



# 厦门大学《嵌入式系统》课程期末试卷

软件学院 软件工程系 2016 级 软件工程专业

主考教师：曾文华

试卷类型：(A 卷)

一、 填空题（40 个空，每一空 1 分，共 40 分；在答题纸填写答案时请写上每个空格的对应编号）

- 1、嵌入式系统的前身通常称为 单片机 (1)。
- 2、最新的 ARM 处理器产品是 ARM Cortex-A 系列、ARM Cortex-R 系列、ARM Cortex-M 系列，其中 Cortex-A 系列又称为 高性能处理器 (2)，Cortex-R 系列又称为 实时处理器 (3)，ARM Cortex-M 系列又称为 低成本、低功耗 (4) 处理器。
- 3、 $\mu$ CLinux 的  $\mu$  是指 微 (5)，C 是指 控制 (6)。 $\mu$ CLinux 是专门针对没有 MMU (存储管理单元) (7) 的处理器设计的。
- 4、ARM 处理器的特权模式是指除 用户模式 (8) 外的其他六种模式。
- 5、Thumb 指令是 16 (9) 位的指令集，ARM 指令是 32 (10) 的指令集；Thumb 指令集是 ARM 指令集的一个 子集 (11)。
- 6、Cramfs 是专门针对闪存设计的 只读压缩 (12) 的文件系统。
- 7、Flash Memory (闪存) 有两种技术，分别是 NOR (13) Flash 和 NAND (14) Flash。
- 8、make 编译工具读取的文件名称为 Makefile (15) 文件。
- 9、UP-NetARM2410-S 实验箱（嵌入式 Linux 系统）启动后（即打开实验箱的电源开关，或者按下实验箱的 Reset 键），先执行 Boot Loader (16)，进行硬件和内存的初始化工作，然后加载 Linux 内核 (17) 和 根文件系统 (18)，完成 Linux 系统的启动。
- 10、常用的 Boot Loader 有：Blob (19)、U-boot (20) 和 vivi (21)。
- 11、加载模块的命令是 insmod (22)，卸载模块的命令是 rmmod (23)，查看模块的命令是 lsmod (24)。
- 12、Linux 的设备驱动程序开发调试有两种方法，第一种是直接编译到 内核 (25)；第二种是编译为 模块 (26) 的形式，单独加载运行调试。
- 13、查看设备文件的命令是：ls -l /dev (27)。
- 14、创建字符设备文件的命令是（假设设备名为/dev/lp0，主设备号为 6，次设备号为 0）：mknod /dev/lp0 c 6 0 (28)。

- 15、内核调试时，可以通过 \_\_printk (29) 函数，把关心的信息打印（显示）到终端或日志文件中。
- 16、UP-NetARM2410-S 实验箱 Linux 系统挂载的第一个文件系统是 根文件系统 (30)。
- 17、设备的控制操作是通过调用 file\_operations 结构体中的 \_\_ioctl() (31) 函数完成的。
- 18、查询设备中断号的命令是： \_\_cat \_\_/proc/interrupts (32)。
- 19、Qt/E 是著名的 Qt 开发商 TrollTech 发布的面向 嵌入式系统 (33) 的 Qt 版本。
- 20、UP-NetARM2410-S 实验箱的 CPU 为基于 ARM920T 内核的三星 S3C2410X (34) 芯片。
- 21、UP-NetARM2410-S 实验箱的 FLASH 存储器采用的是 SAMSUNG K9F1208 64M NAND，相当于普通 PC 机的 硬盘 (35)，UP-NetARM2410-S 实验箱的 SDRAM 存储器采用的是 2 片 HY57V561620AT-H 32M，相当于普通 PC 机的 内存条 (36)。
- 22、RS-485 数据信号采用差分传输方式，也称作平衡传输，它使用一对双绞线，将其中一线定义为 A，另一线定义为 B。通常情况下，发送驱动器 A、B 之间的正电平在+2V~+6V，是一个逻辑状态，表示 1 (37)，负电平在-2V~-6V，是另一个逻辑状态，表示 0 (38)。
- 23、如果需要将实验箱的 IP 地址设置为 192.168.1.34，其命令是： \_\_ifconfig eth0 192.168.1.34 (39)。
- 24、块设备驱动程序没有 read 和 write 操作函数，对块设备的读写是通过 请求函数 (40) 完成的。

**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、IP 核： **Intellectual Property，知识产权核，知识产权模块**
- 2、VHDL： **Very-High-Speed Integrated Circuit Hardware Description Language，超高速集成电路硬件描述语言**
- 3、CPSR： **程序状态寄存器，Current Program Status Register**
- 4、SPI： **串行外设接口，Serial Peripheral Interface**
- 5、MMU： **Memory Management Unit，存储器管理单元**
- 6、RISC： **Reduced Instruction Set Computer，精简指令集计算机**
- 7、Cramfs： **Compressed ROM File System，只读压缩的文件系统**
- 8、I<sup>2</sup>C（IIC，I2C）： **Inter Integrated-Circuit，内部集成电路总线**
- 9、NFS： **Network File System，网络文件系统**

10、DSP: Digital Signal Processor, 数字信号处理器

三、简答题（8 小题，共 25 分；在答题纸填写答案时请写上每小题的对应编号）

1、常见的嵌入式操作系统有哪些？（4 分）

答：

- 嵌入式 Linux
- VxWorks
- $\mu$ C/OS-II
- Windows CE
- Sysbian
- Android
- iOS
- QNX
- Palm OS
- LynxOS
- Nucleus
- ThreadX
- eCos

2、什么是嵌入式系统的交叉开发（交叉开发模式，交叉开发环境）？（3 分）

答：

即宿主机/目标机模式。宿主机为 PC 机；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。

3、MMU（Memory Management Unit）的主要作用是什么？（3 分）

答：

- （1）地址映射
- （2）对地址访问的保护和限制

4、请分别说出 UP-NetARM2410-S 实验箱的根文件系统、用户文件系统、临时文件系统采用的是什么格式的文件系统？（3 分）

答：

UP-NetARM2410-S 实验箱的根文件系统采用 Cramfs(压缩式的文件系统)格式，实验箱的用户文件系统采用 YAFFS（Yet Another Flash File System）格式，实验箱的临时文件系统采用 Ramfs（基于 RAM 的文件系统）格式。

5、虚拟机（Fedora）“NFS 服务”的功能是什么？（3 分）

答：

将虚拟机的文件夹（如：/arm2410s），设为 NFS 共享，在实验箱上执行 mount

命令，即可将虚拟机的文件夹 (/arm2410s)，共享到实验箱上的文件夹上（如：/host）。

6、串行通讯的方式有单工、半双工和全双工等 3 种方式，现有甲机和乙机要进行通讯，请描述这 3 种方式是怎么进行通讯的？（3 分）

答：

单工方式：甲机发送、乙机接收；或者，乙机发送、甲机接收。

半双工方式：甲机发送、乙机接收，乙机发送、甲机接收。

全双工方式：甲机、乙机同时发送和接收。

7、甲乙两台嵌入式设备都有 RS-485 接口，先要通过 RS-485 接口实现两台设备的通讯(采用半单工通讯方式),请问怎么连接两台设备的 RS-485 接口信号？（3 分）

答：甲设备 RS-485 接口信号 A (D+) 连接乙设备 RS-485 接口信号 A (D+)，甲设备 RS-485 接口信号 B (D-) 连接乙设备 RS-485 接口信号 B (D-)，两个设备 RS-485 接口信号 A、B 之间各接一个 120 欧姆的电阻。

8、请写出 ARM 指令的格式。（3 分）

答：

`<opcode> {<cond>} {S} <Rd>, <Rn> {, <shift_op2>}`

`<>`内的项是必须的，`{ }`内的项是可选的

opcode: 指令助记符（操作码），如 LDR, STR 等

cond: 执行条件（条件码），如 EQ, NE 等

S: 可选后缀，加 S 时影响 CPSR 中的条件码标志位，不加 S 时则不影响

Rd: 目标寄存器

Rn: 第 1 个源操作数的寄存器

op2: 第 2 个源操作数

shift: 位移操作

四、综合题（7 小题，共 25 分；在答题纸填写答案时请写上每小题的对应编号）

1、RS-232 通信程序在实验箱上执行时出现如下错误：

/dev/ttyS0: No such file or directory

经检查，实验箱中串口设备位于/dev/tts/下：

```
[/dev/tts]ls
```

```
0 1 2
```

请问执行什么命令，可以解决这个错误？（3 分）

答： `ln -sf /dev/tts/0 ttyS0`

2、设当前目录中有多线程应用程序的源程序 `pthread.c`，以及 `Makefile` 文件，`Makefile` 文件的具体内容如下：

```
CROSS = armv4l-unknown-linux-
CC = ${CROSS}gcc
LDLFLAGS +=
EXTRA_LIBS += -lpthread
EXP_INSTALL = install -m 755
OBJS = pthread.o
INSTALL_DIR = ../bin
EXEC = $(INSTALL_DIR)/pthread ./pthread
all: $(EXEC)
$(EXEC): $(OBJS)
$(CC) $(LDLFLAGS) -o $@ $(OBJS) $(EXTRA_LIBS)
install:
$(EXP_INSTALL) $(EXEC) $(INSTALL_DIR)
clean:
-rm -f $(EXEC) *.elf *.gdb *.o
```

请问在当前目录下执行 `make` 后，其结果是什么（屏幕上显示什么）？（4 分）

答：

```
armv4l-unknown-linux-gcc -c -o pthread.o pthread.c
armv4l-unknown-linux-gcc -o ../bin/pthread pthread.o -lpthread
armv4l-unknown-linux-gcc -o pthread pthread.o -lpthread
```

3、设当前目录中有 `demo` 模块驱动程序的源程序 `demo.c`，以及 `Makefile` 文件，`Makefile` 文件的具体内容如下：

```
KERNELDIR = /arm2410s/kernel-2410s
INCLUDEDIR = $(KERNELDIR)/include
CROSS_COMPILE = armv4l-unknown-linux-
AS          =$(CROSS_COMPILE)as
LD          =$(CROSS_COMPILE)ld
CC          =$(CROSS_COMPILE)gcc
CPP         =$(CC) -E
```

```

AR      =$(CROSS_COMPILE)ar
NM      =$(CROSS_COMPILE)nm
STRIP   =$(CROSS_COMPILE)strip
OBJCOPY =$(CROSS_COMPILE)objcopy
OBJDUMP =$(CROSS_COMPILE)objdump
CFLAGS += -I..
CFLAGS += -Wall -O -D__KERNEL__ -DMODULE -I$(INCLUDEDIR)
SOURCE = demo.c
TARGET = demo.o
all:$(TARGET)
$(TARGET):$(SOURCE)
    $(CC) -c $(CFLAGS) $^ -o $@
clean:
    rm -f *.o *~ core .depend

```

请问在当前目录下执行 make 目录后的结果是什么（屏幕上显示什么）？（4 分）  
答：

显示：

```

armv4l-unknown-linux-gcc -c -I.. -Wall -O -D__KERNEL__ -DMODULE
-I/arm2410s/kernel-2410s/include demo.c -o demo.o

```

4、以下程序为汇编语言调用 C 语言的例子：

_____	@声明要调用的 C 函数
MOV r0, 1	
MOV r1, 2	@通过 r0、r1 传递参数（参数传递规则）
BL add	@调用 C 函数 add；返回结果由 r0 带回 （子程序返回结果规则）

  

```

int  add (int x, int y)
{
    return(x+y);
}

```

请填写程序中空白（划线）的那一行。（3 分）

答：

```

IMPORT add

```

5、以下程序为 C 语言调用汇编语言的例子：

```

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

```

请填写程序中空白的那一行（划线的部分）。（3 分）

答：

`__asm`

6、设某个字符设备驱动程序的初始化函数程序如下：

```

1) int __init s3c2410_dcm_init(void)
2) {
3)     int ret;
4)     ret = register_chrdev(0, DEVICE_NAME, &s3c2410_dcm_fops);
5)     if (ret < 0) {
6)         DPRINTK(DEVICE_NAME " can't get major number\n");
7)         return ret;
8)     }
9)     dcmMajor=ret;
10) #ifdef CONFIG_DEVFS_FS
11)     devfs_dcm_dir = devfs_mk_dir(NULL, "dcm", NULL);
12)     devfs_dcm0 = devfs_register(devfs_dcm_dir, "0raw",
        DEVFS_FL_DEFAULT, dcmMajor, DCMRAW_MINOR, S_IFCHR |
        S_IRUSR | S_IWUSR, &s3c2410_dcm_fops, NULL);
13) #endif
14)     DPRINTK (DEVICE_NAME "\tdevice initialized\n");
15)     return 0;
16) }

```

请问上述程序中，第 4）行、第 11）行、第 12）行分别是做什么事情？（5 分）

答：

第 4）行：注册字符设备

第 11) 行: 创建设备文件

第 12) 行: 注册设备文件

7、以下为 A/D 转换设备驱动程序的读函数程序:

```
1) static ssize_t s3c2410_adc_read(struct file *filp, char *buffer, size_t count,
    loff_t *ppos)
2) {
3)     int ret = 0;
4)     if (down_interruptible(&adcdev.lock))
5)         return -ERESTARTSYS;
6)     START_ADC_AIN(adcdev.channel, adcdev.prescale);
7)     interruptible_sleep_on(&adcdev.wait);
8)     ret = ADCDAT0;
9)     ret &= 0x3ff;
10)    DPRINTK("AIN[%d] = 0x%04x, %d\n", adcdev.channel, ret, ADCCON
        & 0x80 ? 1:0);
11)    copy_to_user(buffer, (char *)&ret, sizeof(ret));
12)    up(&adcdev.lock);
13)    return sizeof(ret);
14) }
```

请问上述程序中, 第 11) 行是做什么事情? (3 分)

答:

第 11) 行: 从内核空间拷贝到用户空间。