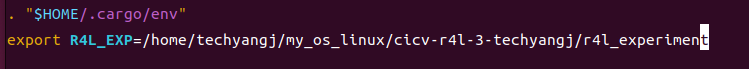
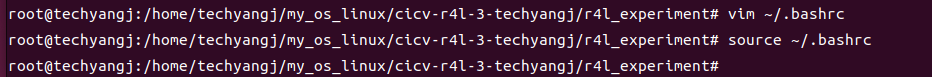
export R4L\_EXP=/path/to/your/r4l\_ experiment/

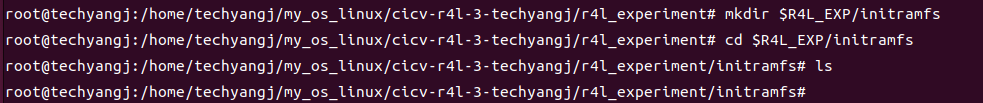




创建镜像

mkdir $R4L\_EXP/initramfs

cd $R4L\_EXP/initramfs



# Create necessary directories

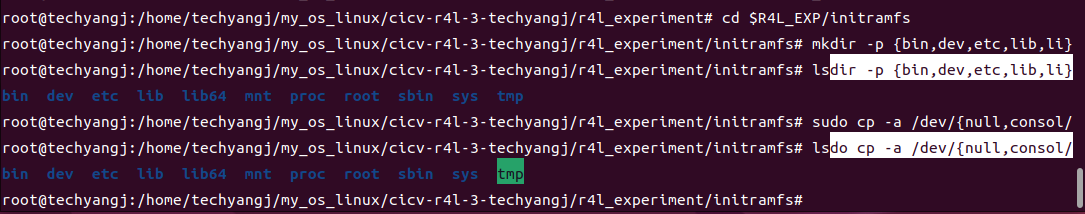
mkdir -p {bin,dev,etc,lib,lib64,mnt,proc,root,sbin,sys,tmp}

# Set Permission

chmod 1777 tmp

# Copy necessary device files from host, root privilege maybe needed.

sudo cp -a /dev/{null,console,tty,ttyS0} dev/

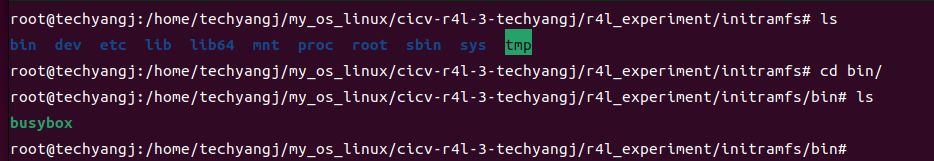


**将之前静态编译的busybox拷贝到initramf/bin下**

cd $R4L\_EXP/initramfs

cp /path/to/your/busybox-1.36.1/busybox  ./bin/

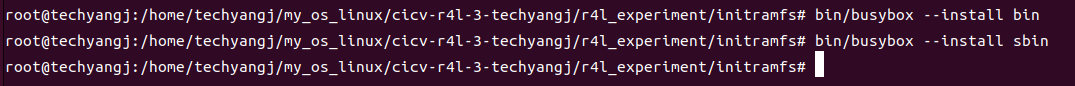
chmod +x bin/busybox



# Install busybox

bin/busybox --install bin

bin/busybox --install sbin



**编写init脚本**

内核以 PID 1 进程的形式执行启动脚本。init 进程负责正常启动系统的其他部分，并设置必要的配置。这些设置包括挂载文件系统、为以太网设备分配 IP 地址等。

cd $R4L\_EXP/initramfs

cat << EOF > init

#!/bin/busybox sh

# Mount the /proc and /sys filesystems.

mount -t proc none /proc

mount -t sysfs none /sys

# Boot real things.

# NIC up

ip link set eth0 up

ip addr add 10.0.2.15/24 dev eth0

ip link set lo up

# Wait for NIC ready

sleep 0.5

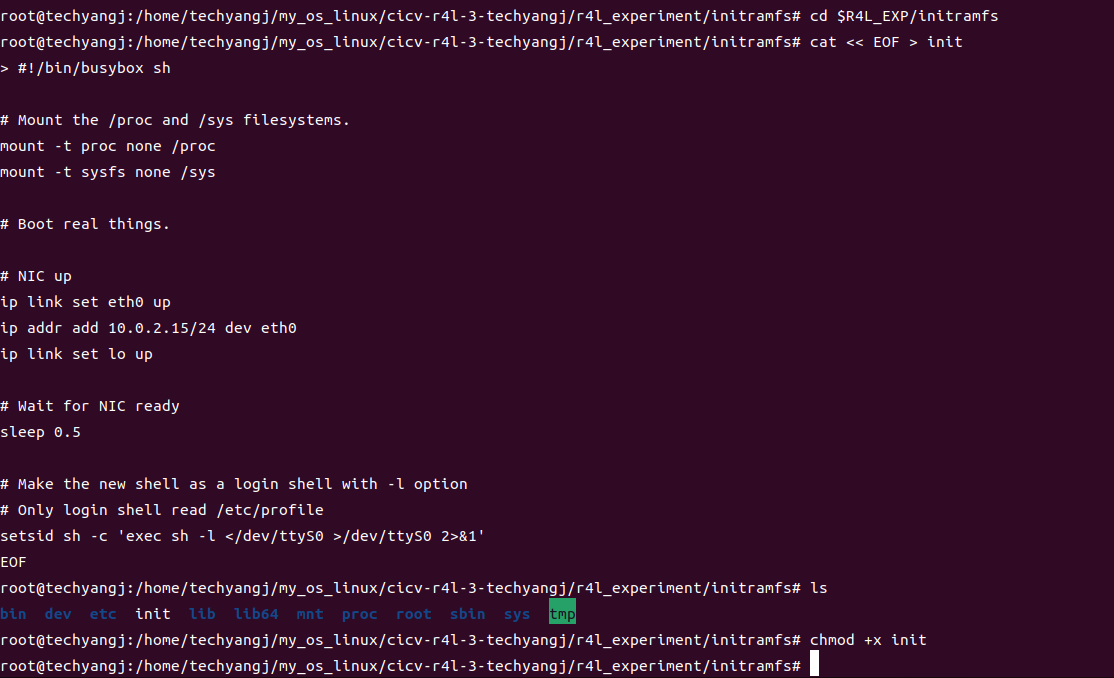
# Make the new shell as a login shell with -l option

# Only login shell read /etc/profile

setsid sh -c 'exec sh -l </dev/ttyS0 >/dev/ttyS0 2>&1'

EOF

chmod +x init



**更多设置**

cd $R4L\_EXP/initramfs

# name resolve

cat << EOF > etc/hosts

127.0.0.1    localhost

10.0.2.2     host\_machine

EOF

# common alias

cat << EOF > etc/profile

alias ll='ls -l'

EOF

# busybox saves password in /etc/passwd directly, no /etc/shadow is needed.

cat << EOF > etc/passwd

root:x:0:0:root:/root:/bin/bash

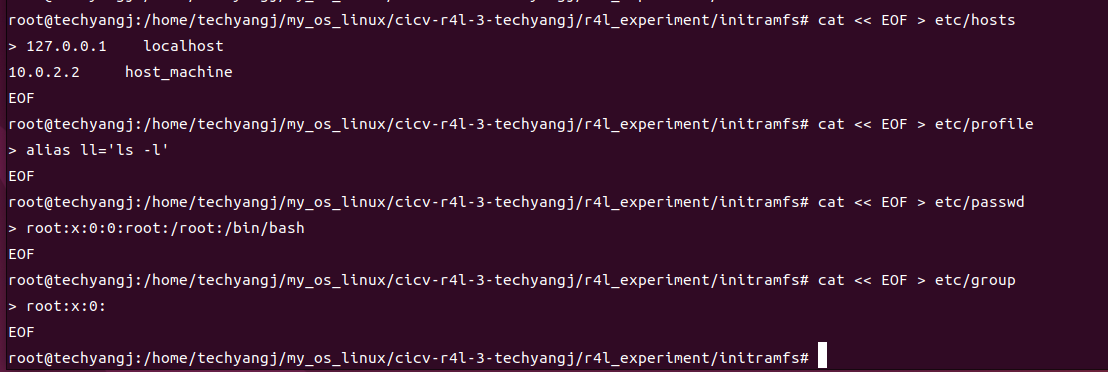
EOF

# group file

cat << EOF > etc/group

root:x:0:

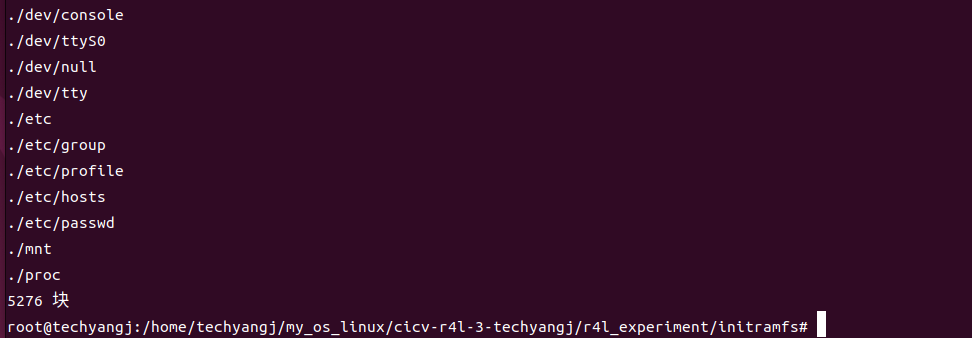
EOF



**构建initramfs镜像**

cd $R4L\_EXP/initramfs

find . -print0 | cpio --null -ov --format=newc | gzip -9 > ../initramfs.cpio.gz



**通过boot.sh脚本启动**

cd $R4L\_EXP

# 以下是boot.sh的内容：

#!/bin/sh

kernel\_image="../linux/arch/x86/boot/bzImage"

qemu-system-x86\_64 \

-kernel $kernel\_image \

-append "console=ttyS0" \

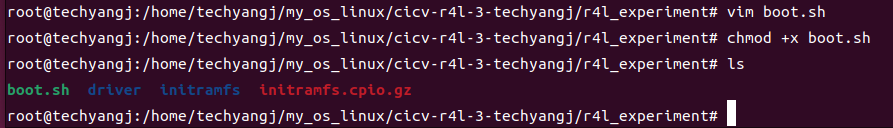
-initrd ./initramfs.cpio.gz \

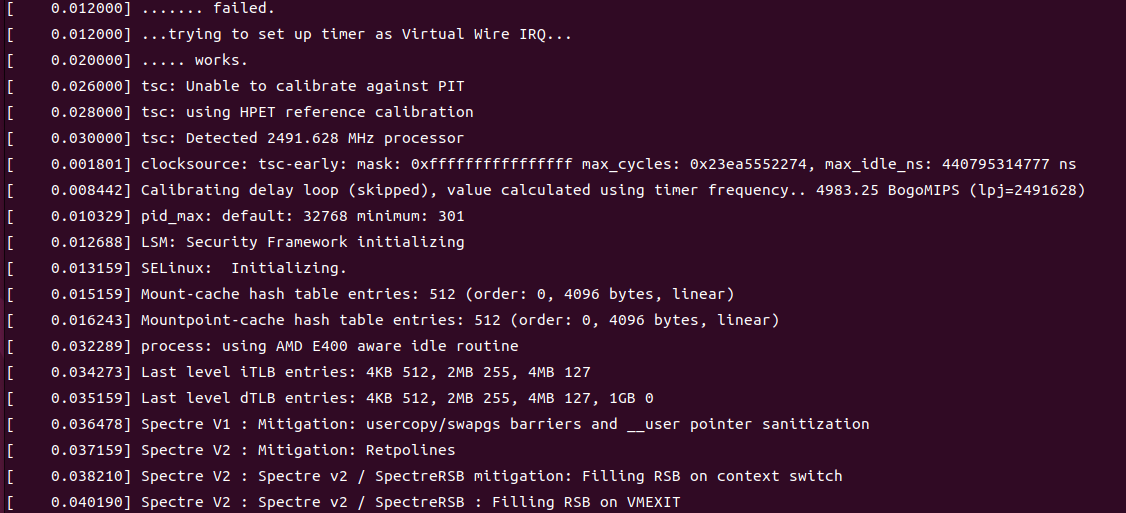
-nographic

# 然后执行以下命令启动

chmod +x boot.sh

./boot.sh  # Press <C-A> x to terminate QEMU.





### 支持NFS

**在主机上设置NFS服务器**

sudo apt-get install nfs-kernel-server

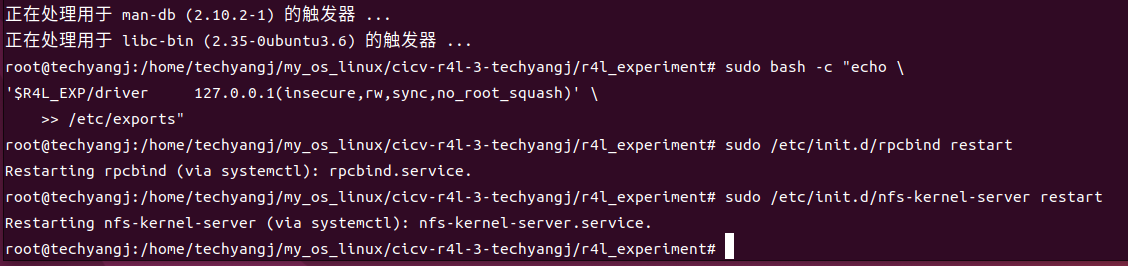
sudo bash -c "echo \

'$R4L\_EXP/driver     127.0.0.1(insecure,rw,sync,no\_root\_squash)' \

    >> /etc/exports"

sudo /etc/init.d/rpcbind restart

sudo /etc/init.d/nfs-kernel-server restart



**在方式2中init中脚本加入，实现自动挂载**

# 2. Add this line in init script. Put it just after the line of sleep 0.5.

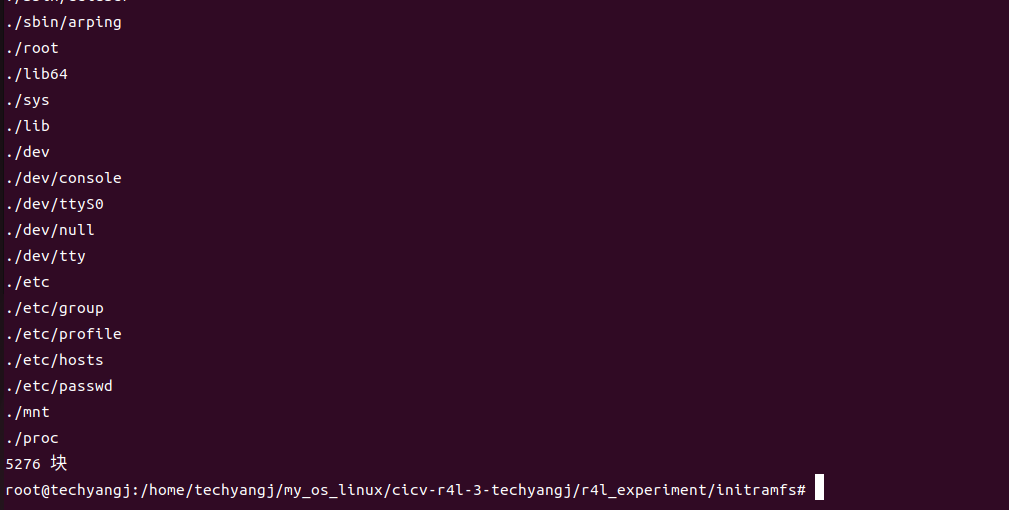
mount -t nfs -o nolock host\_machine:/home/techyangj/my\_os\_linux/cicv-r4l-3-techyangj/r4l\_experiment/ /mnt

# 然后rebuild initramfs

cd $R4L\_EXP/initramfs

find . -print0 | cpio --null -ov --format=newc | gzip -9 > ../initramfs.cpio.gz





### 支持telnet server

**首先设置 pts device node**

cd $R4L\_EXP/initramfs

mkdir dev/pts

mknod -m 666 dev/ptmx c 5 2

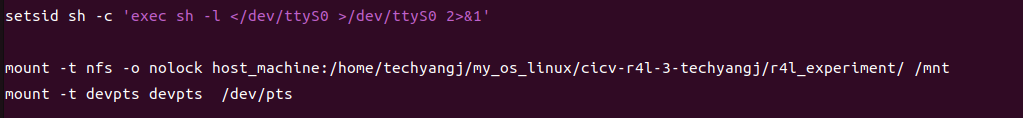
# 同样在init脚本中设置自动挂载，在NFS设置后面加入

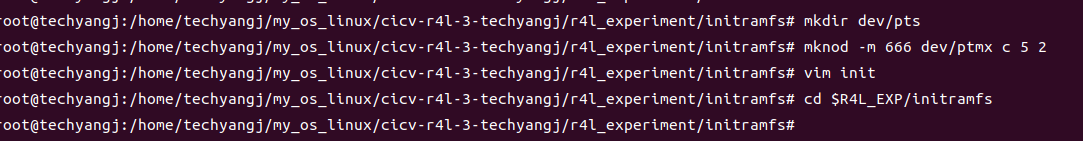
mount -t devpts devpts  /dev/pts

# 然后rebuild initramfs

cd $R4L\_EXP/initramfs

find . -print0 | cpio --null -ov --format=newc | gzip -9 > ../initramfs.cpio.gz



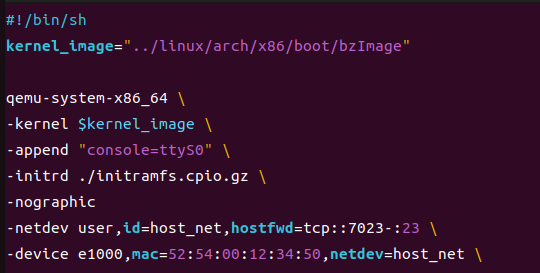




**在boot.sh中加入一下qemu启动参数：**

-netdev user,id=host\_net,hostfwd=tcp::7023-:23 \

-device e1000,mac=52:54:00:12:34:50,netdev=host\_net \



**开启telnet server：**

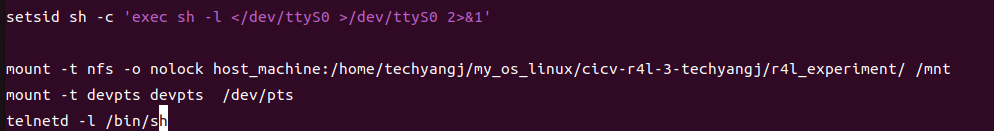
# 同样在init脚本中设置自动启动，在telnetserver设置后面加入

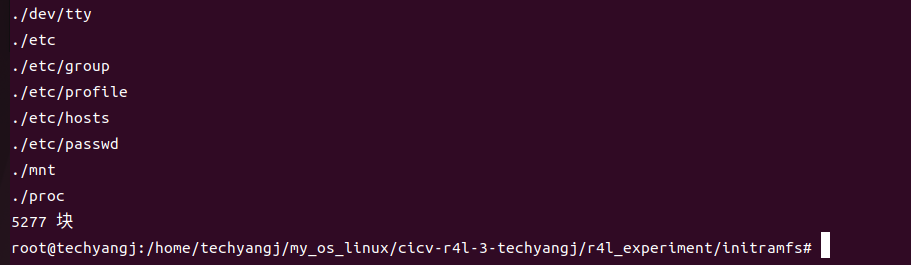
telnetd -l /bin/sh

# 然后rebuild initramfs

cd $R4L\_EXP/initramfs

find . -print0 | cpio --null -ov --format=newc | gzip -9 > ../initramfs.cpio.gz





**在本机通过telnet server连接qemu控制台**

telnet localhost 7023