厦门大学《嵌入式系统》课程期末试卷

信息学院　　软件工程系　　2019 级　　软件工程专业

主考教师：曾文华　试卷类型：（A 卷）答案　考试时间：2022.1.7

一、填空题（30 个空，每 1 空 1 分，共 30 分；在答题纸填写答案时请写上每个空格的对应编号）

1. ARM Cortex-A 系列处理器又称__高性能（1）__处理器，ARM Cortex-R 系列处理器是针对__实时性（2）__要求高的嵌入式系统提供的解决方案，ARM Cortex-M 系列处理器是针对__成本和功耗（3）__敏感的嵌入式系统提供的解决方案。

2. Ubuntu 是 Linux 系统最受欢迎的的__发行版（4）__。

3. ARM 处理器有两种状态，分别是__ARM（5）__状态和__Thumb（6）__状态。

4. μCLinux 是专门针对没有__MMU（存储管理单元）（7）__的处理器设计的。

5. RT-Linux 是具有__硬实时（8）__特性的多任务操作系统；RT-Linux 通过在 Linux 内核与硬件中断之间增加一个精巧的可抢先的__实时内核（9）__，把标准的 Linux 内核作为__实时内核（9）__的一个进程与用户进程一起调度。

6. 嵌入式 Linux 系统启动后，先执行__Bootloader（10）__，进行硬件和内存的初始化工作，然后加载__Linux 内核（11）__和__根文件系统映像（12）__，完成 Linux 系统的启动。

7. Linux 是单内核的，单内核存在可扩展性以及可维护性差的缺点，__模块（13）__机制的引入就是为了弥补这一缺点。

8. Linux 内核支持动态可加载模块，模块通常是__设备驱动（14）__程序。

9. Linux 的设备驱动程序开发调试有两种方法，一种是直接编译到__内核（15）__，另一种是编译为__模块（16）__的形式；第一种方法效率较__低（17）__，第二种方式效率较__高（18）__。

10. Linux 抽象了对硬件的处理，所有的硬件设备都可以作为普通文件一样对待，可以使用标准的系统调用接口来完成对设备的打开（open）、关闭（close）、读写（read、write）和__I/O 控制操作（ioctl）（19）__，驱动程序的主要任务是实现这些系统调用函数。

11. 若要创建一个设备名为/dev/lp0、主设备号为 6、次设备号为 0 的字符设备，其命令为__mknod /dev/lp0 c 6 0 （20）__。

12. 使用 mmap 系统调用（mmap()函数），可以将__内核（21）__空间的地址映射到__用户（22）__空间。

13. Android 的软件架构采用了分层结构，由上至下分别为：Application 应用层、Application Framework

应用框架层、Android Runtime & Libraries 运行时库和本地库层、 Linux Kernel （23） 内核层。

14. Linux 字符设备就是采用 字节流 （24） 形式通讯的 I/O 设备，绝大部分 Linux 设备都是字符设备。

15. 块设备没有 read 和 write 操作函数，对块设备的读写是通过 请求 （25） 函数完成的。

16. Atlas 200 DK 是华为公司生产的面向 AI（26）应用的开发者套件，其核心是 Ascend 310 AI（27）处理器。

17. ModelArts 是华为云面向 AI （28） 开发者的一站式 AI （28） 开发平台。

18. Qt 是一个由 Qt Company 开发的跨平台 C++ 图形用户界面 （29） 应用程序开发框架。

19. 对网络设备的访问必须使用 套接字（Socket）（30） ，而非读写设备文件。

二、名词解释（请写出下列英文缩写的中文全称，10 小题，每 1 小题 1 分，共 10 分；在答题纸填写答案时请写上每小题的对应编号）

1. CPSR：Current Program Status Register，当前程序状态寄存器

2. JTAG：Joint Test Action Group，联合测试工作组

3. JFFS3：Journalling Flash File System Version3，闪存日志型文件系统第 3 版

4. YAFFS：Yet Another Flash File System，是专为嵌入式系统使用 NAND 型闪存而设计的一种日志型文件系统

5. GPIO：General Purpose Input/Output，通用输入输出

6. SPI：是串行外设接口（Serial Peripheral Interface）

7. I2C：（Inter Integrated-Circuit，IIC，I2C，内部集成电路）总线

8. CAN：全称为 Controller Area Network，即控制器局域网

9. NFC：近场通信（Near Field Communication）

10. Android NDK：Android Native Development Kit

三、简答题（10 小题，共 30 分；在答题纸填写答案时请写上每小题的对应编号）

1. （3 分）常见的嵌入式操作系统有哪些？

答：
嵌入式 Linux
VxWorks
μC/OS-II
Windows CE
Sysbian
Android
iOS
其它：QNX，Palm OS，LynxOS，NucleusPLUS，ThreadX，eCos

2. （3 分）ARM 的处理器有哪 7 种运行模式？

答：
系统模式（SYS）
用户模式（USR）
快速中断模式（FIQ）
管理模式（SVC）
数据访问终止模式（ABT）
外部中断模式（IRQ）
未定义指令终止模式（UND，未定义模式）

3. （2 分）什么是交叉开发（交叉编译）？

答：
宿主机/目标机模式：
宿主机：PC 机（x86 环境）
目标机：可以是实际的运行环境，也可以用仿真系统替代实际的运行环境（ARM 环境）

4. （4 分）宿主机与目标机通常有 4 种连接方式，请结合 IMX6 实验箱分别说明每一种连接方式的具体内容和应用场景。

答：
宿主机与目标板的连接方式：
（1）串口：实验箱的 COM1 TO USB 就是串口，在宿主机（电脑）上运行“Xshell 2.0”时，就是利用该串口，实现宿主机（电脑）与目标机（实验箱）的连接。
（2）以太网接口（RJ45）：挂载方式在“Xshell 2.0”上执行“ mount -t nfs 59.77.5.101:/imx6 /mnt”时，就是利用网口，将 Ubuntu 上的文件挂载到实验箱上。
（3）USB 接口：如实验箱的 USB OTG 口就是 USB 接口，实现将 Windows 系统文件夹下的实验箱内核（操作系统）烧写到实验箱。
（4）JTAG 接口：在做 ST32 实验时，利用 JTAG 接口，将可执行文件烧写到从 CPU 的 Flash 中。

5. （3 分）NFS 服务、Samba 服务和 VMware Tools 分别实现什么功能？

答：

（1）NFS 服务实现虚拟机（Ubuntu 系统）与实验箱的文件共享，即虚拟机（Ubuntu 系统）的文件夹可以通过 NFS 方式挂载（共享）到实验箱，从而在实验箱上可以访问虚拟机的文件夹（文件）。

（2）Samba 服务可以实现 Windows 系统的文件夹与虚拟机（Ubuntu 系统）的文件夹共享。

（3）VMware Tools 可以实现从 Windows 操作系统拖拽文件到 Ubantu 系统的桌面上面；用户也可以从 Windows 上面的文档里面复制一句话粘贴到 Ubantu 系统里面。

6. （2 分）在 Ubuntu 上执行 make 命令（交叉编译生成可执行文件）前，需要先执行“source /opt/poky/1.7/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi”命令，请问该命令的作用是什么？

答：指定交叉编译器的路径。

7. （2 分）如果采用挂载方式运行程序，需要在“Xshell 2.0”上先执行 mount 命令（NFS 挂载命令）。假设 Ubuntu 的 IP 地址为 59.77.5.101，需要将 Ubuntu 的/imx6 目录挂载到实验箱的/mnt 目录，请写出 mount 命令。

答： mount -t nfs 59.77.5.101:/imx6 /mnt 。

8. （4 分）Android HelloWorld 工程可以在 4 个地方运行，请说出这 4 个地方的具体名称。

答：

（1）在 Android 的虚拟设备（AVD）上运行 HelloWorld 工程

（2）在 Genymotion 虚拟设备上运行 HelloWorld 工程

（3）在 Android 手机上运行 HelloWorld 工程

（4）在实验箱上运行 HelloWorld 工程

9. （3 分）简述在 IMX6 实验箱上开发 Android NDK 程序的步骤。

答：

第一步：在 Ubuntu 环境下编写 hello-jni.c 和 Android.mk 程序，并编译生成 libhello-jni.so 库文件

第二步：在 Android Studio 环境下编写 HelloJni 工程

第三步：将 libhello-jni.so 库文件拷贝到 Android Studio 的 HelloJni 工程中

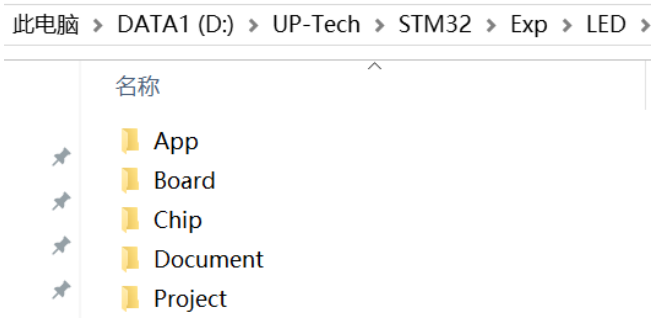
第四步：在 Android Studio 环境下编译 HelloJni 工程，并在实验箱上运行 HelloJni 工程

10. （4 分）简述云（ModelArts）+ 端（Atlas 200 DK）协同垃圾分类实验的具体步骤。

答：
第一步：使用 OBS 服务将实验用训练代码和数据集上传到 OBS 桶中
第二步：在 ModelArts 中创建 Notebook 训练作业
第三步：在 Notebook 的 Jupyter 中运行训练作业，生成模型（.air 格式）
第四步：将生成的模型从 OBS 下载到本地电脑硬盘
第五步：在 MobaXterm 环境下，完成模型的转换
第六步：从互联网上下载一些垃圾图片
第七步：在 MobaXterm 环境下，运行垃圾分类程序

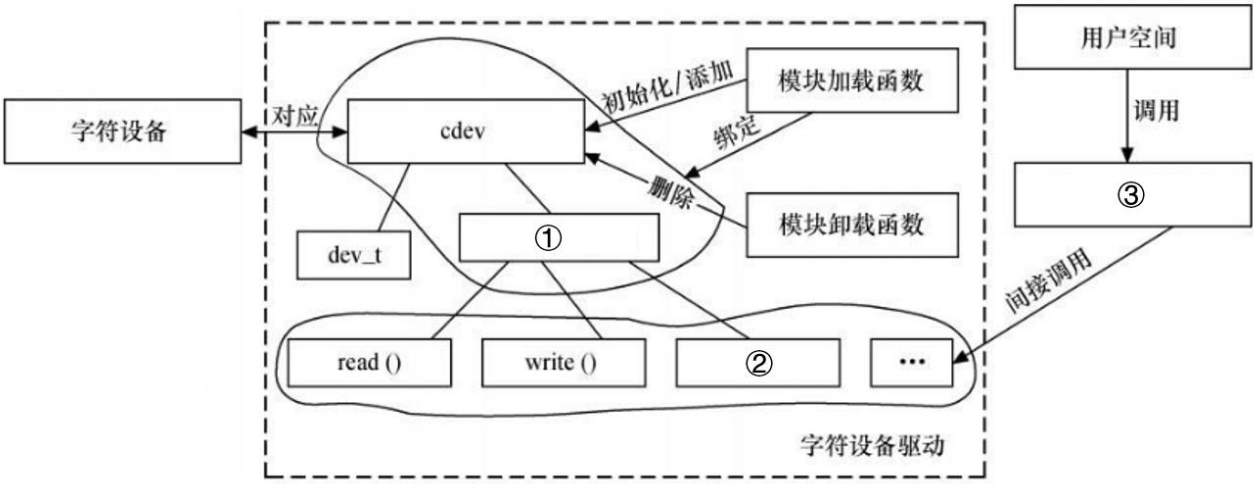
四、综合题（8 小题，共 30 分；在答题纸填写答案时请写上每小题的对应编号）

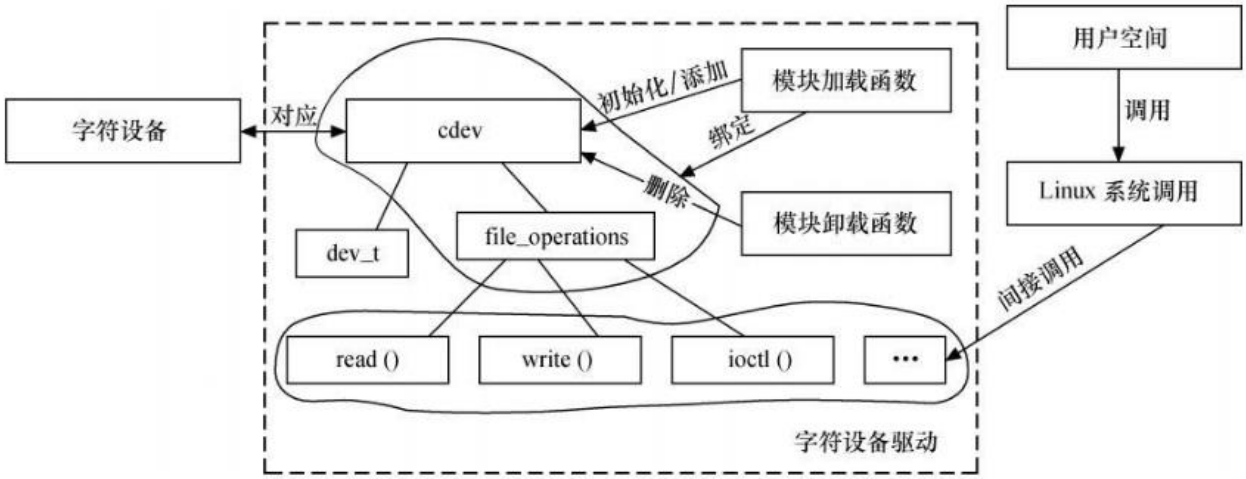
1. （3 分）STM32 LED 灯实验程序的工程项目文件夹如下，请问该工程项目文件夹的 5 个子文件夹中分别存放什么内容？



答：
App：用户程序
Board：开发板驱动程序
Chip：STM32 处理器芯片驱动程序
Document：工程说明文档
Project：工程文件

2. （3 分）下图为字符设备驱动框架，请填写图中 3 个空白方框的内容。





答：

（1）file_operations

（2）ioctl()

（3）Linux 系统调用

3. （5 分）已知当前目录下有 pthread.c 和 Makefile 两个文件，其中 Makefile 文件的内容如下：

CC = arm-poky-linux-gnueabi-gcc -march=armv7-a -mthumb-interwork -mfloat-abi=hard -mfpu=neon -

```
mtune=cortex-a9 --sysroot=/opt/poky/1.7/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi
EXTRA_LIBS += -lpthread
EXP_INSTALL = install -m 755
INSTALL_DIR = ../bin
EXEC = ./pthread
OBJS = pthread.o
all: $(EXEC)
$(EXEC): $(OBJS)
    $(CC) -o $@ $(OBJS)   $(EXTRA_LIBS)
install:
    $(EXP_INSTALL) $(EXEC) $(INSTALL_DIR)
clean:
    -rm -f $(EXEC) *.elf *.gdb *.o
```

请问在当前目录下分别执行 make、make install、make clean 命令，分别会显示什么结果？如果要将编译后的可执行文件能够在 Ubuntu 环境（x86 环境）下运行，请问如何修改 Makefile 文件？
假设：pthread.c 文件是正确的（不会出现编译错误）。在写显示结果时请用 CC1 代替 arm-poky-linux-gnueabi-gcc -march=armv7-a -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a9 --sysroot=/opt/poky/1.7/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi -O2 -pipe -g -feliminate-unused-debug-types；用 CC2 代替 arm-poky-linux-gnueabi-gcc -march=armv7-a -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a9 --sysroot=/opt/poky/1.7/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi。

答：
（1）执行 make 命令：

```
CC1     -c -o pthread.o pthread.c
CC2 -o pthread pthread.o     -lpthread
```

（2）执行 make install 命令：

```
install -m 755    ./pthread ../bin
```

（3）执行 make clean 命令：

```
rm -f ./pthread *.elf *.gdb *.o
```

（4）将 CC = arm-poky-linux-gnueabi-gcc -march=armv7-a -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a9 --sysroot=/opt/poky/1.7/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi

修改为 CC = gcc

4. （5 分）以下 2 个程序为 C 语言调用汇编语言的程序，请问程序 1 和程序 2 分别属于什么调用形式（什么汇编）？并补充 2 个程序中 3 个划线处的内容。

程序 1：

```
__EXPORT__  add                    @声明 add 子程序将被外部函数调用
add:
    ADD r0,r0,r1
    MOV pc,lr

__Extern__  int add(int x, int y);          //声明 add 为外部函数
void main()
{
    int a=1, b=2, c;
    c = add(a, b);
}
```

程序 2：

```
void enable_IRQ(void)
{
    int tmp;
    ____asm__                                          //声明内联汇编代码
    {
        MRS tmp, CPSR
        BIC tmp, tmp, #0x80
        MSR CPSR_c, tmp
    }
}
```

答：

（1）嵌入式汇编

（2）内联汇编

EXPORT

Extern

__asm

5. （5 分）以下程序为改进后的读取小键盘按键值的主程序，请问该程序中的第 5）、13）、15）、16）、18）行分别是完成什么任务？

```
1)  int main(int argc,char *argv[])
2)  {
3)      int keys_fd;
4)      struct input_event t;
```

```
5)      keys_fd = open(KEYDevice, O_RDONLY);
6)      if(keys_fd <= 0)
7)      {
8)          printf("open key device error!\n");
9)          return 0;
10)     }
11)     while(1)
12)     {
13)         if(read(keys_fd,&t,sizeof(t)) == sizeof(t))
14)         {
15)             if(t.type == EV_KEY)
16)                 if(t.value == 0)
17)                 {
18)                     printf("%c\n",key_value(t.code));
19)                 }
20)         }
21)     }
22)     close(keys_fd);
23)     return 0;
24) }
```

答：

5）打开输入设备文件

13）读取输入设备事件，并判断是否读取成功

15）判断是不是小键盘

16）判断小键盘的按键是不是按下

18）将小键盘的键值转换为 0-9 和*、#显示出来

6. （3 分）以下程序为步进电机的主程序，请问该程序中的第 7）、14）、16）行分别是完成什么任务？

```
1)  int main(int argc, char *argv[])
2)  {
3)      unsigned char data;
4)      int mem_fd;
5)      unsigned char *cpld;
6)      mem_fd = open("/dev/mem", O_RDWR);
7)      cpld = (unsigned char*)mmap(NULL,(size_t)0x04,PROT_READ | PROT_WRITE |
    PROT_EXEC,MAP_SHARED,mem_fd,(off_t)(0x8000000));
8)      if(cpld == MAP_FAILED)
9)          return;
10)     while(1)
11)     {
12)         printf("请输入步进电机状态：\n");
```

```
13)             scanf("%d",&data);
14)             *(cpld+(0xe2<<1)) = data;     //步进电机地址 0xe2<<1
15)         }
16)         munmap(cpld,0x04);
17)         close(mem_fd);
18)         return 0;
19)  }
```

答：

7）内存映射操作函数（mmap 函数）：将内核空间映射到用户空间

14）根据输入的数据，控制步进电机是顺时针转、逆时针转，还是停止

16）解除内存映射操作

7. （3 分）假设 HelloWorld 程序的可执行文件 hello 位于 Ubuntu 的/imx6/whzeng/hello 目录下，Ubuntu 的 IP 地址为 59.77.5.101，并且实验箱与 Ubuntu 系统已经 Ping 通。请问在“Xshell 2.0”上执行哪几条命令，可以运行 hello 程序（请写出具体的命令）？

答：

```
mount  -t  nfs  59.77.5.101:/imx6        /mnt
cd /mnt/whzeng/hello
./hello
```

8. （3 分）已知 Ubuntu 系统某文件夹下有一个可执行文件 led，现要求通过下载的方式将该文件下载（传送）到 IMX 实验箱，请写出具体的操作步骤（包括具体的操作命令）。假设 Windows 系统下有 tftpd32.exe 文件；Windows 系统的 IP 地址为 59.77.5.110。

答：

（1）将 led 文件从 Ubuntu 文件夹拖到 Windows 系统下存放有 tftpd32.exe 文件的文件夹

（2）执行 tftpd32.exe 文件

（3）在“Xshell 2.0)”下，执行：

```
cd /home/root
tftp -gr led 59.77.5.110
```