***Croos\_compiling OpenSSH***

**INDEX**

**1. Prerequisites**

**1. OpenSSH**

**1.1. zlib**

**1.2. openssl**

**1.3. openssh**

**Building Pre-requesites:**

**```````````````````````````````**

[*root@linux*](mailto:root@linux)*: mkdir -p /usr/local/{mini2440,mini2440/openssh}*

[root@linux](mailto:root@linux): *mkdir -p /embedd/mini2440/openssh*

[root@linux](mailto:root@linux): cd */embedd/mini2440/openssh*

[*root@linux*](mailto:root@linux)*: export INST\_DIR=/usr/local/mini2440/openssh*

[*root@linux*](mailto:root@linux)*: export PATH=$PATH:/path/2/mini2440/crosscompile/toolchian/ourput/host/usr/bin*

**1. Openssh**

**1.1. zlib:**

[root@linux](mailto:root@linux): *wget* [*http://zlib.net/zlib-1.2.8.tar.gz*](http://zlib.net/zlib-1.2.8.tar.gz)

[root@linux](mailto:root@linux): *tar xvf zlib-1.2.8.tar.gz*

[root@linux](mailto:root@linux): *cd zlib-x.x.x*

[root@linux](mailto:root@linux): *prefix=/usr CC=arm-linux-gcc CFLAGS="-O4" ./configure*

[root@linux](mailto:root@linux): *make && make DESTDIR=/usr/local/mini2440/openssh install*

**1.2. openssl:**

[root@linux](mailto:root@linux): wget <https://www.openssl.org/source/openssl-1.0.1h.tar.gz>

[root@linux](mailto:root@linux): tar xvf openssl-1.0.1h.tar.gz

[root@linux](mailto:root@linux): cd openssl-1.0.1h

[root@linux](mailto:root@linux): ./Configure dist --prefix=/usr

[root@linux](mailto:root@linux): make CC="arm-linux-gcc" AR="arm-linux-ar r" RANLIB="arm-linux-ranlib"

[root@linux](mailto:root@linux): make DESTDIR=/usr/local/mini2440/openssh install

It will install in /usr/local/mini2440/openssh/usr

now copy every thing in openssh to your rootfs.

**1.3. openssh:**

[root@linux](mailto:root@linux): wget <ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/portable/openssh-6.6p1.tar.gz>

[root@linux](mailto:root@linux): tar xvf openssh-6.6p1.tar.gz

[root@linux](mailto:root@linux): cd openssh-6.6p1

[root@linux](mailto:root@linux): ./configure --host=arm-linux --prefix=/usr CC="arm-linux-gcc -L/usr/local/mini2440/openssh/usr/lib -I/usr/local/mini2440/openssh/usr/include" AR="arm-linux-ar"

[root@linux](mailto:root@linux): vi Makefile

STRIP\_OPT have to be removed because it is impossible to use /usr/bin/install on x86-64 to strip binaries on ARM. The rule check-config will run the generated sshd on the host, so we have to avoid that.

modify STRIP\_OPT=-s as STRIP\_OPT=

remove "check-config" dependency in target "install"

[root@linux](mailto:root@linux): make -j4

[root@linux](mailto:root@linux): make DESTDIR=/usr/local/mini2440/openssh install

[root@linux](mailto:root@linux): vi $Rootfs/etc/inittab

# executing rcS script

null::sysinit:/etc/init.d/rcS

# sshd respawing

::respawn:/sbin/sshd -D

# enabling console respawn

ttySAC0::respawn:-/bin/sh

you shud mount devpts /dev/pts in rcS

[root@linux](mailto:root@linux): vi $Rootfs/etc/init.d/rcS

#!/bin/sh

mount -t proc null /proc

mount -t sysfs null /sys

mount -t devtmpfs null /dev

mkdir -p /dev/mq

mount -t mqueue null /mq

# These below 2 steps are necessary for ssh connection to HOST

mkdir -p /dev/pts

mount -t devpts devpts /dev/pts

echo "/sbin/mdev" > /proc/sys/kernel/hotplug

/sbin/mdev -s

export PATH=\

/bin:\

/sbin:\

/usr/bin:\

/usr/sbin:\

/usr/local/bin

#enabling brightness

echo 1 > /sys/devices/platform/s3c24xx\_led.5/leds/backlight/brightness

#overwrite the timeout for console blank to 0. default val 600sec (10M).

chmod +w /sys/module/kernel/parameters/consoleblank

echo 0 > /sys/module/kernel/parameters/consoleblank

exec /etc/init.d/openssh.sh

[root@linux](mailto:root@linux): vim $Rootfs/etc/init.d/openssh.sh

#!/bin/sh

#SSHP=/usr/local/mini2440/openssh/etc

SSHP=/etc

touch /etc/group

touch /etc/passwd

mkdir -p /root /home

addgroup -S root

#addgroup -S sshd

adduser -S root -G root

echo "Enter Passwd for root: "

passwd

ssh-keygen &&

ssh-keygen -t rsa1 -f $SSHP/ssh\_host\_key -N "" &&

ssh-keygen -t dsa -f $SSHP/ssh\_host\_dsa\_key -N "" &&

ssh-keygen -t rsa -f $SSHP/ssh\_host\_rsa\_key -N "" &&

ssh-keygen -t ecdsa -f $SSHP/ssh\_host\_ecdsa\_key -N "" &&

ssh-keygen -t ed25519 -f $SSHP/ssh\_host\_ed25519\_key -N ""

[root@linux](mailto:root@linux):