记录一下利用vmware+gdb串口调试linux内核的步骤

注意 我们把主机（服务器）称为HOST，vm上的linux称为TARGET

1. 安装vmware，在vmware上安装linux

略……

安装完成后在xxxx.vmx文件最后加上一行：

TARGET如果是32位机:

*debugStub.listen.guest32="1"*

TARGET如果是64位机:

*debugStub.listen.guest64="1"*

1. 为虚拟机添加串口

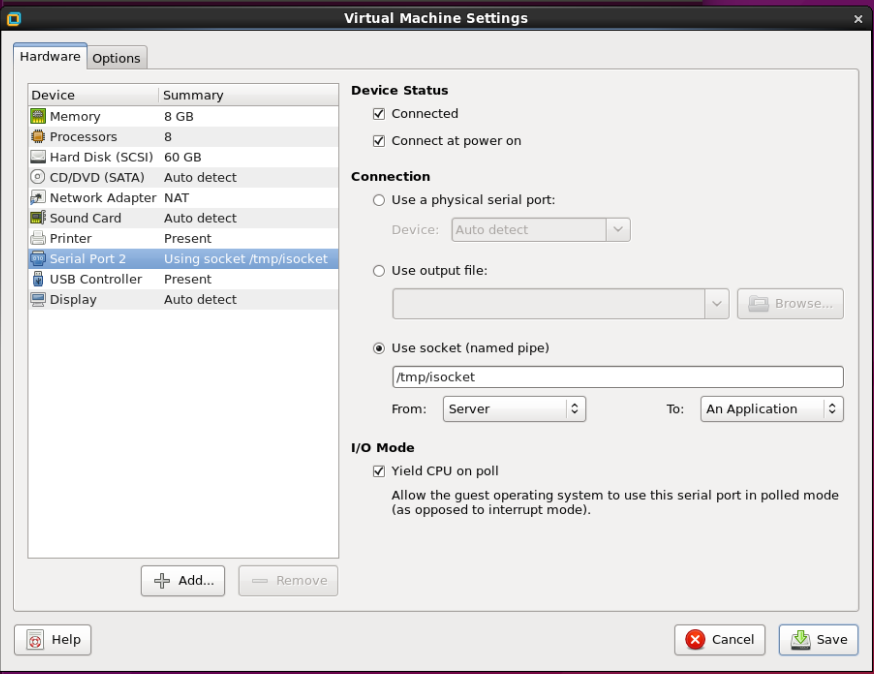
（1）添加串口

注意：虚拟机的网络连接选项要设置为NAT方式（其余方式未测试是否可行，可能需要多余的配置）

虚拟机需要在关机状态下添加串口。

Vmware->settings->hardware->add->serial port->output to socket->socket(named pipe): /tmp/isocket from:server to: an application √connect at power on->√ yield CPU on poll ->save

如图所示：



启动虚拟机 完成安装

（2）测试

HOST需要安装两个软件：socat和telnet

如果没有权限，如何安装rpm包文件：[centos非root用户(普通用户)安装rpm包](http://blog.csdn.net/sdcxyz/article/details/47094221)

简单来说就是：解压、修改环境变量 让bash能找到可执行文件。

注意：Socat最好不要安装最新版本的，它依赖于GLIBC\_2.14（大概是C运行库），服务器上的只到GLIBC\_2.12版本，我用的是[socat-1.7.1.3-1.el6.rf.x86\_64.rpm](https://centos.pkgs.org/6/repoforge-x86_64/socat-1.7.1.3-1.el6.rf.x86_64.rpm.html)

telnet：<http://vault.centos.org/6.3/os/x86_64/Packages/> Ctrl+F 搜索telnet  [telnet-0.17-47.el6.x86\_64.rpm](http://vault.centos.org/6.3/os/x86_64/Packages/telnet-0.17-47.el6.x86_64.rpm) [telnet-server-0.17-47.el6.x86\_64.rpm](http://vault.centos.org/6.3/os/x86_64/Packages/telnet-server-0.17-47.el6.x86_64.rpm)

测试串口：

//HOST：

**$ socat /tmp/isocket tcp-listen:9001 &**

**$ telnet 127.0.0.1 9001**

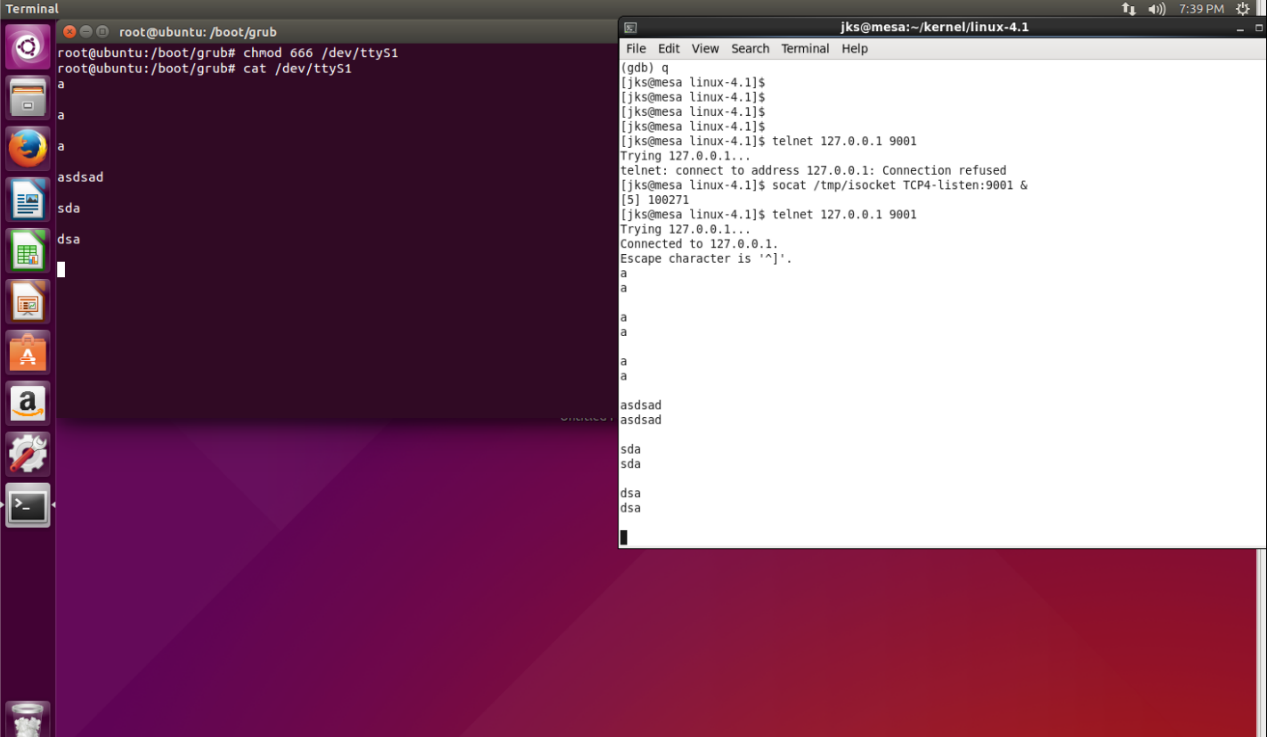
//此时已经建立连接

//TARGET:

# chmod 222 /dev/ttyS1 //ttyS1是虚拟机默认的串口

# cat /dev/ttyS1

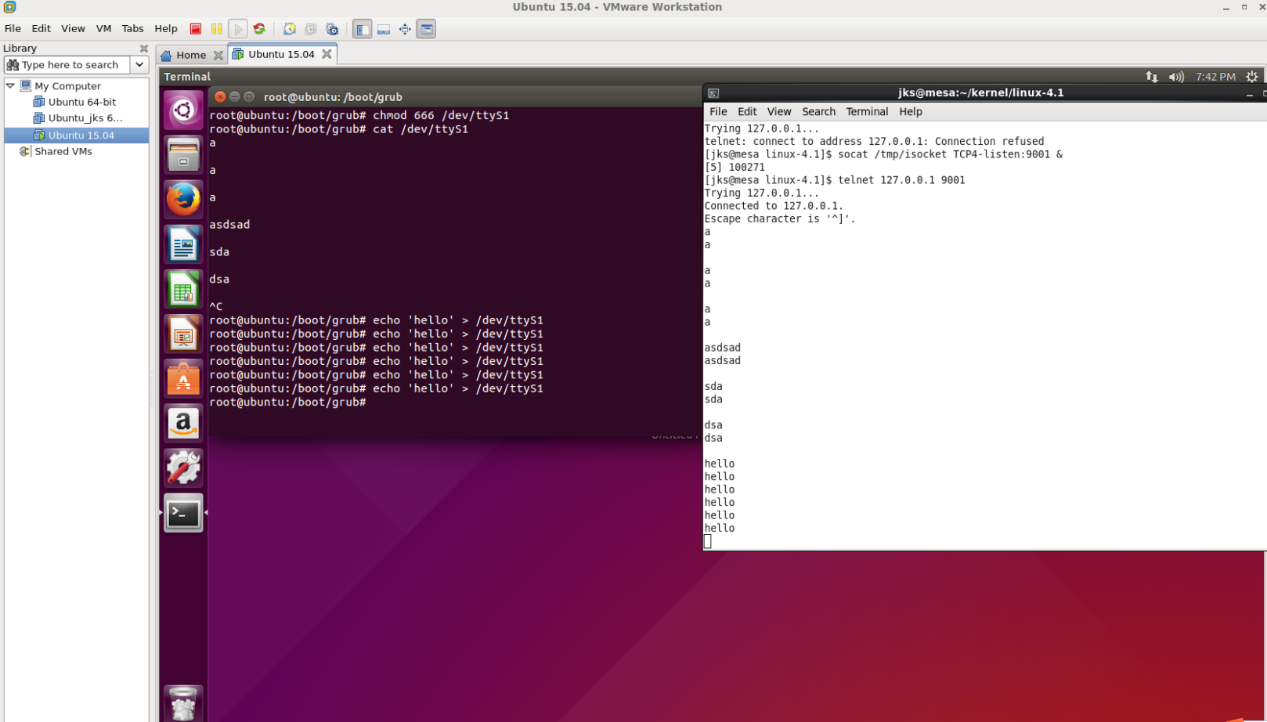
#此时在HOST上输入信息，TARGET可以通过ttyS1可以接受到（HOST>>TARGET）



//TARGET to HOST

#echo ‘hello’ > /dev/ttyS1

//HOST的命令行可以收到hello



telnet输入ctrl + ]结束

解释一下：socat和telnet命令建立连接，HOST通过/tmp/isocket 这个文件收发数据 ，TARGET通过/dev/ttyS1收发数据。

如果串口的收发都没有问题，测试成功，可以继续进行。

1. 编译linux内核

Gdb调试的时候 需要vmlinux文件（实际的内核文件），有两种方法获取

1. 提取内核

Extract-vmlinux是一个bash脚本，可以把现有系统的vmlinux提取出来

<https://github.com/torvalds/linux/blob/master/scripts/extract-vmlinux>

（复制一下，chmod改成可执行文件）

# ./extract-vmlinux /boot/vmlinuz-xxxx-generic > vmlinux

1. 完全重新编译

下载内核源码解压进入源码目录 -> make menuconfig(对内核进行配置) -> make -j4 -> make bzImage->make modules -> make modules\_install -> make install -> reboot

重启之后uname -a 查看内核信息

方法1的vmlinux 尺寸比重新编译的小一些，应该可以用（我没用）。

我们想要的是vmlinux文件，需要把vmlinux文件在HOST和TARGET放到相同路径下，如/home/jks/kernel/linux-x.x/ (很有可能是玄学，未尝试不同路径是否可行，理论上可以)。

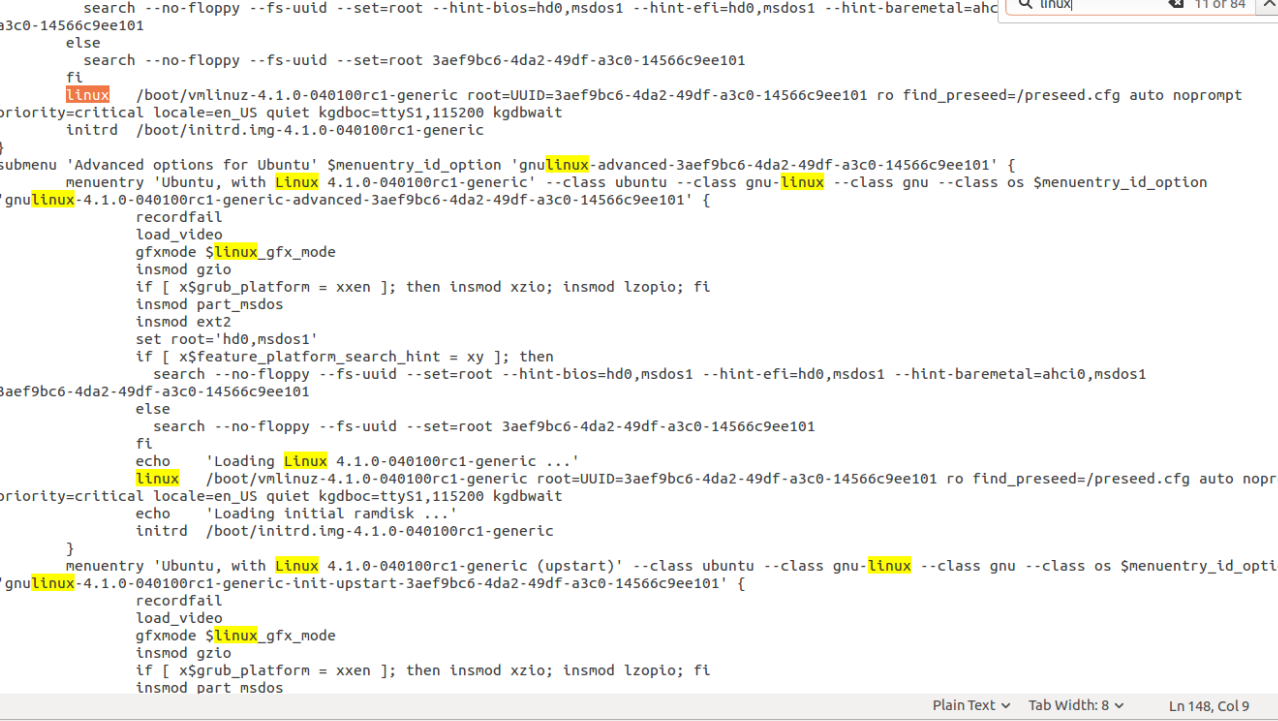
1. GDB调试

首先在TARGET的/boot/grub/grub.cfg中，在linux命令下增加kgdboc和kgdbwait参数，实际打开grub.cfg文件可能有多个linux命令（后面带着内核版本参数），那我们就在每个后边都加上：

kgdboc=ttyS1,115200 kgdbwait

实测 kgdbwait参数是在TARGET启动时等待建立连接，不加的话TARGET就直接启动了，不知道怎么建立连接办法……所以一定要加上。

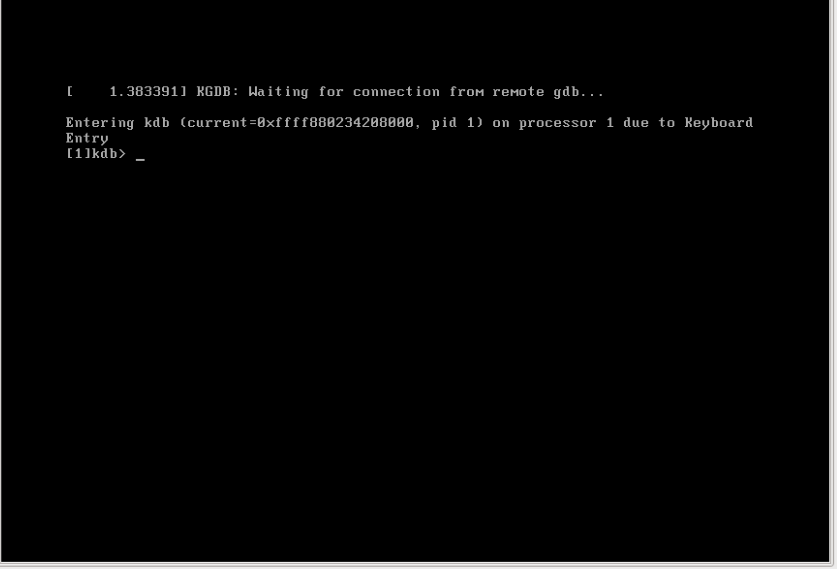
如图所示：



改完之后，

重启TARGET

TARGET会停在



//HOST：

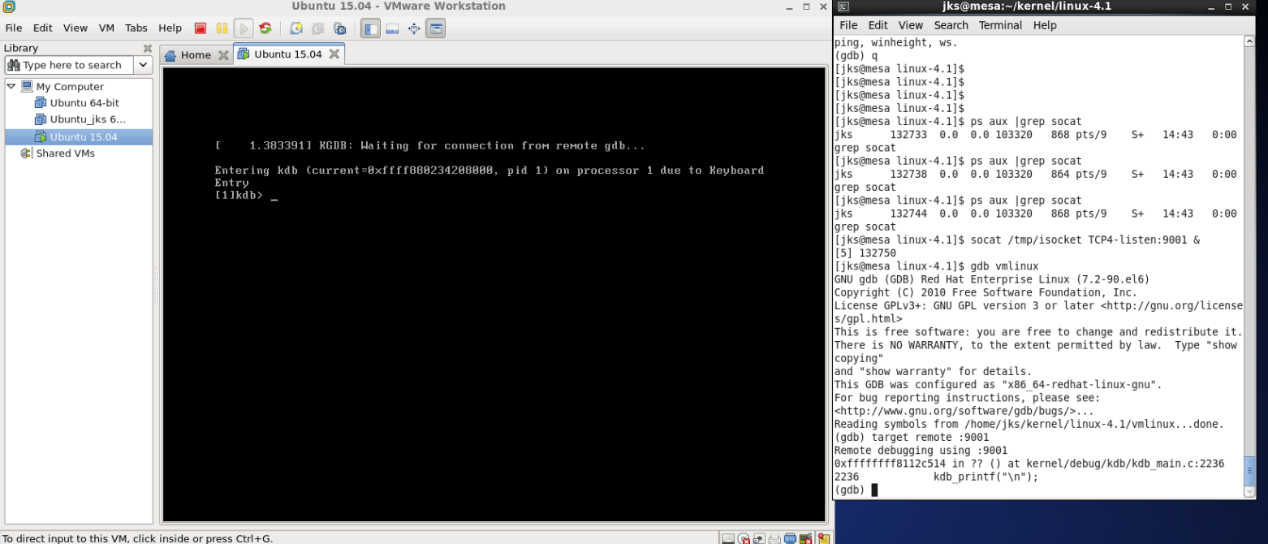
**$ socat /tmp/isocket tcp-listen:9001 &**

$ cd /vmlinux所在目录

$ gdb ./vmlinux

(gdb) target remote :9001

成功的话是这样：



gdb

n //单步调试

c(continue) //TARGET自己执行启动 HOST失去所有权

TARGET中，将所有权交还HOST：

# echo g > /proc/sysrq-trigger

目前为止，就会这么多