测试题0及参考答案

（1）ARM汇编程序由\_**机器\_\_**\_\_指令、\_\_**汇编\_**\_\_指令和\_**\_伪\_\_**\_指令构成。

（2）ARM伪指令可以分为以下几类： **符号定义伪指令 数据定义伪指令汇编控制伪指令 信息报告伪指令 宏指令以及其他伪指令**。

（3）假设存储数据0x12345678于ARM微处理器内存0X30000000开始的位置，则0X30000001内存位置的数据为\_C\_\_\_\_（A.0X12 B.0X34 C.0X56 D.0X78）（采用小端模式进行存储）

（4）一般情况下，ARM微处理器异常处理模式共有\_\_\_7\_\_\_种，机器启动后第一条指令执行的是\_\_A\_\_\_（A.复位异常处理函数指令 B.中断异常处理指令 C.IRQ异常处理指令 D.指令预取终止异常）。

（5）调用函数FUN(X,Y,Z)，则实参值分别通过\_\_r0\_\_、\_r1\_\_\_、\_r2\_\_寄存器来进行传递，如果参数超过4个，则参数传递规则为\_\_\_\_通过栈进行传递\_\_\_\_\_\_\_\_。

（6）举例列出一款ARM7TDMI微内核的嵌入式微处理器\_S3C44B0X\_，ARM920T微内核的嵌入式微处理器\_S3C2410\_，ARM11内核的嵌入式微处理器\_\_S3C6410\_\_\_\_，并列举2款64位ARM微内核\_Cortex-A53 \_\_、\_\_Cortex-A57\_\_\_\_\_\_\_\_。

（7）利用汇编和C混合编程，设计代码完成求a,b,c中最大值功能，要求写出汇编启动代码和C代码。

（略）此知识点不需要掌握

测试题1及参考答案

1.嵌入式Linux操作系统包括 bootloader 、 内核 、文件系统 三部分组成。

2.在PC机上Linux系统编译使用的编译器名为 gcc ，ARM处理器嵌入式编译器名为 arm-linux-gcc 。

3.bootloader的功能： ①引导操作系统内核启动②提供辅助命令工具 。

4.列出最常用的bootloader： 、 、 、 、 、 。

5.在uboot中，打印开发板上环境变量值的命令为 printenv ，设置IP地址为192.168.1.1的命令为

setenv ipaddr 192.168.1.1 ，假如嵌入式内核名为vmlinux，通过tftp加载内核的命令为

tftp vmlinux 内存地址 ，启动嵌入式Linux内核的命令为 bootm 。

6.从C语言角度来理解，嵌入式Linux内核仅是众多 函数 的集合体，其有一个类似于main的函数，名称为 start\_kernel 。

7.嵌入式Linux内核裁剪命令为 make menuconfig ；裁剪完成后生成配置选择文件 .config ；编译内核命令为 make zImage 。

8.启动linux后常用的命令如ifconfig、cp、ls等通常位于 C （A.bootloaderB.内核C.文件系统）中；嵌入式Linux环境下生成文件系统的常用工具为 mkyaffs2img ；以yaffs方法为例，将嵌入式文件夹压缩成一个yaffs2格式的文件系统命令为 mkyaffs2img 文件夹 镜像文件名 。

9.构建嵌入式Linux系统时，烧写bootloader的工具为 hjtag ；烧写Linux内核一般使用

A （A.bootloader B.文件系统 C.交叉编译器）来烧写，烧写文件系统一般使用

A （A.bootloader B.文件系统 C.交叉编译器）来烧写。

10.简述构建嵌入式Linux系统基本过程

略，见实验材料

测试题2及参考答案

学号: 姓名： 日期：

1. Linux中自动生成makefile的工具集名称为 ***autotools***  ；工具集包括aclocal、autoscan、autoconf、autoheader和 ***automake***  等几部分组成。

2.一个源码文件hello.c利用自动生成makefile的工具集后得到文件一批文件后，利用命令 ***#./configure***

生成Makefile，利用命令 ***#make dist***  生成发行压缩包文件hello.tar.gz。

3.嵌入式Linux应用程序移植常用configure命令生成Makefile,一般来说，指定安装目录为/opt/soft的命令为 ***./configure --prefix=/opt/soft***  ；如果待生成的可执行目标板为ARM处理器，则命令为  ***./configure --prefix=/opt/soft --HOST=arm-linux***  。

4.以web服务器boa移植为例，简述其基本过程。

[略]见实验材料

5.嵌入式数据库经常使用sqlite，为可以在开发板上运行的轻型数库。通常情况下sqlite提供

***命令*** 和  ***函数编程***  两种使用方式。

6.在sqlite下创建数据库stu.db命令为 ***sqlite3 stu.db***  ；创建包含学号，姓名和电话号码的数据表StuPhone命令为 ***create table StuPhone( id integer primary key, name text, phoneNo text );***  ；插入10010201,wangming,13900008888记录的命令为 ***insert into StuPhone values(10010201,wangming,13900008888)***  。

测试题3及参考答案

1. 嵌入式Linux内核是可裁剪系统，通常情况下使用 模块 机制进行设计。模块程序和应用程序分别位于Linux系统的 内核 空间和 用户 空间。

2. 嵌入式Linux系统模块程序hello.c被编译成模块hello.ko后，动态插入内核的命令为 #insmod hello.ko ，查看Linux内核中有哪些模块的命令为 #lsmod ，删除内核中模块hello.ko的命令为 #rmmod hello 。

3.编写一模块，向内核中添加两整数相加功能函数和两整数相减功能函数，并要求在加载模块时打印出“hello，I am in kernel now!”，卸载模块时打印“hello, I will leave from kernel now!”,编写Makefile，并简述加载到内核和卸载出内核的基本命令。

(1)模块文件



（2）Makefile

***obj-m:=hello.ko***

***all:***

***make –C /opt/linux-2.6.38 SUBDIRS=$(shell pwd) modules***

***clean:***

***rm -rf \*.ko \*.o***

（3）编译和加载

***#make***

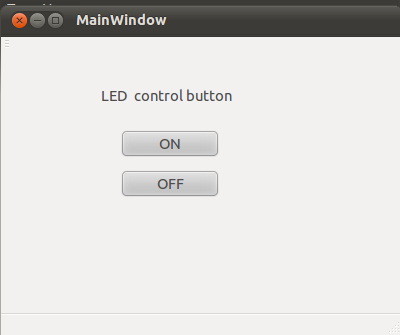
***#inmsod hello.ko***

测试题4及参考答案

1. 在嵌入式Linux开发过程中，EmbeddedQT通常情况下用于  ***C***  （A.内核代码B.驱动代码C.图形应用程序）设计。EmbeddedQT采用工具开发包的形式提供给用户，一般情况下包括 ***图形设计器***  、 ***QT的C++类库*** 和Makefile 制作工具，字体国际化工具等。

2. QT开发中采用 信号和插槽 机制来连接两个对象之间的通讯，假设对象a的clicked信号和对象b的handleFunction（）相关联，对应语句为： ***connect(&a,SIGNAL(clicked()),&b,SLOT(handleFunction()));***

3．假设有嵌入式LED报警灯驱动设备文件/dev/led，点亮led灯调用函数ioctl(fd,LEDON),熄灭led灯调用函数ioctl(fd,LEDOFF),请设计QT应用程序完成如下任务，点击ON按钮点亮LED灯，点击OFF按钮熄灭LED灯。写出这两个按钮的相关联的函数。假设ON按钮名称为m\_on,OFF按钮名称为m\_off。



关联函数：

***connect(m\_Win,SIGNAL(clicked()),m\_on,SLOT(LedOnFun()));***

***connect(m\_Win,SIGNAL(clicked()),m\_off,SLOT(LedOffFun()));***

***void LedOnFun( )***

***{***

***int fd;***

***fd=open(“/dev/leddev”,O\_RDWR);***

***ioctl(fd,LEDON);***

***close(fd);***

***}***

***void LedOffFun()***

***{***

***int fd;***

***fd=open(“/dev/leddev”,O\_RDWR);***

***ioctl(fd,LEDOFF);***

***close(fd);***

***}***

测试题5及参考答案

1. 嵌入式Linux驱动设备分为\_***字符设备\_***\_\_、***\_块设备\_***\_和网络设备三种类型，其中通常情况下键盘、鼠标、LCD屏等设备驱动属于***\_字符\_***设备，硬盘设备属于\_***块\_***\_设备。

2. 设备要能被使用，通常需要设备驱动软件，在Linux系统中，设备驱动驱动软件位于\_***\_A\_***\_(A.内核空间 B.用户空间)，假设有LED报警灯设备驱动软件leddrv.ko，则加载该驱动软件到内核命令为***\_#insmod leddrv.ko\_***\_，测试完成后，将该软件从内核中卸载命令为\_***#rmmod leddrv\_\_\_***\_。

3. 除网络设备外，设备驱动通常采用\_文件\_\_方式进行访问，一般包括\_***\_打开设备\_***\_、读写控制设备，最后***\_关闭设备\_\_***\_。在Linux中设备文件名本质上是设备号，将两者之间建立关联的命令为mknod，假设LED设备驱动主设备号为253，次设备号为0，设备名为leddev，则建立设备文件的具体命令为\_***#mknod /dev/leddev c 253 0\_\_***。

4．简述编写看门狗驱动基本步骤

***编写硬件驱动代码，包括：***

*** 编写硬件接口函数***

*** 建立文件系统与硬件接口函数的关联***

*** 注册字符设备***

*** 添加模块代码***

*** 编译设备驱动并加载到内核，包括：***

*** 编写Makefile***

*** 编译***

*** 加载***

*** 创建设备节点***

*** 编写应用程序访问底层设备驱动***