报错

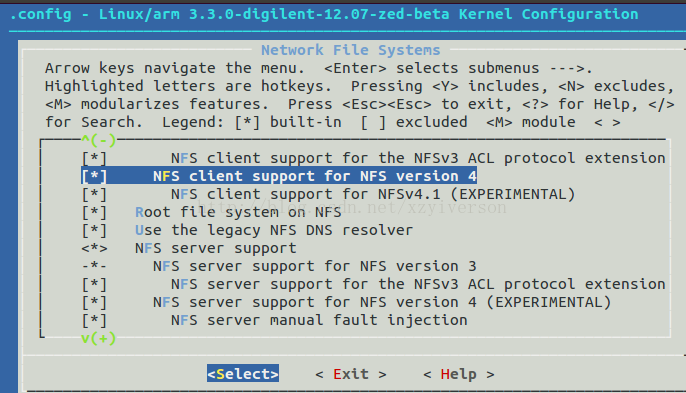


这是因为make menuconfig是要调用libncurses5-dev这个库的，安装这个库，

apt-get install libncurses5-dev  完成后，make menuconfig

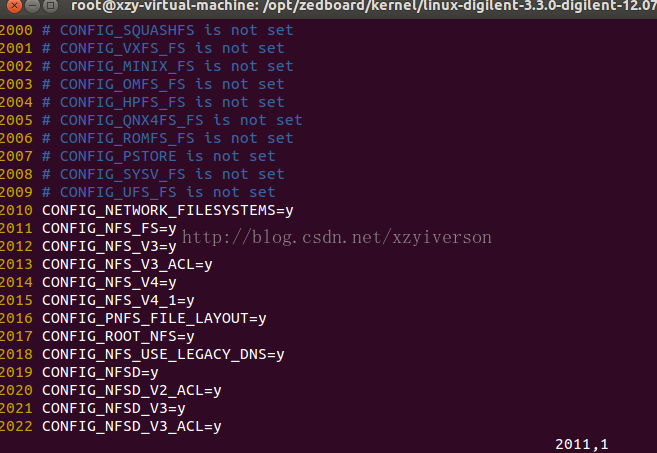






你可以看下.config，他是没有关于NFS的信息的，就是没有开启。

退出保存后  vim .config ，查找CONFIG\_NFS可以看到,这个就是刚刚新增加的。



Make menuconfig所做的修改会保存在.config文件中。

$ make ARCH=arm

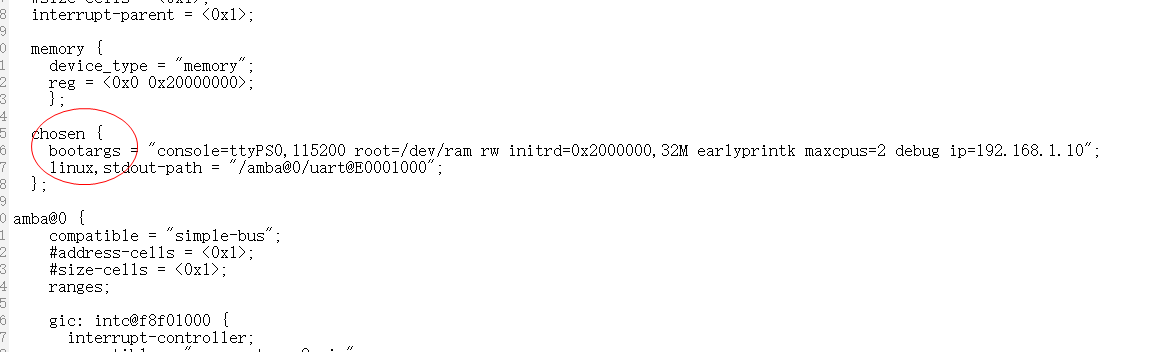
make ARCH=arm CROSS\_COMPILE=arm-xilinx-linux-gnueabi-  UIMAGE\_LOADADDR=0x8000 uImage

最后编译出来的zImage放在：/linux-xlnx/arch/arm/boot/zImage

参考：<http://blog.csdn.net/xzyiverson/article/details/15683379>

## 三．修改uboot传递参数bootargs

bootargs在.dts里面打开dts可以看到



我们要将:

bootargs = "console=ttyPS0,115200 root=/dev/ram rw initrd=0x2000000,32M earlyprintk maxcpus=2 debug ip=192.168.1.10";

修改为：(不能回车，要一整行)

bootargs="console=ttyPS0,115200 root=/dev/nfs rw nfsroot=192.168.199.193:/home/ctq/zynq/\_rootfs  ip=192.168.199.40:192.168.199.193:192.168.199.1:255.255.255.0::eth0:off earlyprintk rootwait devtmpfs.mount=0";

修改好后保存编译设备树：

>/home/ctq/zynq/linux-xlnx/scripts/dtc/dtc -I dts -O dtb -o devicetree.dtb devicetree\_ramdisk.dts

参考：<http://blog.163.com/thinki_cao/blog/static/83944875201439112133825/>

四：最后还要按照文档自己做一个可用的文件系统，偷个懒用ramdisk里面的也行

完。。。。。

---------------------------------------------------------------------------------------------------------------------------------

随笔：

遇到无法生效的问题 ，修改文件系统的/etc/init.d/rcS:

    mkdir -p /ctq\_pc\_realtime\_file

    mount -t nfs -o nolock,vers=2 192.168.199.193:/home/ctq/zynq/\_rootfs /ctq\_pc\_realtime\_file

    echo "++ mount \_rootfs in ctq\_pc"

重新解包、挂载文件系统：

gunzip ramdisk.image.gz

chmod u+rwx ramdisk.image

mkdir tmp\_mnt/

sudo mount -o loop ramdisk.image tmp\_mnt/

cd tmp\_mnt/

修改目标文件---

重新打包文件系统

sudo umount tmp\_mnt/

gzip ramdisk.image

加上uboot头：

mkimage -A arm -T ramdisk -C gzip -d ramdisk.image.gz uramdisk.image.gz

---------------------------------------------------------------------------------------------------------------------------------

cp /home/ctq/CodeSourcery/Sourcery\_CodeBench\_Lite\_for\_Xilinx\_GNU\_Linux/arm-xilinx-linux-gnueabi/libc/lib/\* lib

cp /home/ctq/CodeSourcery/Sourcery\_CodeBench\_Lite\_for\_Xilinx\_GNU\_Linux/arm-xilinx-linux-gnueabi/libc/usr/lib/\* usr/lib

uboot里设置IP地址方法

setenv serverip 192.168.199.1

setenv ipaddr 192.168.199.40

setenv netmask 255.255.255.0

setenv bootargs root=/dev/nfs nfsroot=192.168.199.193:/home/ctq/zynq/\_rootfs/ ,tcp ip=192.168.199.40:192.168.199.193:192.168.199.1:255.255.255.0:embedded:eth0:off console=ttySAC0

## 工程所有文件备份：

