



# 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) 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 系列处理器又称为\_\_\_\_(1)\_\_\_\_处理器，ARM Cortex-R 系列处理器是针对\_\_\_\_(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\_\_\_\_(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\_\_\_\_(4)\_\_\_\_。
3. ARM 处理器有两种状态，分别是\_\_\_\_(5)\_\_\_\_状态和\_\_\_\_(6)\_\_\_\_状态。
4.  $\mu$ CLinux 是专门针对没有\_\_\_\_(7)\_\_\_\_的处理器设计的。
5. RT-Linux 是具有\_\_\_\_(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\_\_\_\_(9)\_\_\_\_，把标准的 Linux 内核作为\_\_\_\_(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\_\_\_\_(10)\_\_\_\_，进行硬件和内存的初始化工作，然后加载\_\_\_\_(11)\_\_\_\_和\_\_\_\_(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，\_\_\_\_(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\_\_\_\_(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\_\_\_\_(15)\_\_\_\_，另一种是编译为\_\_\_\_(16)\_\_\_\_的形式；第一种方法效率较\_\_\_\_(17)\_\_\_\_，第二种方式效率较\_\_\_\_(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\_\_\_\_(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\_\_\_\_(20)\_\_\_\_。
12. 使用 mmap 系统调用（mmap()函数），可以将\_\_\_\_(21)\_\_\_\_空间的地址映射到\_\_\_\_(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、\_\_\_\_(23)\_\_\_\_内核层。

14. Linux 字符设备就是采用\_\_\_\_(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\_\_\_\_(25)\_\_\_\_函数完成的。

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

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

18. Qt 是一个由 Qt Company 开发的跨平台 C++\_\_\_\_(29)\_\_\_\_应用程序开发框架。

19. 对网络设备的访问必须使用\_\_\_\_(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
2. JTAG
3. JFFS3
4. YAFFS
5. GPIO
6. SPI
7. I2C
8. CAN
9. NFC
10. Android NDK

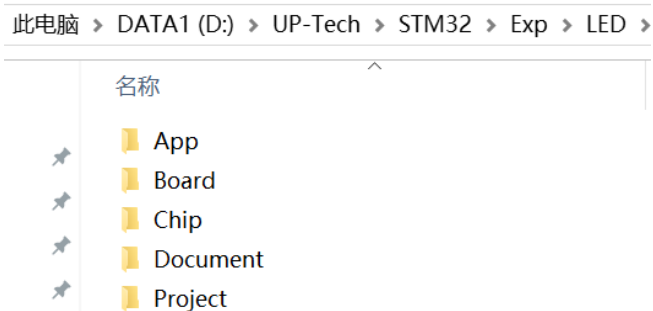
三、简答题（10 小题，共 30 分；在答题纸填写答案时请写上每小题的对应编号）

1. （3 分）常见的嵌入式操作系统有哪些？
2. （3 分）ARM 处理器有哪 7 种运行模式？
3. （2 分）什么是交叉开发（交叉编译）？
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。
5. （3 分）NFS 服务、Samba 服务和 VMware Tools 分别实现什么功能？

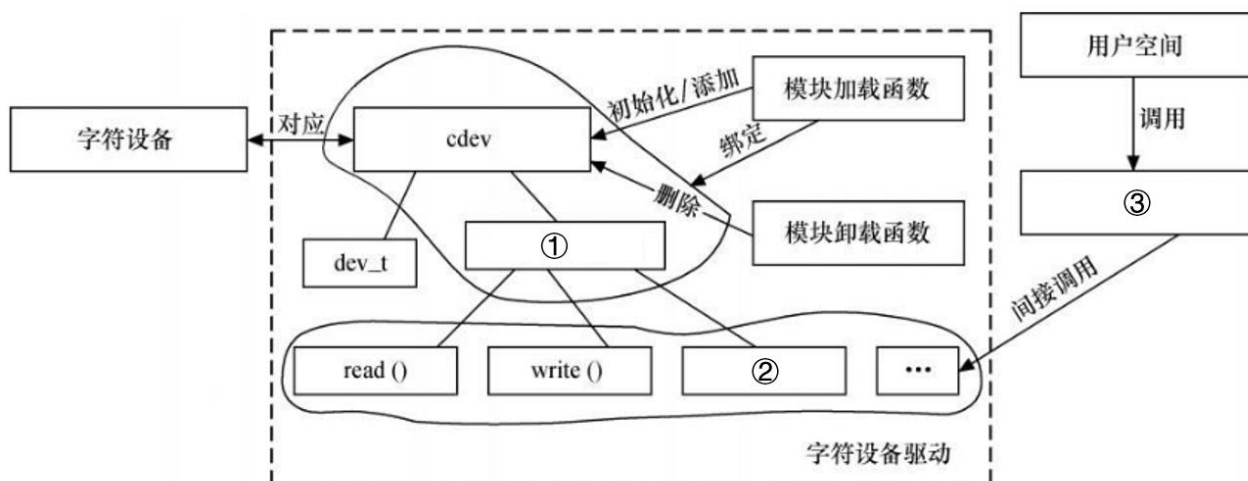
6. （2 分）在 Ubuntu 上执行 make 命令（交叉编译生成可执行文件）前，需要先执行“source /opt/poky/1.7/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi”命令，请问该命令的作用是什么？
7. （2 分）如果采用挂载方式运行程序，需要在“Xshell 2.0”上先执行 mount 命令（NFS 挂载命令）。假设 Ubuntu 的 IP 地址为 59.77.5.101，需要将 Ubuntu 的/imx6 目录挂载到实验箱的/mnt 目录，请写出 mount 命令。
8. （4 分）Android HelloWorld 工程可以在 4 个地方运行，请说出这 4 个地方的具体名称。
9. （3 分）简述在 IMX6 实验箱上开发（运行）Android NDK 程序的步骤。
10. （4 分）简述云（ModelArts）+ 端（Atlas 200 DK）协同垃圾分类实验的具体步骤。

#### 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？



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（①、②、③）的内容。



3. (5分)已知 Ubuntu 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。

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 (①、②、③) 的内容。

程序 1:

```
____①____ add                @声明 add 子程序将被外部函数调用
add:
    ADD r0,r0,r1
    MOV pc,lr
```

```
____②____ 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;
    ③ //声明内联汇编代码
    {
        MRS tmp, CPSR
        BIC tmp, tmp, #0x80
        MSR CPSR_c, tmp
    }
}
```

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) }
```

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) }
```

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)?

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 IMX6 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。