**《嵌入式系统和通信专题实践》**

### 实验设备分配

本课程实验中，2人一组共用一套设备，请事先确定好队友，每组派一个人在实验开始时排队领取设备。实验地点：翡翠科教楼C503。

实验报告每人交一份，封面必须注明同组队友的学号与姓名。实验截图可以小组共用，实验报告的其他内容不能雷同。

前16个课时建议使用实验室的Fedora虚拟机（用户名为Knight，密码为knight）、后16个课时需要自带笔记本电脑（一组带1台即可）。

### 关于验收

每个实验完成后，请在记录表上登记申请验收，老师将按照登记顺序检查（完成实验的时间会影响验收评分）

### 实验报告提交

实验报告以电子版形式提交，必须为pdf格式（文件名为学号-姓名-嵌入式系统实验报告n，例如：2018001234-张三-嵌入式系统实验报告1），实验结束后一周内发送给班长（若班长未选课，请协商交给其他班干部），由班长统一发送到pengminhfut@qq.com。为防止被邮件服务器当作垃圾邮件，请务必将邮件标题设为“2018级通信工程X班-嵌入式系统实验报告n”。

### 实验报告要求

实验报告包含封面（姓名，学号，课程名称，指导老师，队友）、正文。

正文必须有关键步骤的截图（截图必须清晰完整——用A4纸打印的情况下能够看清图中的文字和用户名等信息）。

请对每个截图编号，在截图中用绿色方框标记出关键内容，并在图片之前用文字说明这个图是什么目的（没有文字说明或未在图片中标记关键信息的，实验报告将会扣分）。

截图必须含有自己（或队友）姓名学号等信息。正文必须有必要的文字说明。

实验报告必须按照规范命名，学号-姓名-嵌入式系统实验报告n.pdf，必须提交pdf格式的文件。

实验报告严禁抄袭，一旦发现抄袭现象（例如：使用他人的截图，或者使用虚假截图），做0分处理。

注：PC界面的截图可使用键盘的PrintSreen键；嵌入式设备的显示屏截图可使用 snapshot 命令，例如snapshot pic1.png可以对开发板当前的屏幕显示内容截图，并保存为pic1.png。

目录

[实验一、嵌入式Linux开发环境（4课时） 3](#_Toc7726489)

[实验二、交叉编译（4课时） 12](#_Toc7726490)

[实验三、UDP网络编程（4课时） 17](#_Toc7726491)

[实验四、shell编程（4课时） 19](#_Toc7726492)

[实验五、Linux数据库编程（8课时） 21](#_Toc7726493)

[实验六、嵌入式聊天软件设计（8课时） 23](#_Toc7726494)

[实验七、嵌入式聊天软件设计2（附加） 25](#_Toc7726495)

# 实验一、嵌入式Linux开发环境（4课时）

一、准备工作

后续实验大部分都需要使用串口，由于实验室PC没有串口，需要使用usb转串口线（实验室提供）。

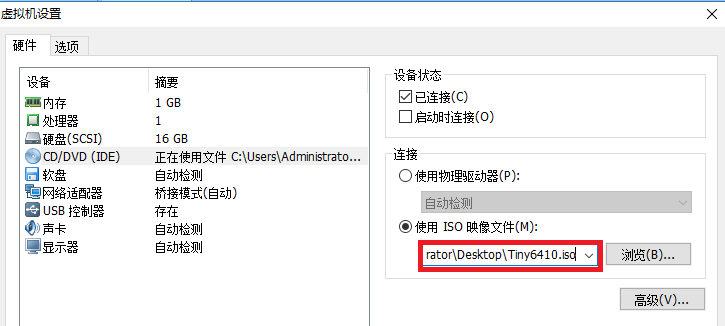
将转接线的usb口连接电脑，串口端连接开发板的右侧串口；接通电源、用网线连接开发板与PC。



二、通过串口控制开发板

实验室PC安装了VMWare以及Fedora虚拟机，后续实验将使用该虚拟机。

配置虚拟机时，“CD/DVD”请选择使用老师提供的Tiny6410.iso文件；网络连接需要使用“桥接模式”,桥接的对象必须使用有线网卡（需要在虚拟机的网络配置里面完成相关配置）。



为了在虚拟机的Linux中使用串口，需要将串口添加到虚拟机中，有2种添加方式：将连接串口转接线的usb直接添加到虚拟机、或将串口添加到到虚拟机。可以根据自己的喜好选择其中一种方法（推荐方法1）。

**方法1）将usb添加到虚拟机**

确保虚拟机已经添加了usb控制器。并在虚拟机状态栏的右下角将usb转串口线“连接”到Linux虚拟机里。具体方法为，先在windows中关闭占用该usb转接线的软件（例如超级终端，如果未打开这类软件，则无需处理），然后右键usb图标（若有多个usb，请选择usb转串口转接线连接的usb——鼠标指向该usb，会有“Prolific USB-Serial Controller D”之类的提示），选择“连接”。



**方法2）将串口添加到虚拟机**

这种方法相对复杂一些，不推荐使用。

由于物理机并没有串口，为了将串口添加到虚拟机，需要首先在物理机中添加串口。物理机添加串口的具体方法为：

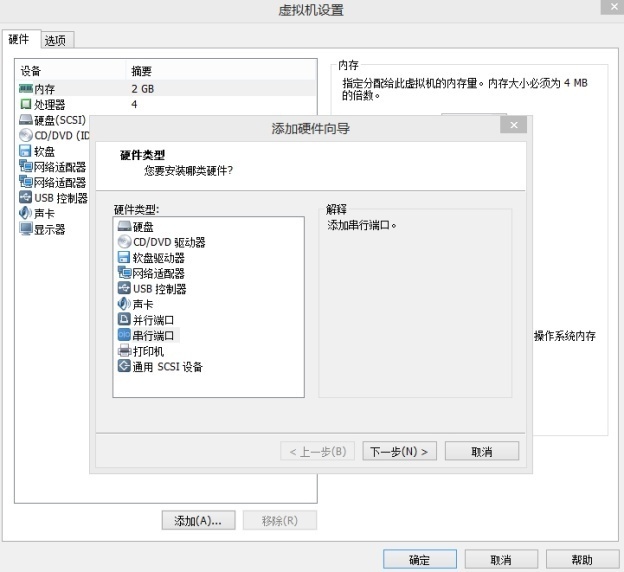
安装完usb转串口驱动并接上usb转串口线后，在设备管理器中查看串口序号（如下图）：

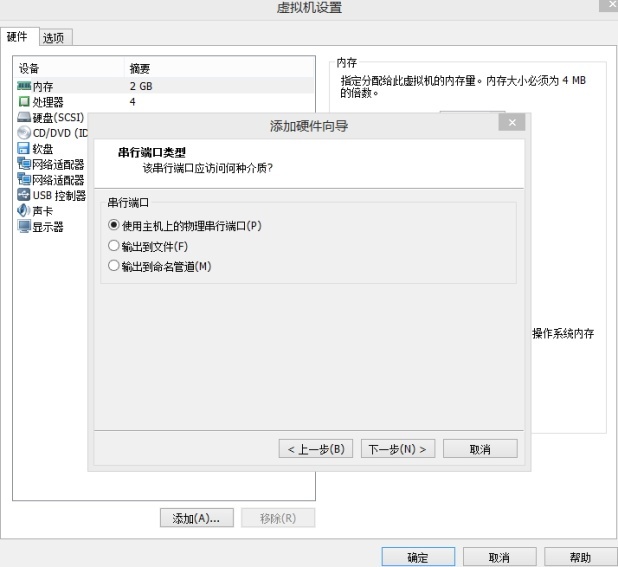


记住这个序号（这里为COM3），在后面会用到。

在windows下用串口助手（或超级终端等类似工具，课件目录已提供该工具）连接该串口（如com3），配置串口参数为：波特率115200，8 位数据位，1 位停止位，没有流控和校验。

物理机的串口（如COM3）设置成功后，就可以在虚拟机中添加串口了，选择“使用主机上的物理串行端口”，选用前面步骤中确认可正常工作的串口号（如COM3）。







启动虚拟机后，在VMWare的右下角连接串口。具体方法为，先在windows中解除对该串口的占用（关闭超级终端），然后右键VMWare软件状态栏的串口图标，点击“连接”。



使用这种方法，虚拟机Linux的串口通常为ttyS0（不同Linux系统会有不同）。

注：若使用自带的电脑，可能存在usb转串口驱动缺失的情况，需要安装相应驱动，然后使用locate pl2303找到驱动所在位置后，通过modprobe usbserial和modprobe pl2303加载驱动模块。

三、配置minicom

使用dmesg命令查看系统的启动信息，如果看到最后几行出现“USB Serial Device converter now attached to ttyUSB0”之类的字段，表明串口正常，只需要配置minicom就可以了。

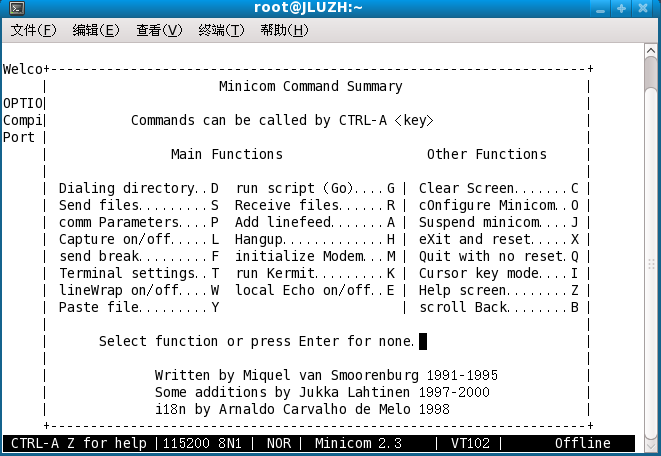
注：若将usb直接添加到虚拟机（即前面的方法1），通常串口为ttyUSB0；若将串口添加到虚拟机（即前面的方法2），通常串口为ttyS0。（不同系统下，串口编号命名有所不同，可通过cat /proc/devices确定存在哪些串口）

在宿主机Linux 系统下的终端中输入（通常需要root权限执行）：

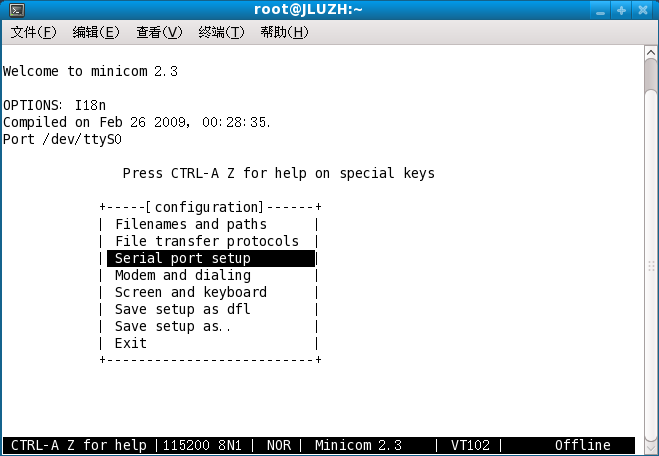
minicom -s

注： Fedora可先su，然后使用minicom；但在ubuntu中应该使用sudo minicom，su后执行minicom在某些版本的ubuntu中会报错。

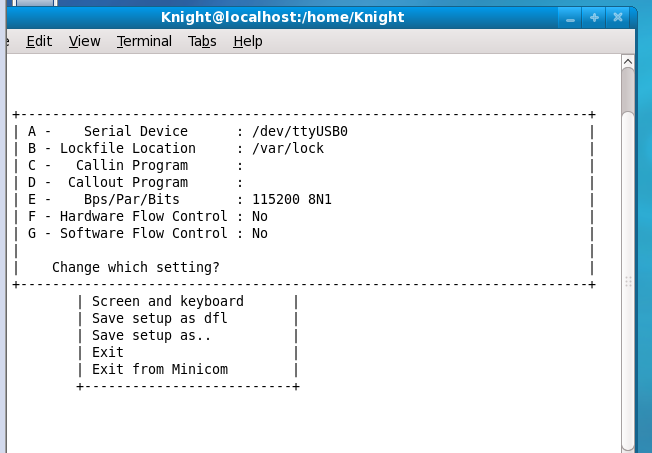
minicom的设置界面如下（若使用minicom -s，会直接进入配置界面；若只输入minicom，需要先按Ctrl+a 键，再按z进入帮助，按提示信息进行如下配置）



按“o”可进入配置界面：



选择**Serial port setup**，将串口配置为：**波特率115200，8 位数据位，1 位停止位，没有流控和校验**。并将其设置存为默认值（**Save setup as dfl）**。



一定要将Serial Device改为正确的串口设备，不同的系统，串口设备不同，通常为/dev/ttyUSB0（若前面使用方法1，或使用双系统）或/dev/ttyS0（若前面使用方法2）。

保存（Save setup as dfl）退出后回到minicom 界面。

配置好minicom后，重新启动开发板（不要直接关闭，按右侧的重启按钮即可），即可看到minicom界面上出现启动信息。

可能出现的错误及解决方案（大部分错误可通过以下方案解决）：

1. 在确定连接了usb转串口线的情况下，若在/dev/下看不到ttyUSB开头的文件，可使用mknod /dev/ttyUSB0 c 188 0添加（也可以为其它序号）。
2. 若存在/dev/ttyUSB0，但minicom不成功，可尝试在另一终端用root身份执行lsusb，然后执行dmesg，看看是否出现了USB串口。然后在minicom对应终端敲回车重试。

四、通过minicom配置开发板

为了使minicom与宿主机顺利通信，需要为它们配置IP，配置IP可通过ifconfig命令实现。

在minicom终端输入：

ifconfig eth0 192.168.0.1**xx**

另外开一个终端，为宿主机配置IP：

ifconfig eth0（或ens33等其他名称） 192.168.0.**yy**

注： **xx、yy**为**合作双方的学号后两位**。例如，2016211511和2016211512合作完成本实验，可将目标板IP配置为192.168.0.111，宿主机IP设置为192.168.0.12；若单人独立完成实验，xx、yy均使用自己的学号后两位。

请将上面两个ifconfig及其执行结果截图放入实验报告中（**图1**、**图2**）并用绿色方框标记出自己学号对应的IP信息（后续所有图片都需要做类似标记）。

配置完成后，需要确保主机和开发板能够ping通。并将ping命令的结果截图，开发板和宿主机两边的ping都需要截图放入实验报告中（**图3、图4**）。

注：若不能ping通，首先需要确认网线是否正常（可将该网线连接两台PC，确认能ping通）；然后检查虚拟机网络接口是否配置为桥接方式，同时，所桥接的物理网卡必须为所使用的有线网卡。

要记住哪个终端为minicom终端（该终端输入的命令将操作开发板）。如果不确定哪个终端为minicom终端，可使用uname -a查看终端是否连接开发板（即minicom终端）。

由于开发板上运行的是嵌入式Linux，因此，其命令与Linux命令完全一样，同学们可以尝试在上面执行一些常用的命令。尤其是使用uname -a、file等命令了解系统和文件信息，并熟悉ls、cd、pwd、df、mount、umount、vi的使用。

五、建立主机开发环境

在我们的实验环境中，宿主机为x86体系处理器，而目标板处理器为SAMSUNG S3C6410。

在宿主机上，我们要建立交叉编译调试的开发环境。环境的建立需要许多的软件模块协同工作，这将是一个比较繁杂的工作。

将Tiny6410.iso添加到Linux虚拟机（先设置虚拟机，在CD/DVD中选择Tiny6410.iso；然后点击虚拟机右下角光盘形状图标，选择“连接”），进入Fedora虚拟机，确保已挂载虚拟光盘。



cd /media

ls

#查看光盘是否已经被挂载。若没有挂载，需要先看是否有/dev/cdrom设备，然后将该设备mount到/mnt/cdrom，mount -t iso9660 -ro loop ...

cd /media/Tiny6410

#进入光盘，通常在/media/Tiny6410或/mnt/cdrom目录，具体命令请根据实际情况执行

… #开发环境配置

准备工作：将光盘Tiny6410目录中的所有文件复制到 Fedora9 某个目录下（如/home/Knight，后面的命令均以该目录为例），然后按步骤执行以下操作。

**Step1** 安装arm-linux-gcc

（仅针对自带电脑的情况；实验室虚拟机正常情况下已完成该操作，需先检查，若已完成，请**不要**重复执行）：

然后进入/home/Knight目录，执行解压命令：

cd /home/Knight

tar xvzf arm-linux-gcc-4.5.1-v6-vfp-20101103.tgz -C /

该命令将把 arm-linux-gcc 安装到/opt/FriendlyARM/toolschain/4.5.1目录。 注意：上述命令可能需要root权限。

**Step2**：把编译器路径加入系统环境变量

（实验室虚拟机正常情况下已完成该操作，需先检查，若已完成，请**不要**重复执行）。

root权限下通过vi或gedit编辑/root/.bashrc。修改最后一行为：

export PATH=$PATH:/opt/FriendlyARM/toolschain/4.5.1/bin

重新登录系统（不必重启机器，开始->logout ，然后重新登录即可），使以上设置生效，在命令行输入 arm-linux-gcc -v，若出现arm-linux-gcc版本信息，说明已配置成功。

**Step3**：解压uboot与linux内核源码

首先创建目录（若已有该目录，则无需安装），后面步骤的所有源代码都会解压安装到此目录中

mkdir -p /opt/FriendlyARM/tiny6410/linux

解压uboot 源代码（请先查看，若已有对应文件，则无需重复操作）

进入/opt/FriendlyARM/tiny6410/linux目录。执行解压：

tar xvzf /home/Knight/u-boot-mini6410-20111018.tar.gz

将创建生成 u-boot-mini6410 目录，里面包含了完整的 U-boot 源代码。

在工作目录/opt/FriendlyARM/tiny6410/linux 中执行：

tar xvzf /home/Knight/linux-2.6.38-20150708.tgz

将创建生成 linux-2.6.38 目录，里面包含了完整的内核源代码。

六、尝试直接通过键盘控制开发板

开发板有usb口，可直接使用usb键盘实现输入（也可使用开发板的虚拟键盘输入）。这样，可以独立操作开发板，无需借助串口。

请尝试通过开发板的触摸屏打开“终端”软件，在终端中熟悉ls等命令。然后输入touch zhangsan和touch lisi命令（zhangsan、lisi为双方姓名），紧接着使用snapshot pic1.png命令对开发板的屏幕截图（图5）。

注：该图片可使用下一步实验的ftp方法下载下来。

七、通过ftp和telnet访问开发板

该部分可以用Windows，也可以用Linux。若使用windows，开始这部分实验之前，请先将windows中有线网卡的IP配置为192.168.0.1yy（yy为第二位同学的学号后两位）

1、ftp访问开发板

无论在 Linux 系统还是 Windows 系统中，一般都自带一个命令行的ftp程序。请使用Windows的ftp命令登录远程主机（开发板）并传递文件，但需要主机（开发板）提供ftp服务和相应的权限。实验室开发板在开机时启动了ftp服务。

接下来，我们将从PC的命令行窗口登录开发板，并与开发板传递文件。

ftp帐号为：plg（密码为plg）。

登录后，首先使用binary命令设置传输模式（非常重要，否则可能导致图片打不开）。

使用put、get命令上传、下载文件。

put 本地文件名 [远程服务器（即开发板）文件名]

get 远程服务器文件名（请下载上一步实验的截图pic1.png） [本地文件名]

注：若不是当前路径的文件，需要加上路径；[]内的内容表示可省略，直接上传或下载到默认路径。下载完成后，可在PC当前目录看到下载的文件；上传完毕后，目标板的/home/plg目录下可看到上传的文件。（需要注意，plg用户需要对下载/上传的目录具有相应权限）

在Windows中将put、get截图（图6、图7）。然后使用bye命令退出ftp。

注：可通过该方式将后续实验中开发板的截图下载到windows；Fedora虚拟机中的截图可通过vmware共享目录、u盘等其它方式拷贝出来。若使用vmware共享目录，需要先在fedora虚拟机的设置->选项->共享文件夹中设置一个windows的目录，这个目录可在Fedora虚拟机中通过/mnt/hgfs目录访问。

2、telnet登录开发板（选作）

在Windows的命令行窗口输入（若想使用Linux登录开发板，则在Linux的终端中输入，本次实验要求用windows，以便与minicom或其它终端区分）“telnet 192.168.0.1xx”，在登录界面，输入plg (密码为plg)进入系统。

通过该界面，可以远程操作开发板。（**后续实验中，若只是为了控制开发板，可通过网线连接后，使用telnet可控制，并不一定要用串口**）

在该界面熟悉命令pwd、uname -a、who，访问完成后，可通过exit命令退出telnet。

**注意**：如果自带电脑，且系统为64位版本，可能需要安装32位支持包（因为iso中提供的是32位版本的arm-linux-gcc）。Ubuntu下可使用如下命令：

sudo apt-get install libx32z1-dev（低版本Ubuntu为libc6:386或ia32-libs）

**说明：后续所有实验都依赖于本实验，在每次实验前都需要完成类似的操作。**

# 实验二、交叉编译（4课时）

一、准备工作

在进行本实验前，需要确保实验一的嵌入式开发环境已经配置好（实验一的五、六、七部分无需重复）。

二、配置NFS 服务

在NFS 服务中，宿主机（Servers）是被挂载（mount）端，为了远端目标板（Clients）可以访问主机的文件，我们需要主机配置两方面内容：打开NFS 服务，允许“指定用户”使用。

关闭防火墙和iptables服务：

* 在ubuntu中，可通过ufw disable关闭防火墙。
* 在CentOS/Fedora中，可通过setup命令，选择System services，将nfs一项选中（出现[\*]表示选中），并去掉ipchains 和iptables两项服务（即去掉它们前面的\*号）；然后在Firewall configuration中，选中 Disable。

接着通过编辑文件“exports”使“指定的用户”能访问“指定的文件夹”（可能需要root权限）：

vi /etc/exports

这条命令用于编辑etc目录下的exports。如果没有exports，则会自动新建。运行上面的指令后进入vi编辑界面。按 i 键进入**编辑模式**然后输入：

/arm 目标板IP(rw,sync,no\_root\_squash)

**截图（实验报告中包含该图片，图1）**后保存退出vi。

上面内容要求：“/arm”可以被目标板访问到并读写。

键入service rpcbind restart（旧版本Linux中为service portmap restart）、service nfs restart（ubuntu中为service nfs-kernel-server restart，需要先安装nfs-kernel-server）重新启动NFS服务，使我们的设置生效。

注：在此之前需要确保IP地址（**包含自己和合作伙伴的学号后两位**）已正确配置，具体方法请参考上一实验。

这样按照上面的步骤，就可以在minicom下，让实验板通过网络挂载宿主机的共享文件夹，如同操作实验板本身的文件夹一样操作宿主机的共享文件夹。

准备就绪后，就可以进行简单的交叉编译和在开发板上运行程序了。

三、挂载共享目录

将宿主机的目录挂载到开发板。在minicom端口输入

mount –t nfs –o nolock 宿主机IP:/arm /mnt

**将该命令的正确执行结果截图（图2）并放入实验报告中**。（宿主机IP和目标板IP中，至少有一个应当包含自己学号的后两位，否则截图无效）

注：如果出现问题，可从以下几方面排查：

* 检查网线连接和宿主机IP，确认宿主机与开发板之间能ping通。
* /etc/exports文件配置不对（例如，exports里面的IP不是开发板的IP而是主机的IP），或者修改了此配置文件但没有重启NFS服务使之生效
* 确认Firewall是否已经关闭。
* 如果nfs出现丢包，可以在mount时使用tcp。mount -t nfs -o nolock -o tcp 宿主机IP:/arm /mnt

如果上面的命令成功，则输入：

cd /mnt

ls

**截图（图3）显示ls的结果并放入实验报告中**。

注：由于开发板的nfs客户端版本较老，ls可能看不到信息（新版本Linux中这一现象很普遍）。若存在这一问题，可以尝试在mount选项中加上nfsvers=3（或者其他版本号，请多尝试几次）。注意，需要先umount后才能重新mount。

四、编写程序

宿主机上另开一个终端（这里将其称为终端２，前面的称为终端１，终端1输入的命令将在开发板上执行，终端2输入的命令将在宿主机中执行）输入如下命令（若目录不存在，请先创建，其中zhangsan替换为自己的姓名拼音）：

cd /arm/nfs\_zhangsan

vi HelloWorld.c

编写HelloWorld程序后保存退出。

五、交叉编译并运行

在终端２中输入（若HelloWorld.c是root用户创建，可能需要root权限编译）命令进行交叉编译：

arm-linux-gcc -o helloworldzhangsan HelloWorld.c

其中，**zhangsan为自己的姓名拼音**。**请将正确编译的结果截图并放入实验报告中**（图4）

如果此步出现错误信息，请查看错误信息，并且重新编辑原来的C文件修改错误，直到正确编译。

此时生成的helloworldzhagnsan文件不能在PC上运行，只能在开发板上运行。如果尝试在终端2执行该程序将会报错。可使用命令查看该文件的信息，这里可以看到arm字样。

file helloworldzhangsan

回到开着minicom 的终端1上输入

cd /mnt/nfs\_zhangsan

ls

可以看到里面有helloworldzhagnsan文件（如果看不到，请检查宿主机中该目录和目录下的文件的权限，以及mount选项中nfsvers是否正确）。输入

./helloworldzhangsan

即可得到程序运行结果。**将该结果截图放入实验报告中（图5）**。

六、运算速度测试程序

将同一程序分别进行编译和交叉编译，在开发板和主机上分别运行，测试两者的运行时间，同时也可以对比不同主机的运行时间，对开发板和主机的运算速度做一个简单的测试。

要求如下：

程序主体功能为随机生成10000个整数（数字大小在1000000以内），并对这些数冒泡排序，在执行排序之前，打印排序前的数组和当前时间，排序结束后，再次打印当前时间和排序后的数组。计算两次时间之差即为排序算法所需的执行时间。

提示：可使用man time.h查看与计时相关的函数帮助。获得的时间最好为微秒级，至少要为毫秒级。

请**将程序在宿主机和开发板中的运行结果分别截图，分析两者的差异**。（图6，图7，一张截图中必须同时包含耗费的时间，以及部分排序前的数、排序后的数）

参考程序（仅包括时间获取部分，剩余部分需自行完成）

#include<stdio.h>

#include<sys/time.h>//time.h在/usr/include/time.h，可以直接打开文件看。

void main()

{

int i;

long start\_sec, start\_usec, end\_sec, end\_usec, time\_spend;

struct timeval start\_time, end\_time;

//TODO：随机生成10000个数（可通过srand(10)设置随机数种子，并使用rand()%1000000生成1000000以内的随机数）

//获取当前时间,能精确到微秒,其中tv\_usec存放的为微秒

i=gettimeofday(&start\_time,NULL); //这里的NULL换成真正的变量更好

start\_sec =(double)start\_time.tv\_sec;

start\_usec =(double)start\_time.tv\_usec;

printf("Starttime: tv\_sec=%d,sus\_t=%d\n", start\_sec, start\_usec);

//TODO：添加排序功能

//TODO：再次获取并打印时间，然后与前面的时间相减，得到time\_spend

printf("Endtime: tv\_sec=%d,sus\_t=%d\n", end\_sec, end\_usec);

//TODO：根据Starttime和endtime计算耗费的时间

printf("Zhangsan &Lisi Test: %d micro seconds in PC\n",time\_spend);// **Zhangsan、Lisi**为组员的姓名，在嵌入式中的测试则将PC改为ARM

}

注意：可通过rand函数设置随机数，srand设置随机数的种子，只有同样的随机数种子得到的结果才能反映不同平台的速度（不同随机数种子得到的数不同，排序所需时间本身就有可能有较大差别）。

七、运算速度测试程序2（可课后完成）

请将上一步实验的排序算法改为快速排序算法，重新测试并比较耗费的时间（不要更改随机数种子）。

请**将程序在PC中运行并截图，分析快速排序与冒泡排序耗费的时间**。（图8，截图要求同图6）。说明：请确保快速排序和冒泡排序的运行环境相同（若快速排序是在课后完成的，请将冒泡排序也在相同环境下再运行一次进行比较）。

注：

1.可通过lokkit命令（若安装了此命令），在图形化界面中关闭防火墙；

2.可通过图形化方式启动NFS服务：serviceconf打开系统服务配置窗口，在左侧一栏找到 nfs 服务选项框，并选中它，然后点工具栏 的“Enable”启动它

# 实验三、UDP网络编程（4课时）

一、准备工作

本实验将实现UDP网络编程，在实验之前，需要复习UDP通信知识。可参照Linux课程的相关讲义。

本次实验的最后一步需要2组配合完成，请先选择搭档，并在第2步时选用不同的方式。

二、实验内容

1、PC之间通信

两人合作完成UDP通信程序，实现PC中的交互式通信（至少需要实现交互式聊天，可使用127.0.0.1与本机通信），并且需要将IP和端口作为程序执行时的参数（提示：通过main函数的argc、argv参数实现），方便程序移植。

要求将程序在PC上编译为server\_zhangsan和client\_lisi（zhangsan和lisi为合作双方的姓名）；

请**将聊天窗口截图（请将服务器和客户端所截的图都放入实验报告，图1、图2）**。若截图中未出现自己的姓名，截图无效，下同。

说明：前2部分实验2人完成，若截图中出现的姓名数量大于2，截图无效。第3部分实验4人完成，截图中需要且只能出现4个人的姓名学号。**若有特殊情况导致不符合要求，需要用标红的文字说明原因**，否则截图无效。

2、开发板与PC之间通信

使用交叉编译的方式分别编译客户端和服务器端程序得到serverARM\_zhangsan和clientARM\_lisi。

将PC作为服务器、开发板作为客户端（或者将PC作为客户端、开发板作为服务器，两种方式任选一种，由于最后一步需要2组配合，2组人员请使用不同的方式），使用server\_zhangsan与clientARM\_lisi（或serverARM\_zhangsan与client\_lisi）实现开发箱与PC之间的通信，**将聊天窗口截图（请将双方所截的图都放入实验报告，图3、图4）**。注意：开发板需要用虚拟键盘输入并通过开发板的snapshot命令截图，不能通过minicom聊天。

3、开发板之间的通信

自己的开发板与其它开发板实现聊天（开发板通过虚拟键盘，或连接usb键盘直接输入，不依赖于minicom）。请**将聊天窗口通过开发板截图（双方所截的图都放入实验报告，截图中含两组人员的学号信息，图5、图6）**。注意，截图需要在开发板中使用snapshot命令。

# 实验四、shell编程（4课时）

一、编写满足下列要求的shell程序

程序功能：为班级成员创建目录（以学号命名，需要考虑转专业的非连续学号）

通过shell的前2个参数指定起始学号和结束学号、其余参数为单独学号，在/tmp中创建以学号命名的目录，创建前需要判断是否存在同名目录。例如，参数为2016211820、2016211970、2016212005、2016211801…（参数中必须有一个为自己的学号），则需要创建2016211820、2016211821、……、2016211969、2016211970、2016212005、2016211801这些目录。单独学号若在前两个参数区间内，无需重复创建；单独学号若不在前两个参数区间内，则表示这个学号是后来转入本班的同学，需要创建该目录。单独学号的数量可以有很多，不设上限。

程序要求：

1、出错判断：若shell参数数量不对（小于2）、或者输入的参数不是数字、或输入的学号范围过大（要求学号总数量小于200）、或第二个参数小于第一个参数，提示对应的出错信息；

2、创建目录：若不存在这些目录，请创建这些目录并在程序结束时打印出新添加的目录；若只存在部分目录，请创建缺失部分，并在程序结尾处打印出新添加的目录名；若全部存在，打印出“所有目录均已存在，无需创建”。

执行上述程序4次（输入的参数出错、/tmp中不存在此类目录、存在部分此类目录、存在全部指定目录，分别截图）。附上**shell程序代码（代码1）**、**执行过程截图（图1、图2、图3、图4）**。

二、编写shell脚本，实现计算器

先输出作者学号与姓名，并提示用户输入2个整数和运算符，输出运算结果，实现方式不限，但必须满足如下要求：

至少需要实现加减乘除运算；

需要判断输入不正确的情况（例如，输入的不是数字和运算符），如果输入不正确，需要打印提示并接受用户重新输入；

计算出结果并打印出来后，程序重复执行计算器功能，直到用户输入指定字符（例如quit、q等）。

请在实验报告中附上**源码**和**完整运行截图**，截图中需要分别展示+、-、\*、/运算**（图5）**，还需要展示负数运算、除数为0、输入的运算符不对、输入的不是数字这4种特殊情况**（图6）**。

# 实验五、Linux数据库编程（8课时）

一、MySQL安装与启动

本次实验请使用自己的PC，且需要提前安装MySQL，安装方法请参考课件（若使用的是虚拟机，安装完成后，需要将网络改为桥接，且桥接的网卡需要设置为有线网卡）。

安装完成后，通过service mysqld start命令启动mysql服务器。

二、通过MySQL命令建立数据表

1、设计一个数据库，用于管理聊天用户信息

数据表至少包括ID、昵称、密码、好友列表（friend1、friend2、friend3，好友以ID方式存储）、是否在线（online，枚举型）、最后一次登录的IP（ip，字符串或INT）、端口（port，smallint）信息。

通过mysql命令，完成数据库、数据表的建立。数据库的名称为自己的姓名拼音、数据表的名称为user011(011改为自己学号后三位)。数据表至少应该包含id、nickname、password、friend1-friend3、online、ip、port。对创建数据表的命令及其执行结果截图（图1）

注意：输入命令时，引号、冒号等符号必须使用英文输入法输入（不要使用全角字符、也不要直接从网页中复制）。

2、数据查询与更新

选定数据表user011，并在该数据表中插入数据（至少2条记录，即自己和队友）。对插入数据的命令及其执行结果截图（图2）。注意，请将id、friend1设置为合作双方的学号、ip设置为本机IP、端口设置为学号后4位（若以0开头，请在前面加1）。

三、数据库编程

通过c语言编程实现用户管理，编译（必须使用make命令编译，所得可执行文件为**zhangsan**）并**截图**（图3）

以下功能在同一程序中实现，可设计一个菜单分别实现，菜单样例（建议自行设计菜单）：

一级菜单：请选择需要1）查询；2）更新

请输入要查询/更新的用户名；

二级菜单：需要查询/更新该用户的什么信息？

1）好友列表；2）是否在线；3）ip与端口；

i）通过用户名查询该用户的好友列表并截图（图4）；通过用户名查询该用户是否在线（图5）；通过用户名查询该用户的IP、端口（图6）；

ii）通过用户名更新该用户是否在线信息并再次查询（图7）；通过用户名更新用户的IP、端口并再次查询（图8）。

在实验报告中**附上源码**。

注意：

部分Linux下，执行mysql可能需要root权限

编程时，输入的信息可通过sprintf或strcpy等方式写入某个字符串（推荐使用sprintf），然后在程序中将该字符串作为mysql语句的一部分。

# 实验六、嵌入式聊天软件设计（8课时）

本次实验6人（即3组）一起完成（所有成员都需要参与，且都要熟悉整个系统的原理）

实验要求：

结合前面的实验，完成服务器端程序和客户端程序（需2个客户端）。其中，服务器端在PC中运行（该PC上安装了数据库）；客户端程序在开发板上运行。PC、开发板均通过网线接入实验室内网，确保服务器和2个客户端网络畅通。

注意：由于开发板的出厂MAC地址都相同，为了确保接入内网后网络正常，请通过ifconfig命令更改MAC地址（例如，将MAC地址改为01:02:03:04:ab:cd，abcd为学号后4位）。

通过交叉编译，得到可在开发板上使用的客户端程序。

服务器端程序功能：

1. 同时具备通信和数据库管理功能（数据库部分参照前面的实验）；
2. 绑定IP、端口，等待接收客户端消息（TCP、UDP均可）；
3. 收到用户（客户端）发过来的登录请求（同时会附带UDP聊天端口号）后，在数据库中更新该用户的在线状态、IP、端口（这个端口是该用户与其它用于聊天的端口，而非与服务器通信的端口）；
4. 通过用户id在数据库中查询该用户的好友列表，并通过在线好友的id查询好友的IP、端口（这个端口将用于UDP聊天），发送给用户；
5. （选作）收到离线消息后，更新用户的在线状态；
6. （选作）一旦有用户的在线状态或IP发生更改，需要更新数据库，并将新的状态和IP通知其好友。

客户端程序功能：

1. 程序运行时需要指定参数（服务器IP、与服务器通信的端口、与其他用户聊天的端口）；
2. 客户端的通信包括2部分：与服务器之间交互信息；两个客户端之间使用udp聊天（建议通过多进程或多线程实现全双工）
3. 用户输入id（有兴趣的同学可进一步确认密码），连接服务器（服务器的地址和端口在程序执行参数中指定）；
4. （在绑定本机UDP聊天端口的基础上）向服务器发送本机IP、UDP聊天用的端口号；
5. 接收到服务器发来的在线好友的IP、端口信息（可多次接收）；
6. 用户输入好友的id，开始聊天（注意：实际在后台是通过ip、端口通信的，但对使用者来说无需关注IP、端口）；
7. （选作）程序正常退出前，向服务器发送离线消息。

在一台PC中运行服务器端程序。在两台嵌入式设备上同时运行客户端程序，实现两个客户端程序之间的聊天（注意：两台客户端之间初始状态下并不知道对方的IP、端口，必须通过服务器来查询）。

请使用makefile分别编译并截图（图1、图2、图3）；

分别执行上述3个程序并截图（图4、图5、图6）。截图中需要有关键信息，包括：客户端输入用户名、显示出在线好友列表、选择好友进行聊天，聊天内容需要包含所有成员学号姓名。

说明：

1. 为了实现上述功能，需要在PC服务器、2台嵌入式设备上分别实现socket通信。
2. 两个客户端需要分别与服务器通信（通常为TCP）；同时，两个客户端之间使用UDP通信。
3. 需要在后台先完成socket和绑定等功能

请在实验报告中附上文本形式的源码。

# 实验七、嵌入式聊天软件设计2（附加）

本次实验4人（即2组）一起完成

实验要求：

结合实验三、实验六、实验七，完成嵌入式聊天软件。在实验七基础上，程序增加以下功能：

1. 客户端每隔一段时间向服务器发送“心跳”信号，一旦出现断网等异常，更新用户状态为离线；
2. 客户端软件具有保存、导入聊天记录的功能；
3. 客户端实现文件收发功能（选作）
4. 该软件具有UI，图形化界面或文字界面均可，且可在嵌入式设备上运行（选作，请自行下载安装qt-everywhere-opensource-src）；

在嵌入式设备中运行程序并截图。截图有多个，至少包含：客户端界面截图（图形界面或文字界面均可，图1）、双方聊天过程截图（图2、图3，聊天内容需包含所有参与人员的学号姓名）、保存与导入聊天记录截图（图4、图5）。

请在实验报告封面上注明所有的合作队友。

说明：

图形界面至少包含

1、至少包含2个文本框（一个用于接收消息，一个用于发送消息）

2、发送消息按钮、保存消息按钮（若设置自动保存则无需这个按钮）、导入消息按钮；

3、好友列表框与对应的获取好友列表按钮（或其它方式组件，形式不限，只要能够展示好友列表即可）

4、文本框与对应按钮（或下拉菜单等其它方式）用于输入聊天对象。

请在实验报告中附上文本形式的源码。

附加：实现群聊功能（提示：所有用户发送的消息都发给服务器，服务器把收到的消息发给所有客户端），需要在数据库中加入群号码、群成员等信息。