

1、以下说法不正确的是 (B)。

- A、任务可以有类型说明      B、任务可以返回一个数值  
C、任务可以有形参变量      D、任务是一个无限循环

2 下列描述不属于 RISC 计算机的特点的是 (C)。

- A. 流水线每周期前进一步。    B. 更多通用寄存器。  
C. 指令长度不固定，执行需要多个周期。  
D. 独立的 Load 和 Store 指令完成数据在寄存器和外部存储器之间的传输。

3 存储一个 32 位数 0x2168465 到 2000H~2003H 四个字节单元中，若以大端模式存储，则 2000H 存储单元的内容为 (D)。

- A、0x21      B、0x68      C、0x65      D、0x02

4  $\mu$ COS-II 中对关键代码段由于希望在执行的过程中不被中断干扰，通常采用关中断的方式，以下 X86 汇编代码正确而且不会改变关中断之前的中断开关状态的是 (D)

- A. 先 CLI、执行关键代码、再 STI  
B. 先 STI、执行关键代码、再 CLI  
C. 先 POPF、CLI、执行关键代码、再 PUSHF  
D. 先 PUSHF、CLI、执行关键代码、再 POPF。

5 RS232-C 串口通信中，表示逻辑 1 的电平是 (D)。

- A、0v      B、3.3v      C、+5v~+15v      D、-5v~-15v

6 ARM 汇编语句“ADD R0, R2, R3, LSL#1”的作用是 (A)。

- A.  $R0 = R2 + (R3 \ll 1)$       B.  $R0 = (R2 \ll 1) + R3$   
C.  $R3 = R0 + (R2 \ll 1)$       D.  $(R3 \ll 1) = R0 + R2$

7 IRQ 中断的入口地址是 (C)。FIQ 的入口地址为 0x0000001C

- A、0x00000000      B、0x00000008      C、0x00000018      D、0x00000014

8 S3C2420X I/O 口常用的控制器是 (D)。

- (1) 端口控制寄存器 (GPACON-GPHCON)。  
(2) 端口数据寄存器 (GPADAT-GPHDAT)。  
(3) 外部中断控制寄存器 (EXTINTN)。  
(4) 以上都是。

9 实时操作系统中，两个任务并发执行，一个任务要等待其合作伙伴发来信息，或建立某个条件后再向前执行，这种制约性合作关系被成为 (A)。

- A. 同步    B. 互斥    C. 调度    D. 执行

10 和 PC 系统机相比嵌入式系统不具备以下哪个特点 (C)。

- A、系统内核小    B、专用性强    C、可执行多任务    D、系统精简

11、ADD R0, R1, #3 属于 (A) 寻址方式。

- A. 立即寻址    B. 多寄存器寻址    C. 寄存器直接寻址    D. 相对寻址

12、GET 伪指令的含义是 (A)

- A. 包含一个外部文件    B. 定义程序的入口    C. 定义一个宏    D. 声明一个变量

13、存储一个 32 位数 0x876165 到 2000H~2003H 四个字节单元中，若以小端模式存储，则 2000H 存储单元的内容为 (C)。

- A、0x00    B、0x87    C、0x65    D、0x61

14、 $\mu$ COS-II 操作系统不属于 (C)。

- A、RTOS    B、占先式实时操作系统    C、非占先式实时操作系统    D、嵌入式实时操作系统

15、若  $R1=2000H$ ,  $(2000H)=0x86$ ,  $(2008H)=0x39$ , 则执行指令 LDR R0, [R1, #8]! 后 R0 的值为 (D)。

A. 0x2000      B. 0x86      C. 0x2008      D. 0x39

16、寄存器 R13 除了可以做通用寄存器外，还可以做 (C)。

A. 程序计数器      B. 链接寄存器      C. 栈指针寄存器      D. 基址寄存器

17、FIQ 中断的入口地址是 (A)。

A. 0x0000001C      B. 0x00000008      C. 0x00000018      D. 0x00000014

18、ARM 指令集和 Thumb 指令集分别是 (D) 位的。

A. 8 位, 16 位      B. 16 位, 32 位      C. 16 位, 16 位      D. 32 位, 16 位

19、ARM 寄存器组有 (D) 个寄存器。

A. 7      B. 32      C. 6      D. 37

20、若 R1=2000H, (2000H)=0x28, (2008H)=0x87, 则执行指令 LDR R0, [R1, #8]! 后 R0 的值为 (D)。

A. 0x2000      B. 0x28      C. 0x2008      D. 0x87

21、寄存器 R15 除了可以做通用寄存器外，还可以做 (A)

A. 程序计数器      B. 链接寄存器      C. 堆栈指针寄存器      D. 基址寄存器

22、嵌入式系统有硬件和软件部分构成，以下 (C) 不属于嵌入式系统软件。

A. 系统软件      B. 驱动      C. FPGA 编程软件      D. 嵌入式中间件

26、假设 R1=0x31, R2=0x2 则执行指令 ADD R0,R1,R2 LSL #3 后, R0 的值是 (C)

A. 0x33      B. 0x34      C. 0x39      D. 0x38

23、和 PC 机系统相比下列哪个不是嵌入式系统独具的特点 (C)

A. 系统内核小      B. 专用性强      C. 可执行多任务      D. 系统精简

24、Unicode 编码与 (A) 编码方式兼容。

A. ASCII 码      B. GBK      C. GB2312      D. 区位码

25、Cache 用于存放主存数据的部分拷贝，主存单元地址与 Cache 单元地址之间的转换用 (A) 完成。

A. 硬件      B. 软件      C. 用户      D. 程序员

26 在  $\mu$ C/OS-II 系统中，OSTimeTick() 函数只被以下 (A) 函数或过程所调用。

A. OSTickISR      B. OSShed      C. OSCtxSw      D. OSIntCtxSw

27 每种嵌入式操作系统都有自身的特点以吸引相关用户，下列说法错误的是 (D)。

A. 嵌入式 Linux 提供了完善的网络技术支持；

B.  $\mu$ CLinux 是专门为没有 MMU 的 ARM 芯片开发的；

C.  $\mu$ C/OS-II 操作系统是一种实时操作系统 (RTOS)；

D. WinCE 提供完全开放的源代码。

28 下列关于存储管理单元 (MMU) 说法错误的是 (B)。

A. MMU 提供的一个关键服务是使各个任务作为各自独立的程序在其自己的私有存储空间中运行。

B. 在带 MMU 的操作系统控制下，运行的任务必须知道其他与之无关的任务的存储需求情况，这就简化了各个任务的设计。

C. MMU 提供了一些资源以允许使用虚拟存储器。

D. MMU 作为转换器，将程序和数据的虚拟地址 (编译时的连接地址) 转换成实际的物理地址，即在物理主存中的地址。

29 下列 CPSR 寄存器标志位的作用说法错误的是 (D)。

A. N: 负数      B. Z: 零      C. C: 进位      D. V: 借位

30 S3C2410X 处理器为用户进行应用设计提供了支持多主总线的 IIC 接口。处理器提供符合 IIC 协议的设备连接的串行连接为 (C)。

A. SCL 和 RTX      B. RTX 和 RCX      C. SCL 和 SDA      D. SDA 和 RCX

31 RQ 中断的优先级是 ( D )。

A. 1      B. 2      C. 3      D. 4

32 LDR R2, [R3,#128] 的寻址方式是 ( C )。

A. 寄存器寻址      B. 寄存器移位      C. 变址寻址      D. 间接寻址

33 S3C2410 有 ( B ) 个 USB 接口。

A. 1      B. 2      C. 3      D. 4

34 ARM 处理器的工作模式有 ( C ) 种。

A. 5      B. 6      C. 7      D. 8

## 一、 判断题

- 1、程序运行过程中的资源可以是显示器、键盘，但不可以是一个数组或变量 (X)
- 2、在μCOS-II 操作系统中，已知 Task1 的优先级为 12，Task2 的优先级为 26。假如在 Task2 运行过程中发生键盘中断，在执行中断服务程序时 Task1 进入就绪状态，则中断返回时 Task1 得到 CPU 的使用权。(√)
- 3、若定义任务 task1 的优先级为 12，则不能再定义其他任务的优先级也为 12 (√)
- 4、任务创建后即开始调度 (X)
- 5、液晶工作在文本模式下时，既可以显示英文字符也可以显示汉字 (X)
- 6、处于就绪状态的任务只能切换到运行状态 (X)
- 7、等待消息函数 WaitMessage (0) 中的“0”表示此函数的超时等待时间为 0 秒 (X)
- 8、若 OSTaskSuspend (prio)函数把正在延时的任务挂起，则调用一次对应的唤醒函数即可使该任务切换到就绪状态。(X)
- 9、调试程序时，需要停止程序运行后才可查看变量内容 (√)
- 10、μCOS-II 操作系统中移植成功后即可使用文件系统 (X)
- 11、将模拟信号转换成数字信号的电路是 DA 转换器。(×)
- 12、IIC 是同步串行数据总线。(√)
- 13、S3C2410 的处理器对内存的访问只能通过 Load/Store 指令来实现。(√)
- 14、Nand Flash 比 Nor Flash 成本高，可靠性差。(×)
- 15、C 语言程序可以嵌套加入汇编程序模块。(√)
- 16、运算放大器的输出电压范围一般大于电源电压的范围。(×)
- 17、由于 CPU 内部寄存器的访问速度较高，根据 ATPC 标准，应尽可能使函数的参数控制在 4 个以下。(√)
- 18、ARM 处理器结构体系中具有 T 变种处理器核可以工作在 ARM 状态和 Thumb 状态。(√)

## 二、 简答题

ARM 体系结构的寄存器组织：37 个 32 位寄存器，其中 31 个为通用寄存器，6 个为状态寄存器，哪些寄存器是可编程访问的，取决微处理器的工作状态及具体的运行模式。

ARM 微处理器的异常状态：7 种异常

**4、ARM 处理器有几种工作模式？并做说明每种工作模式的含义。**

答：ARM 处理器有 7 种工作模式：

用户模式(usr)- 正常程序执行的模式	0.5 分
快速中断模式(fiq)- FIQ 异常响应时进入此模式	0.5 分
中断模式(irq)- IRQ 异常响应时进入此模式	0.5 分
管理员模式(svc)- 系统复位和软件中断响应时进入此模式	0.5 分
中止模式(abt)- 用于虚拟存储及存储保护	0.5 分
系统模式(sys)- 与用户类似，但有直接切换到其它模式等特权	0.5 分

未定义模式(und)- 未定义指令异常响应时进入此模式 0.5 分  
除了用户模式外，其他模式均可视为特权模式 0.5 分

三、 阅读程序题

9、 阅读下列与看门狗有关的寄存器描述，解释每一行代码的功能。

看门狗定时器控制寄存器（WTCON）

寄存器	地 址	读/写	描 述	初 始 值
WTCON	0x53000000	读/写	看门狗定控制寄存器	0x8021

WTCON 的标识位

WTCON	Bit	描 述	初 始 值
Prescaler Value	[15:8]	预装比例值，有效范围值为 0~255	0x80
Reserved	[7:6]	保留	00
Watchdog Timer	[5]	使能和禁止看门狗定时器 0 = 禁止看门狗定时器 1 = 使能看门狗定时器	0
Clock Select	[4:3]	这两位决定时钟分频因素 00:1/16                      01:1/32 10:1/64                      11:1/128	00
Interrupt Generation	[2]	中断的禁止和使能 0=禁止中断产生 1=使能中断产生	0
Reserved	[1]	保留	0
Reset Enable/Disable	[0]	禁止很使能看门狗复位信号的输出 1=看门狗复位信号使能 0=看门狗复位信号禁止	1

看门狗定时器数据寄存器（WTDAT）

寄 存 器	地 址	读/写	描 述	初 始 值
WTDAT	0x53000004	读/写	看门狗数据寄存器	0x8000

看门狗计数寄存器（WTCNT）

寄 存 器	地 址	读/写	描 述	初 始 值
WTCNT	0x53000008	读/写	看门狗计数器当前值	0x8000

```
#define rWTCON    (*(volatile unsigned *)0x53000000) // 第1行
#define rWTDAT    (*(volatile unsigned *)0x53000004) // 第2行
#define rWTCNT    (*(volatile unsigned *)0x53000008) // 第3行
void watchdog_test(void)
{
    rWTCON = ((PCLK/1000000-1)<<8)|(3<<3)|(1<<2);           // 第4行
    rWTDAT = 7812;                                           // 第5行
    rWTCNT = 7812;                                           // 第6行
    rWTCON |= (1<<5);                                       // 第7行
}
```

第1-3 行:定义看门狗控制寄存器、数据寄存器和计数寄存器为rWTCON、rWTDAT和rWTCNT。

第4 行: 设置看门狗的预装比例值为 1000000, 分频因素为 1/128, 并使能中断。(6 分)

第5-6 行: 对数据寄存器和计数寄存器赋值为 7812。(2 分)

第7 行: 启动看门狗。(2 分)

7、阅读以下S3C2410部分用户手册。求: 当PCLK 或 UCLK 为40 MHz时, 串口0 的波特率为 2 4 0 0 bps , 串口1 的波特率为 1 1 5 2 0 0 bps, 相应的控制寄存器如何设置。

#### UART BAUD RATE DIVISOR REGISTER

There are three UART baud rate divisor registers (寄存器) including UBRDIV0, UBRDIV1 and UBRDIV2 in the UART block (模块). The value stored in the baud rate divisor register (UBRDIVn), is used to determine the serial Tx/Rx clock rate(baud rate) as follows:

$$\text{UBRDIVn} = (\text{int})(\text{PCLK} / (\text{bps} \times 16)) - 1 \quad \text{or} \quad \text{UBRDIVn} = (\text{int})(\text{UCLK} / (\text{bps} \times 16)) - 1$$

Where, the divisor should be from 1 to (216-1) and UCLK should be smaller than PCLK.

Register	Address	R/W	Description	Reset Value
UBRDIV0	0x50000028	R/W	Baud rate divisor register 0	—
UBRDIV1	0x50004028	R/W	Baud rate divisor register 1	—
UBRDIV2	0x50008028	R/W	Baud rate divisor register 2	—
UBRDIVn	Bit	Description		Initial State
UBRDIV	[15:0]	Baud rate division value UBRDIVn >0		—

答: 根据 $\text{UBRDIVn} = (\text{int})(\text{PCLK} / (\text{bps} \times 16)) - 1$

寄存器  $\text{UBRDIV0} = (\text{int})(40000000/2400*16)-1=1040=10000010000(\text{B})$

寄存器  $\text{UBRDIV1} = (\text{int})(40000000/115200*16)-1=20=10100(\text{B})$

任务控制块(TCB)的数据结构 OS-TCB 表示任务的状态。当任务的 CPU 控制权被抢占后, 任务控制块用来保存该任务的状态。当任务重新获得 CPU 的控制权后, 任务控制块能够保证任务从被中断的位置继续正确执行。任务控制块提供了任务的基本信息, 如任务的执行状态、优先级和堆栈位置等。

```
1 .text
2 .global _start
3 _start:
4         LDR    R0, =0x56000010    @R0 设为 GPBCON 寄存器。此寄存器
                                     @用于选择端口 B 各引脚的功能:
                                     @是输出、是输入、还是其他

5         MOV    R1, #0x00004000
6         STR    R1, [R0]           @设置 GPB7 为输出口

7         LDR    R0, =0x56000014    @R0 设为 GPBDAT 寄存器。此寄存器
                                     @用于读/写端口 B 各引脚的数据

8         MOV    R1, #0x00000000    @此值改为 0x00000080,
                                     @可让 LED1 熄灭
9         STR    R1, [R0]           @GPB7 输出 0, LED1 点亮

10 MAIN_LOOP:
11         B      MAIN_LOOP
```

## 一、 填空题(每空 2 分，共 40 分)

得分	评阅人

- 1、嵌入式系统的设计可以分成三个阶段：分析、设计和实现
- 2、目前使用的嵌入式操作系统主要有 Windows CE/Windows Mobile Linux、uCos、和 Symbian
- 3、XScale 微处理器使用的是 ARM 公司 ARMV5TE 版内核和指令集。
- 4、微处理器有两种总线架构,使用数据和指令使用同一接口的是冯诺依曼，分开的指令和数据接口、取指和数据访问可以并行进行的是哈佛结构
- 5、ARM 微处理器有七种工作模式，它们分为两类 非特权模式、特权模式。其中用户模式属于非特权模式
- 6、ARM 核有两个指令集，分别是ARM、Thumb
- 7、ARM 微处理器复位后，PC（R15）的地址通常是0X0，初始的工作模式是supervisor
- 8、在 ARM 体系构架中对复杂的内存管理是通过系统控制协处理器cp15和 MMU（存储管理部件）来进行的。当系统发生Data Abort（数据）异常和Prefetch Abort（指令领取）异常时，异常处理程序透过嵌入式操作系统的内存管理机制，通过 MMU 交换物理内存和虚拟内存的页面，以保证程序正常执行。

9、构建嵌入式系统开发环境的工具链有多种，其中开放源码的工具链是 GNU 工具链，ARM 公司提供的工具链是 ADS 工具链

## 二、指令和程序测试题(共 24 分)

1、写一条 ARM 指令，完成操作  $r1 = r2 * 4$  (4 分)

```
MOV R1,R2,  
LSL #3
```

2、初始值 R2=5, R3=4, R4=3, 执行指令 SUBS R2, R3, R4, LSR #2 后, 寄存器 R2, R3 的值分别是多少? (4 分)

R2=3,R3=4

3、有如下程序段，画出程序流程图，并回答执行程序以后 R0 的值是多少。(8 分)

```
num      EQU      2  
start  
        MOV      r0, #1  
        MOV      r1, #6  
        MOV      r2, #3  
        BL       arithfunc  
stop  
        B        stop  
arithfunc  
        CMP      r0, #num  
        MOVHS    pc, lr  
        ADR      r3, JumpTable  
        LDR      pc, [r3, r0, LSL#2]  
JumpTable  
        DCD      (.WORD)      DoA    ; 注意：使用 ADS 开发工具用伪指令 DCD  
        DCD      (.WORD)      DoS    ;      使用 GNU 开发工具用伪指令 .WORD  
DoA  
        ADD      r0, r1, r2  
        MOV      pc, lr  
DoS  
        SUB      r0, r1, r2  
        MOV      pc, lr  
END
```

答：

R0 为 0 执行 DoA, R0 为 0 执行 DoS, 流程图略。R0=5。

4、有程序段如下，实现下面流程图，试补充编写 ARM 汇编代码。(8 分)



程序段：

```
MOV r0, #15
```

```
MOV r1, #9
```

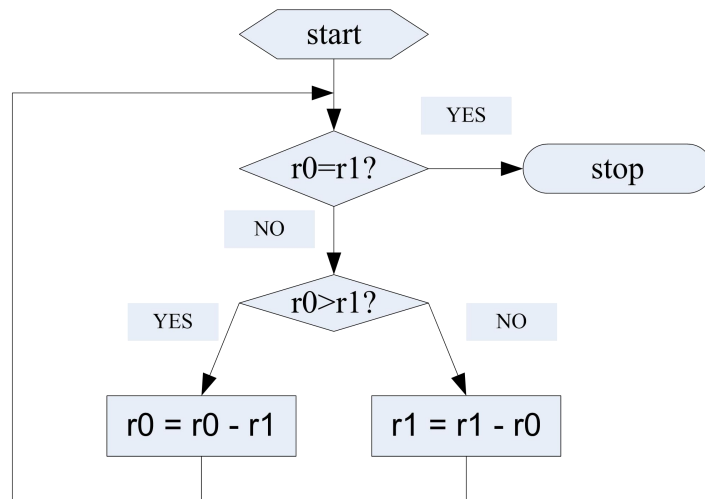
start

;补充代码

stop

```
B stop
```

```
END
```



程序：

Start:

```
CMP r0, r1
```

```
SUBLT r1,r1,r0
```

```
SUBGT r0,r0,r1
```

```
BNE start
```

## 二、 问答题（每题 6 分，共 36 分）

1、嵌入式开发环境主要包括哪些组件？

答：嵌入式系统开发需要交叉编译和在线调试的开发环境，主要包括

- 宿主机
- 目标机（评估电路板）
- 基于 JTAG 的 ICD 仿真器、或调试监控软件、或在线仿真器 ICE
- 运行于宿主机的交叉编译器和链接器、以及开发工具链或软件开发环境
- 嵌入式操作系统

2、ARM 核中什么寄存器用于存储 PC？R13 通常用来存储什么？R14 通常用来存储什么？

答：R15 用于程序计数寄存器 PC，R13 通常用来做堆栈指针寄存器，R14 通常用来做链接寄存器，保存函数调用的返回地址

### 3、Boot Loader 在嵌入式系统中主要起什么作用？完成哪些主要的工作？

答：Boot Loader 是在嵌入式系统复位启动时，操作系统内核运行前，执行的一段程序。通过 Boot Loader，初始化硬件设备，建立内存和 I/O 空间映射图，为最终加载操作系统内核调整好适当的系统软硬件环境。

### 4、简述嵌入式系统的概念、组成及特点。

答：嵌入式系统是以应用为中心，以计算机技术为基础，采用可剪裁软硬件，适用于对功能、可靠性、成本、体积、功耗等有严格要求的专用计算机系统。一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户的应用程序等四个部分组成。其特点有

- 嵌入式系统通常是面向特定应用的
- 嵌入式系统是将先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合后的产物
- 嵌入式系统的硬件和软件都必须高效率地设计，量体裁衣、去除冗余
- 嵌入式系统和具体应用有机地结合在一起，它的升级换代也是和具体产品同步进行
- 为了提高执行速度和系统可靠性，嵌入式系统中的软件一般都固化在存储器芯片或单片机本身中
- 嵌入式系统本身不具备自举开发能力

### 5、搭建嵌入式开发环境，连接目标板，一般使用什么通信接口连接？在 Windows 主机上使用什么软件建立连接？在 Linux 主机上使用什么软件建立连接？

答：RS-232，以太网口、并口  
在 Windows 主机上使用超级终端软件  
在 Linux 主机上使用 Minicom 软件

### 6、进行基于 ARM 核的嵌入式系统软件开发时，调用如下函数：

```
int do_something(int arg1,void *arg2,char arg3,int *arg4)
```

则这四个参数通过什么方式从调用程序传入被调函数？

答：基于 ARM 核的嵌入式系统软件开发时，调用函数和子程序通过 R0——R3 四个寄存器传递参数，超过四个参数使用堆栈传递。因此 arg1 通过 R0 传入，arg2，通过 R1 传入，arg3 通过 R2 传入，arg4 通过 R3 传入。

### 7、简述再进行基于 ARM 核的嵌入式系统软件开发时，调用如下函数的规则约定。

## 一、填空题（每空2分，共18分）

- 1、PC机开机的初始化工作一般由BIOS完成，而嵌入式系统的初始化工作一般由\_\_\_\_\_完成。
- 2、ARM内核三级流水线机制是指指令运行过程中经过的\_\_\_\_\_、译码、执行三个阶段。
- 3、ARM寄存器SPSR被称为\_\_\_\_\_寄存器。
- 4、uClinux与标准Linux的最大区别在于\_\_\_\_\_。
- 5、嵌入式系统是以应用为中心、以\_\_\_\_\_为基础、软硬件可\_\_\_\_\_、功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。
- 6、ARM的数据存取指令Load/Store是唯一用于寄存器和存储器之间进行的指令。
- 7、μC/OS中，为保护任务之间的共享数据和提供任务之间的通信，提供了操作系统任务间通信方法有：\_\_\_\_\_、邮箱、\_\_\_\_\_，事件标志。

## 二、单项选择题（每题2分，共24分）

- 1、下列哪一项不属于嵌入式系统软件结构一般包含的四个层面。（     ）  
A. 嵌入式处理器                      B. 实时操作系统（RTOS）  
C. 应用程序接口（API）层        D. 实际应用程序层
- 2、一条ARM数据处理指令，例如“ADD R0, R1, #immed”中使用的立即数 #immed是有一定限制的，下列立即数表示中合法的是（     ）。  
A. 0x00001234        B. 0x F000000F        C. 0x0000F008        D. 0x F080000
- 3、μCOS-II 操作系统不属于（     ）。  
A、RTOS                              B、占先式实时操作系统  
C、非占先式实时操作系统        D、嵌入式实时操作系统
- 4、ARM汇编语句“ADD R0, R2, R3, LSL#1”的作用是（     ）。  
A.  $R0 = R2 + (R3 \ll 1)$   
B.  $R0 = (R2 \ll 1) + R3$   
C.  $R3 = R0 + (R2 \ll 1)$

D.  $(R3 \ll 1) = R0 + R2$

5、ARM 寄存器组有 ( ) 个寄存器。

B、 32

D、 37

6、 每种嵌入式操作系统都有自身的特点以吸引相关用户，下列说法错误的是（ ）。

A. 嵌入式 Linux 提供了完善的网络技术支持;

B.  $\mu$ CLinux 是专门为没有 MMU 的 ARM 芯片开发的;

C.  $\mu$ C/OS-II 操作系统是一种实时操作系统 (RTOS);

D. WinCE 提供完全开放的源代码。

7、存储一个 32 位数 0x2168465 到 2000H~2003H 四个字节单元中，若以大端模式存储，则 2000H 存储单元的内容为（ ）。

B、0x68

D、 0x02

8、下列不是嵌入式系统的特点的是 ( )

A. 嵌入式系统需要专用开发工具和方法进行设计。

B. 嵌入式系统是技术密集、资金密集、高度分散、不断创新的知识集成系统。

C. 嵌入式系统使用的操作系统一般不是实时操作系统 (RTOS)，系统不具有实时约束。

D. 嵌入式系统通常是面向特定任务的，而不同于一般通用 PC 计算平台，是“专用”的计算机系统

9、寄存器 R15 除了可以做通用寄存器外，还可以做（ ）

## B. 链接寄存器

#### D. 基址寄存器

10、若 R1=2000H, (2000H)=0x28, (2008H)=0x87, 则执行指令 LDR R0, [R1, #8]! 后 R0 的值为 ( )。

### B. 0x28

D. 0x87

- 11、FIQ 中断的入口地址是 ( )。
- A、0x0000001C                      B、0x00000008
- C、0x00000018                      D、0x00000014
- 12、ARM 指令集是 ( ) 位的。
- A. 16                                  B. 32
- C. 8                                    D. 24

### 三、判断题 (每题2分, 共14分)

- 1、若定义任务 task1 的优先级为 12, 则不能再定义其他任务的优先级也为 12。  
( )
- 2、ARM 处理器在未定义模式下可以处理存储器故障, 实现虚拟存储器和存储器保护。  
( )
- 3、μCOS-II 操作系统中移植成功后即可使用文件系统。  
( )
- 4、在 ARM 的数据处理指令中, 大多数均可在指令助记符后加 S 来设置状态寄存器的条件码。  
( )
- 5、ARM 和 Thumb 之间状态的切换不影响处理器的模式或寄存器的内容。  
( )
- 6、当操作数寄存器Rm的状态位bit [0] 为1时, 执行BX Rm指令就可从ARM状态进入Thumb状态。  
( )
- 7、ARM 处理器结构体系中具有 T 变种的处理器核可以工作在 ARM 状态和 Thumb 状态。  
( )

### 四、简答题 (共16分)

- 1、ARM异常发生时, ARM内核对异常的响应过程是怎样的? (4分)
- 2、什么是 BootLoader? 主要有几种工作模式及主要功能是什么? (6 分)
- 3、S3C44B0 的存储系统中, 0xc000000 和 0xc080000 的含义是什么? 它们之间的 512kbyte 空间做什么用? (6 分)

五、分析设计（共28分）

1、改错和注释（10 分）

题目要求：

- （1）阅读下面的程序段，找出程序段中至少 3 处不规范的地方，并改正。（6 分）
- （2）给标记①~④处的语句作注释，说明其作用。（4 分）

```

        AREA      INT, CODE, READONLY
        ENTRY
start    LDR      R1, =SRCSTR
        LDR      R0, =dststr
        BL       strcpy                ; ①

STOP:    B       STOP

strcpy
        .....（此处表示省略的程序段）

        Mov      PC,  LR                ; ②

srcstr   AREA     Strings, DATA, READWRITE ; ③
dststr   DCB     "First string - source",0
        Space   100

        END                                ; ④
```

2、已知 S3C44B0 内核频率为 MCLK = 18.432MHz，ULCON0 是串口 0 的行控制寄存器，UBRDIV0 是串口 0 的波特率配置寄存器。请设置寄存器 ULCON0 和 UBRDIV0，使得 S3C44B0 的串口 0（UART0）的数据帧格式为：普通模式，波特率 115200，奇偶校验无，数据位 8 位，停止位 1 位。（6 分）

ULCONn	Bit	Description
Reserved	[7]	
红外线模式	[6]	是否采用红外通讯模式 0 = Normal mode operation 1 = Infra-Red Tx/Rx mode
奇偶校验模式	[5:3]	奇偶校验位设置 0xx = No parity 100 = Odd parity 101 = Even parity 110 = Parity forced/checked as 1 111 = Parity forced/checked as 0
停止位的数量	[2]	每帧中停止位的个数 0 = One stop bit per frame 1 = Two stop bit per frame
数据位长度	[1:0]	每帧中数据位的个数 00 = 5-bits    01 = 6-bits 10 = 7-bits    11 = 8-bits

3、在图中进行必要的连线，管脚电平设置，地址编号填写等，完成 S3C44B0 最小系统相关设计。（12 分）

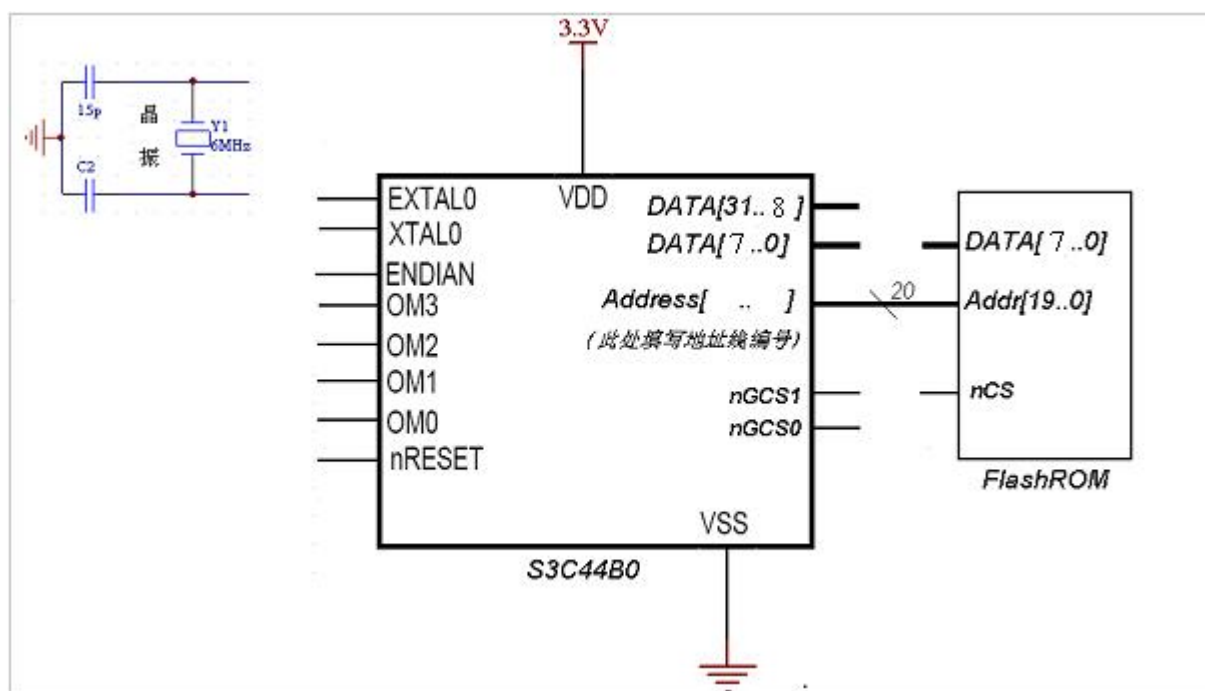
题目要求：1) 使 S3C44B0 的 BANK0 连接一片 FlashROM，且总线宽度为 8 位。

（5 分）

2) 选择晶振作为时钟输入。存储器以小端模式存储字。（4 分）

3) 画出复位电路图，实现上电复位功能和手动按键复位功能。（3 分）

（注：不使用的管脚打×）



### 试题参考答案及评分标准

#### 一、填空题（每空2分，共18分）

1、bootloader。2、取指 3、备份的程序状态寄存器。 4、内存管理。

5、计算机技术；裁剪。 6、数据传送。 7、信号量；消息队列。

#### 二、单项选择题（每题2分，共24分）

1、 A      2、 B      3、 C      4、 A      5、 D      6、 D      7、 D  
8、 C

9、 A      10、 D      11、 A      12、 B

### 三、判断题（每题2分，共14分）

1、 √    2、 X    3、 X    4、 √    5、 √    6、 √    7、 √

### 三、简答题（共16分）

1、（4分）

答：1，保存 CPSR 到 SPSR\_<mod>（1’）

2，设置 CPSR 相应位（1’）

以进行模式切换，根据需要，设置 IRQ 和 FIQ 中断屏蔽位

3，保存异常返回地址到 R14\_<mod>（1’）

4，PC 跳转到异常向量表相应地址处（1’）

2、（6分）

答：简单说来，Bootloader 就是操作系统内核运行的一段小程序，完成进行初始化系统硬件设置的任务，包括 CPU、SDRRAM、Flash、串口等初始化，时钟的设置、存储器的映射。

分为启动加载模式和下载模式。

（1）启动加载（Boot loading）模式又称为“自主”（Autonomous）模式，是指 Bootloader 从目标机上的某个固态存储设备上将操作系统加载到 RAM 中运行，整个过程并没有用户的介入。

（2）在下载模式下，目标机上的 Bootloader 将先通过串口连接或网络连接等通信手段从宿主机下载文件。

3、（6分）

答：0xC0000000 是 SDRAM 的起始地址，系统的程序存储空间从 0xC0800000 开始。512Kbyte 的空间划分出来，作为系统的 LCD 显示缓冲区使用（更新其中的数据，就可以更新 LCD 的显示）。

### 四、分析设计（共28分）

1、（10分）

（1）（6分）

start    LDR    R1, =SRCSTR    ; 此处 start 要顶格书写；SRCSTR 要小写

STOP:    B    STOP    ; 此处 STOP 后应无“:”

Mov    PC, LR    ; 此处 Mov 应大小写一致

（2）（4分）

①跳转到子程序 strcpy

②子程序返回

③定义一个数据段 Strings 读写属性



#### ④结束汇编

2、(6分)

(1)  $ULCON0=0 \times 3$

(2)  $UBRDIV0=(\text{round\_off})[\text{MCLK}/(\text{bsp} \times 16)]-1$

$=(\text{round\_off})[18432000/(115200 \times 16)]-1=10-1=9$

3、(12分)

(1) BANK08, 位数据总线:  $OM1=0, OM0=0$  ---- (1'),

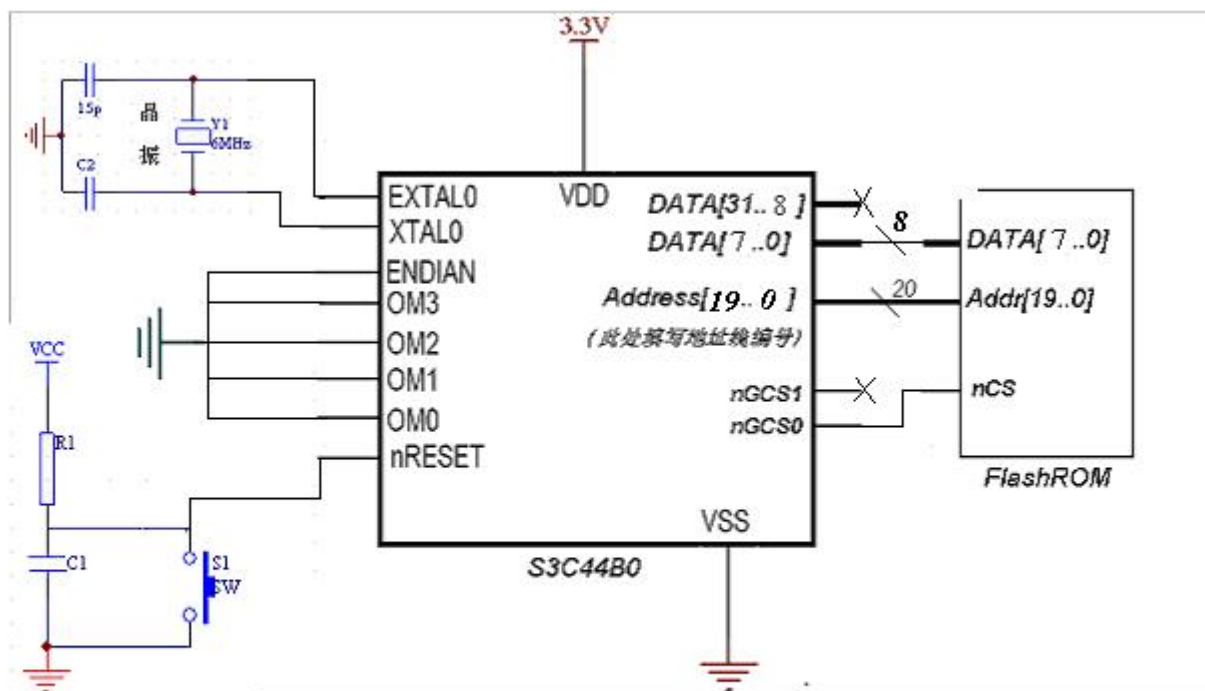
DATA[7..0] 连接 DATA[7..0] (1')

nGCS0 连接 nCS ---- (1'), Address[19..0] (2')

(2) 晶振连线:  $OM3=0, OM2=0$  ---- (1') 晶振连接 EXTAL0 和 XTAL0 ---- (1')

小端模式  $ENDIA=0$  ---- (2')

(3) 复位电路: 上电低电平复位 (2') 按键复位 (1')



#### 一. 选择题

1. 以下哪个不是 RISC 架构的 ARM 微处理器的一般特点: (C)

- A 体积小、低功耗                      B 大量使用寄存器  
C 采用可变长度的指令格式，灵活高效    D 寻址方式灵活简

2. 通常所讲的交叉编译就是在 X86 架构的宿主机上生成适用于 ARM 架构的（ A ）格式的可执行代码。

- A elf            B exe            C pe            D sh

3. 下面不属于 Boot Loader 阶段 1 所完成的步骤的是：（ C ）

- A. 硬件设备初始化。  
B. 拷贝 Boot Loader 的阶段 2 到 RAM 空间中。  
C. 将 kernel 映像和根文件系统映像从 Flash 读到 RAM 空间中。  
D. 设置堆栈。

4. 以下哪个不是 ARM 的 7 种运行状态之一：（ B ）

- A 快中断状态 B 挂起状态 C 中断状态 D 无定义状态

6. 用以下的哪个命令可以把 server 的/tmp mount 到 client 的/mnt/tmp 并且是（ D ）

- A .mount -o ro server:/tmp /mnt/tmp  
B. mount -o ro /mnt/tmp server:/tmp  
C .mount -o ro client:/mnt/tmp server:/tmp  
D.mount -o ro server:/tmp client:/mnt/tmp

7. 以下叙述中，不符合 RISC 指令系统特点的是( B )。

- A. 指令长度固定，指令种类少  
B. 寻址方式种类丰富，指令功能尽量增强  
C. 设置大量通用寄存器，访问存储器指令简单

D. 选取使用频率较高的一些简单指令

8. 通常所说的 32 位微处理器是指( C )。

A. 地址总线的宽度为 32 位      B. 处理的数据长度只能为 32 位

C. CPU 字长为 32 位      D. 通用寄存器数目为 32 个

9.ADD R0,R1,[R2]属于 (B)。

A.立即寻址   B 寄存器间接寻址   C。寄存器寻址   D。 基址变址寻址

10、 ADD R0,R0,#1 属于 (A)

A.立即寻址   B。寄存器间接寻址   C。寄存器寻址   D。 基址变址寻址

11.典型的计算机系统结构是 (A)

A 冯诺依曼体系结构   B 哈佛结构  
C 单总线结构      D 双总线结构

13.下列不是 RISC 指令系统特点的是 ( C )

A 大量使用寄存器      B 采用固定长度指令格式  
C 使用多周期指令      D 寻址方式多

14.与通用操作系统相比嵌入式操作系统还必须具有的特点是( A )

A 强稳定性，弱交互性   B 较强实时性   C 可伸缩性   D 功耗管理与节能

15.下列那种设备不属于嵌入式系统产品 ( C )。

A、 PDA   B ATM 取款机   C 个人计算机   D 机顶盒

- 16.下列是开源的嵌入式操作系统的是（ A ）
- A.嵌入式 linux B Vxworks D Palm OS E WinCE
- 17.下列不属于 ARM 处理器异常工作模式的是（ D ）
- A.快速中断模式 B. 未定义模式  
C.数据访问终止模式 D. 用户模式
- 18.嵌入式软件开发与通用软件开发增加了（ C ）环节
- A. 代码编写 B. 代码调试 C. 代码固化 D. 软件测试
- 19.嵌入式系统调试方式中不占用系统资源的调试方式是（C）
- A 模拟器方式 B 监控器方式  
C. ICE 在线仿真器 D IDE 在线调试器方式
- 20.ADD R0,R1,[R2]中的第二操作数属于（ B ）方式。
- A.立即寻址 B 寄存器间接寻址  
C.寄存器寻址 D.基址变址寻址
- 21.烧写到 FLASH 里的是（ B ）格式的文件。
- A. .asm B. .bin C. .obj D..cpp
22. 汇编程序中的伪指令是在（ C ）阶段起作用。
- A. 编辑 B. 链接 C.编译 D. 执行
- 23、目前嵌入式处理器的种类大约有（ C ）
- A. 500 多种 B.800 多种 C. 1000 多种 D.2000 多种
- 24.LDM 和 STM 指令最多可以操作（ C ）寄存器。
- A. 12 个 B.2 个 C. 16 个 D. 32 个.

## 二、填空题

- 1.ARM 内核有 ( T ) ( D ) ( M ) ( I ) 四个功能模块。
- 2.通用寄存器 R0~R15 可分为以下 3 块：(未分组寄存器 R0-R7) (分组寄存器 R8-R14) (程序计数器 R15)
- 3.ARM 可用两种方法存储字数据：(大端模式 ) (小端模式 )
- 4.ARM 体系结构支持的最大寻址空间为 ( 4GB )。
- 5.ARM 是 ( Advanced RISC Machines ) 的简称。
- 6.ARM CPU 体系结构中支持的特有的两种指令集是：(ARM )(Thumb)
- 7.ARM 处理器的 5 种异常模式是 (快速终端模式) (外部中断模式) (管理模式) (数据访问终止模式) (未定义模式)
- 8.BSP 是 ( board support package ) 的缩写。
- 9.JTAG 是 ( Joint Test Action Group ) 的简称。JTAG 技术是一种调试技术，它是在芯片内部封装了专门的测试电路 TAP (Test Access Port, 测试访问口)，通过专用的 JTAG 测试工具对内部节点进行测试调试技术。
- 10..ARM 汇编程序由 (指令) (伪操作 ) (宏指令) 三种指令组成
- 11.Bootloader 依赖于 ( CPU 体系结构 )，也依赖于具体板级设备配置。
- 12.ARM 中堆栈类型有 ( 满递增 ) (满递减) (空递增 ) ( 空递减 ) 四种。
- 13.ARM 处理器具有 ( ARM ) ( Thumb) 两种操作状态
- 14.ARM 处理器内核型号中的字母 TDMI 的含义分别是 ( thumb 支持 16 位指令集 ) (支持片上 debug ) ( 内嵌硬件乘法器 multipiler ) ( 嵌入式 ICE 支持片上断点和调试点)。

15.FLASH 存储技术中两种流行的存储技术是（norflash）（nandflash），其中成本较低，而接口比较复杂的 flash 存储技术是（nandflash）。

16.ARM 处理器有（快速中断模式）（管理模式）（外部中断模式）（数据访问终止模式）（用户模式）（系统模式）（未定义模式）7 种运行模式。其中特权模式 是：（快速中断模式）（管理模式）（外部中断模式）（数据访问终止模式）（系统模式）（未定义模式）

异常模式是：（快速中断模式）（管理模式）（外部中 断模式）（数据访问终止模式）（未定义模式）。

17.ARM 处理器有（37）个寄存器，其中系统模式和用户模式下有（1）物理寄存器可用其他 5 中模式下都有（5）物理寄存器可用

18. ARM 处理其每种模式下都有一组寄存器可用，其中（系统模式）（用户模式）具有完全一样的寄存器组。

19. ARM 中共有（6）个状态寄存器，其中一个（用户模式和系统模式共用）和五个（对应其他 5 种模式），其中 CPSR 包括（条件标志位）（中断标志位）（当前处理模式标志位）（判别、状态位）内容。

20.Linux 系统中的 SHELL 的主要功能（命令解释）（编程语言）（用户接口），是通过编写脚本文件来实现系统管理和使任务自动化的。

21. linux 系统中的（makefile）是 linux 软件开发管理工具，决定程序的编译链接规则，使工程编译链接自动化，它依赖于（make）工作。

22.目前国内对嵌入式系统普遍认同的定义是：以（应用）为中心，以（计算机）为基础，（软硬件）可裁剪，适应应用系统对功

能，可靠性，成本，体积功耗严格要求的专用计算机系统。

23.常见的嵌入式操作系统有（Linux）（Wince）（Palm）（VxWorks），  
Uc/OS-II 和 Ecos.

24.Linux 一般包括四个部分（内核）（shell）（文件系统）（实用工具）。

25.目前几乎所有的发行版本 Linux 都包含（KDE）和（Gnome）两种图形操作环境。

26.Linux 内核主要有（进程管理）（内存管理）（设备驱动）（文件系统驱动）（网络管理）五个子系统组成。

27.Linux 下常见 shell 有（Bourne Shell）（BASH）（Korn Shell）（C Shell）等。

28.Linux 文件系统将文件分为（普通文件），（目录文件），（特殊文件）。

29.嵌入式处理器可以分为（嵌入式微控制器），（嵌入式微处理器器）  
（DSP 处理器）（片上系统）。

30.Vi 编辑器有三种模式，分别是（插入模式）（命令行模式）（底行模式）。  
其中（ 插入 ）模式下可以输入内容。

31.用 gcc 编译文件生成可执行文件要经历四个相互关联的步骤：（预处理）（编译）（汇编）（链接）。

32. 经过编译后生成的文件后缀一般为（.O）。

33.一般 bootloader 有两种操作模式（启动加载模式）和（下载模式），  
其中（下载）模式只对开发人员有意义。

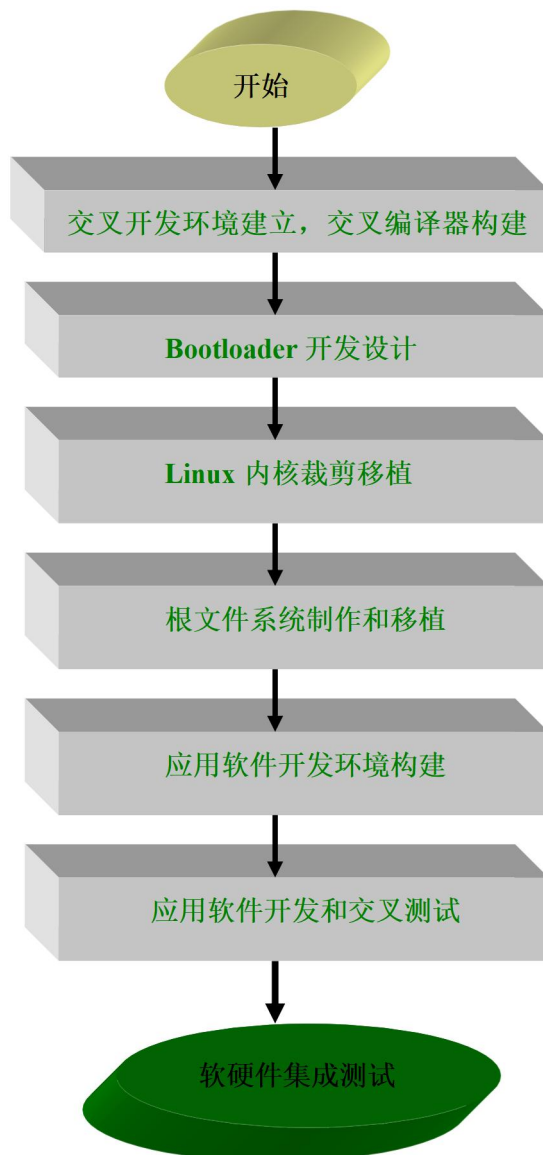
### 三. 简答

1. 嵌入式 Linux 系统组成。

答：自底向上主要包括硬件设备层，bootloader，Linux 内核，各种驱动程序，内核之上的文件系统，各种库以及最上层的应用软件。

## 2. 简述嵌入式 Linux 系统开发流程。

答：嵌入式 Linux 系统开发流程如下图所示：



## 3. 简述嵌入式 Linux 系统根文件系统的作用。

答：根文件系统首先是一种文件系统，该文件系统不仅具有普通文件系统的存储数据文件的功能，但是相对于普通的文件系统，它的特殊之处在于，它是内核启动时所 mount 的第一个文件系统，内核代码的映像



文件保存在根文件系统中，系统引导启动程序会在根文件系统挂载之后从中把一些初始化脚本（如 rcS, inittab）和服务加载到内存中去运行。根文件系统和内核是完全独立的两个部分。在嵌入式 linux 系统中移植的内核下载到开发板上，如果没有根文件系统的支持，是没有办法真正启动 Linux 操作系统的，会出现无法加载文件系统的错误。

4. arm 系列处理器 arm7tdmi 中的 tdm 四个字母的含义。

答：其中 T 功能模块表示 16 位 Thumb，可以在兼顾性能的同时减少代码尺寸。M 功能模块表示 8 位乘法器。D 功能模块表示 Debug，该内核中放置了用于调试的结构，通常它为一个边界扫描链 JTAG，可使 CPU 进入调试模式，从而可方便地进行断点设置、单步调试。I 功能模块表示 EmbeddedICE Logic，用于实现断点观测及变量观测的逻辑电路部分，其中的 TAP 控制器可接入到边界扫描链。

5.简述哈佛体系结构和冯诺依曼体系结构的不同点。

答：冯诺依曼结构是一种程序指令存储器和数据存储器合并在一起的存储器结构。

哈佛体系结构是一种将程序指令存储和数据存储分开的存储器结构。

Intel 处理器采用冯诺依曼结构，ARM 属于哈佛结构处理器。

6..简述 NAND\_FLASH 和 NOR\_FLASH 的异同点？

Nor-flash 的读速度比 Nand-Flash 快 Nor-flash 的写速度比 Nand-Flash 慢 Nor-flash 的擦除速度比 Nand-Flash 慢 大多数写入操作需要先进行

擦除操作

Nand-flash 的擦除单元更小，相应的擦除电路更少

Norflash 待用 SRAM 接口，有足够多的地址引脚，容易进行电路设计 Nand-flash 使用复杂的 I/O 口串行传输数。单位面积内 Nand-flash 存储容量更大 Nand-flash 占据大容量存储市场(8-128M) 可靠性和耐用性

nandFlash 更优秀（可擦写次数 100 万：10 万） 易用性

Nor-flash 不需要软件支持，片上执行 Nandflash 需要驱动程序支持

7.简述 S3C2410 与 ARM920T 的关系.

S3C2410 是韩国三星公司的一款基于 ARM920T 内核的 32 位 RISC 嵌入式微处理器，主要面向手持设备以及高性价比低功耗的应用

8. 嵌入式系统常用调试方法有几种，各有什么优缺点？

### 1. 模拟器方式

调试工具和待调试的嵌入式软件都在主机上运行，通过软件手段模拟执行为某种嵌入式处理器编写的源程序。简单的模拟器可以通过指令解释方式逐条执行源程序，分配虚拟存储空间和外设，进行语法和逻辑上的调试。

### 2. 在线仿真器方式

在线仿真器 ICE 是一种完全仿造调试目标 CPU 设计的仪器，目标系统对用户来说是完全透明的、可控的。由于仿真器自成体系，调试时可以连接目标板，也可以不接目标板。但是 ICE 价格昂贵，而且每种 CPU 都需要一种与之对应的 ICE，使得开发成本非常高。

### 3.监控器方式

主机和目标板通过某种接口（通常是串口）连接，主机上提供调试界面，被调试程序下载到目标板上运行，通过与监控运行于目标机上的监控程序通信，获得调试信息。

9.常见的嵌入式操作系统有哪些？（列举五个以上），嵌入式操作系统有什么特点？

Linux、uClinux、WinCE、PalmOS、Symbian、eCos、uCOS-II、VxWorks、pSOS、Nucleus、ThreadX、Rtems、QNX、INTEGRITY、OSE、C Executive

1) 可裁剪性。支持开放性和可伸缩性的体系结构。 2) 强实时性。EOS 实时性一般较强, 可用于各种设备控制中。 3) 统一的接口。提供设备统一的驱动接口。 4) 操作方便、简单、提供友好的图形 GUI 和图形界面, 追求易学易用。提供强大的网络功能, 支持 TCP/IP 协议及其他协议, 提供 TCP/UDP/IP/PPP 协议支持及统一的 MAC 访问层接口, 为各种移动计算设备预留接口。 5) 强稳定性, 弱交互性。嵌入式系统一旦开始运行就不需要用户过多的干预、这就要负责系统管理的 EOS 具有较强的稳定性。嵌入式操作系统的用户接口一般不提供操作命令, 它通过系统的调用命令向用户程序提供服务。 6) 固化代码。在嵌入式系统中, 嵌入式操作系统和应用软件被固化在嵌入式系统计算机的 ROM 中。 7) 更好的硬件适应性, 也就是良好的移植性。

10.什么是交叉开发环境？简述嵌入式 linux 系统交叉开发环境的构建过程？

答：嵌入式系统通常是一个资源受限的系统，无法直接在嵌入式系统的硬件平台上进行相关的软件开发工作，因此必须采用一种特殊的开发模式，目前一般采用的方法就是首先在通用计算机上编写程序，然后通过交叉编译生成目标平台上可以运行的二进制代码格式，最后再通过一些特殊的下载烧写工具，将二进制程序下载烧写到目标平台上去运行，将这种开发方法叫做交叉开发

嵌入式 Linux 系统交叉开发环境的建立是在系统硬件模板开发完成后，最先要完成的任务，主要任务是在宿主机上建立为目标系统开发系

统软件和应用软件的开发环境，主要任务有两个，首先，在宿主机上安装 Linux 操作系统发行版，宿主机上的发行版 Linux 操作系统的主要任务是运行交叉编译器，为目标系统开发各类软件。其次，是在安装了 Linux 的 PC 机上，建立目标系统的交叉编译器，交叉编译器的功能是为目标系统开发各类软件，包括系统软件。宿主机 Linux 操作系统的选型和安装方式以方便系统开发为唯一选择标准，可以选择开发者熟悉的 Linux 操作系统发行版本，最好与嵌入式目标系统 Linux 操作系统内核版本保持一致。比如目标系统拟采用 2.6 内核的 Linux 操作系统，那么宿主机上 Linux 发行版最好选择 2.6 内核的发行版本或者更高级的版本。

11. 简述 bootloader 在嵌入式系统平台上的移植过程？

1).准备 Bootloader 软件

2).根据目标板硬件平台，修改配置 Bootloader。

3).交叉编译，形成目标板可执行的 Bootloader 二进制 BIN 文件。4).将 Bootloader 烧写进目标板。

5).配置好宿主机上的 minicom 或者超级终端。

6).启动目标板，若 bootloader 编译没有错误，烧写正确，就可以在超级终端中看到目标板启动的信息。

12. 说明 linux 平台下用 C 系统开发使用的工具。

答：Linux 下 c 语言开发主要采用 GNU 开发工具，主要使用编辑器，编译器，和调试器，大型项目开发使用 makefile 来进行项目开发管理，常用的编辑器有 VI,gedit 等，编译器一般用 gcc，调试器使用 gdb 调试器。

13.嵌入式系统有什么特点？

- 1) 嵌入式系统是面向产品，面向用户，面向应用的。
- 2) 是一个技术密集，集成度高，需要不断创新的系统。
- 3) 嵌入式系统软硬件必须是可裁剪的。

#### 14.嵌入式系统组成？

答：嵌入式系统一般由嵌入式计算机和执行部件组成。其中嵌入式计算机是整个嵌入式系统的核心，主要包括硬件层、中间层、系统软件层以及应用软件层；执行部件则是接收嵌入式计算机系统发出的控制指令，执行规定的操

#### 15. 嵌入式系统的发展经过那几个阶段？

答：嵌入式系统发展主要经历了如下四个阶段：

- 1) 无操作系统的阶段
- 2) 以嵌入式 CPU 为基础，简单操作系统为核心的嵌入式系统阶段
- 3) 以通用操作系统为核心的嵌入式实时系统阶段。
- 4) 基于 Internet 为标志的嵌入式系统阶段。

#### 16.如何构建交叉编译工具链？

答：构建交叉编译工具链有三种方法：

1) 分步编译和安装交叉编译工具链，从网上下载相关的源代码包，自己配置编译，生成交叉编译工具链。该方法相对比较困难，适合想深入学习构建交叉工具链的读者。.

2) 通过 **Crosstool** 脚本工具.**Crosstool** 脚本由专业的公司制作的，生成交叉编译工具链的脚本程序，只需要进行简单的配置修改，就可以进行一次编译生成交叉编译工具链，该方法相对于方法一要简单许多，

并且出错的机会也非常少，大多数情况下使用该方法构建交叉编译工具链。

3)直接通过网上（<ftp.arm.kernel.org.uk>）下载已制作好的交叉编译工具链。

17.什么是 bootloader?

答：Bootloader，亦称引导加载程序，是在操作系统内核运行之前运行的一段小程序，程序的主要功能是初始化系统基本硬件设备、建立内存空间的映射图，从而将系统的软硬件环境带到一个合适的状态，以便为最终调用操作系统内核准备好正确的环境。

18.举例说明身边的嵌入式系统。（说出 15 个以上）

19.Linux 内核源码是如何组织的？主要目录下有哪些内容？

答：Linux 内核源码组织成树形结构，主要包括如下一些目录。

**/arch**：目录包括了所有和体系结构相关的核心代码。。

**/documentation**：目录下是一些文档。

**/drivers**：目录中是系统中所有的设备驱动程序。

**/fs**:所有的文件系统代码和各种类型的文件操作代码，它的每一个子目录支持一个文件系统，例如 fat 和 ext2。

**/include**：目录包括编译核心所需要的大部分**头文件**。

**/init**：目录包含核心的初始化代码（不是系统的引导代码）。

**/ipc**：目录包含了核心进程间的通信代码。

**/Kernel**：内核管理的核心代码。

**/lib:** 目录包含了核心的库代码。

**/mm:** 目录包含了所有独立于 **cpu** 体系结构的内存管理代码。

**/net:** 目录里是核心的网络部分代码。

**/scripts:** 目录包含用于配置核心的脚本文件等

**/block:** 块设备驱动程序 I/O 调度。

**/crypto:** 常用加密和散列算法（如 AES,SHA 等），还有一些压缩和 CRC 校验算法。

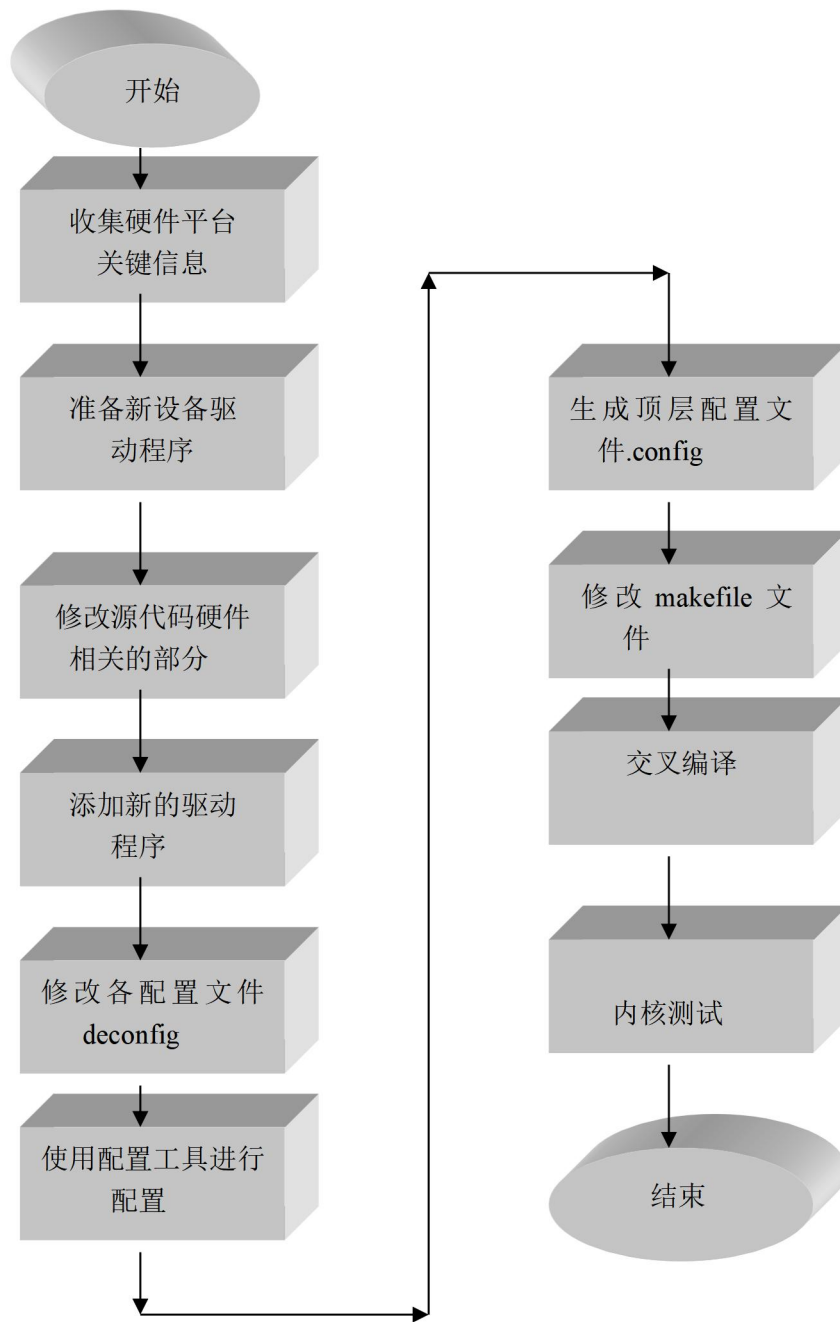
**/security:** 主要包含 SELinux 模块。

**/sound:** ALSA,OSS 音频设备的驱动核心代码和常用设备驱动。

**/usr:** 实现了用于打包和压缩的 **cpio** 等。

20.简述 Linux 内核移植步骤。

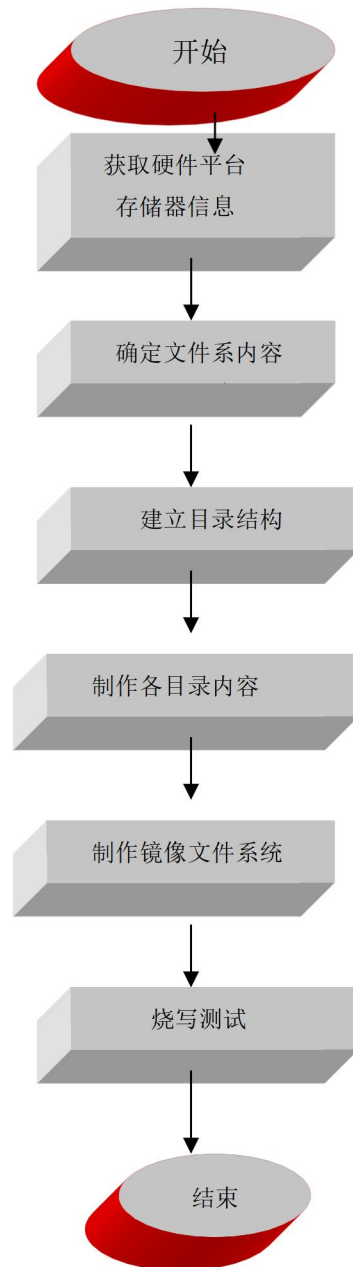
答：Linux 内核移植步骤如下图所示：



21.简述嵌入式 Linux 根文件系统制作过程。

答：嵌入式 Linux 根文件系统制作过程如下图所示：





## 22.Linux 系统软件开发中 makefile 作用是什么？

答：Makefile 一种文件，makefile 定义了一系列的规则来指定，哪些文件需要先编译，哪些文件需要后编译，哪些文件需要重新编译，甚至于进行更复杂的功能操作 makefile 就像一个 Shell 脚本一样，其中也可以执行操作系统的命令。makefile 带来的好处就是——“自动化编译”，一旦写好，只需要一个 make 命令，整个工程完全自动编译，极大的提高了软件开发的效率。make 是一个命令工具，是一个解释 makefile 中指令的命令工具。

### 三.程序设计题

#### 1.阅读程序，写出程序运行结果。

(1)

```
#!/bin/sh
```

```
i=10;
```

```
while [[ $i -gt 5 ]];do
```

```
    echo $i;
```

```
    ((i--));
```

```
done
```

(2) #!/bin/bash

```
for ((i=1;i<=10;i++));do
```

```
    echo $i
```

```
done
```

(3)

```
#!/bin/bash
```

```
if [-d "$1"]; then
```

```
rm -rf $1
```

```
elif [ -f "$1" ]; then
```

```
rm $1 -rf
```

```
fi
```

1、以下说法不正确的是（ b）。

- A、任务可以有类型说明
- B、任务可以返回一个数值
- C、任务可以有形参变量
- D、任务是一个无限循环

2、用图形点阵的方式显示一个 16\*16 点阵汉字需要（ B ）字节。

- A、8
- B、32
- C、16
- D、64

3、μCOS-II 操作系统中最多可以管理（ A ）个任务。

- A、64
- B、56
- C、128
- D、63

4、存储一个 32 位数 0x2168465 到 2000H~2003H 四个字节单元中，若以大端模式存储，则 2000H 存储单元的内容为（ ）。d

- A、0x21
- B、0x68
- C、0x65
- D、0x02

5、以 ARM 为内核的微处理器是（ D ）位的。d

- A、4 位
- B、8 位
- C、16 位
- D、32 位

6、RS232-C 串口通信中，表示逻辑 1 的电平是（ D）。d

- A、0v
- B、3.3v
- C、+5v~+15v
- D、-5v~-15v

7、寄存器 R14 除了可以做通用寄存器外，还可以做（ ）。b

- A、程序计数器
- B、链接寄存器
- C、栈指针寄存器
- D、基址寄存器

- 8、IRQ 中断的入口地址是 (C )。
- A、0x00000000                      B、0x00000008
- C、0x00000018                      D、0x00000014
- 9、采用 RS232-C 串行通信至少需要三根线，其中不包括 ( A)。
- A、电源线                              B、地线
- C、发送数据线                      D、接收数据线
- 10、Unicode 编码与 (A ) 编码方式兼容。
- A、ASCII 码                              B、GBK
- C、GB2312                              D、区位码

## 二、填空题（本大题共 24 个空，每空 1 分，共 24 分）

得分	评阅人

- 11、发展到现在，常用的操作系统包括三大类，分别为：批处理系统、分时系统、和实时系统。
- 15、OSTaskSuspend (self)可把正在运行的任务挂起，参数 self 指任务本身。
- 用此函数挂起的任务可通过OSTaskResume函数唤醒。
- 16、创建信号量 OSSemCreat (0)函数中，参数 0 表示信号量的初始值。
- 17、若一个优先级为 35 的任务就绪，则在就绪表中，OSRdyTbl[]的2置位，变量 OSRdyGrp 的4置位。
- 18、网络通信中面向连接的协议是tcp，面向无连接的协议是udp。

## 三、简答题（本大题共 4 个小题，每小题 5 分，共 20 分）

得分	评阅人

- 19、简述进程和线程有什么区别， $\mu$  COS-II 操作系统中的任务属于进程还是线程？

进程：包含正在运行的一个程序的所有状态信息，包括程序的代码、程序的数据、PC 值、一组通用的寄存器的当前值，堆、栈，一组系统资源（如打开的文件）。进程有动态性、独立性、并发性的特点。

线程：是较进程更小的能独立运行的基本单位，是进程当中的一条执行流程，它可以满足统一进程内不通实体间的并发执行而又共享相同的地址空间。

一个进程中可以同时存在多个线程

各个线程之间可以并发地执行

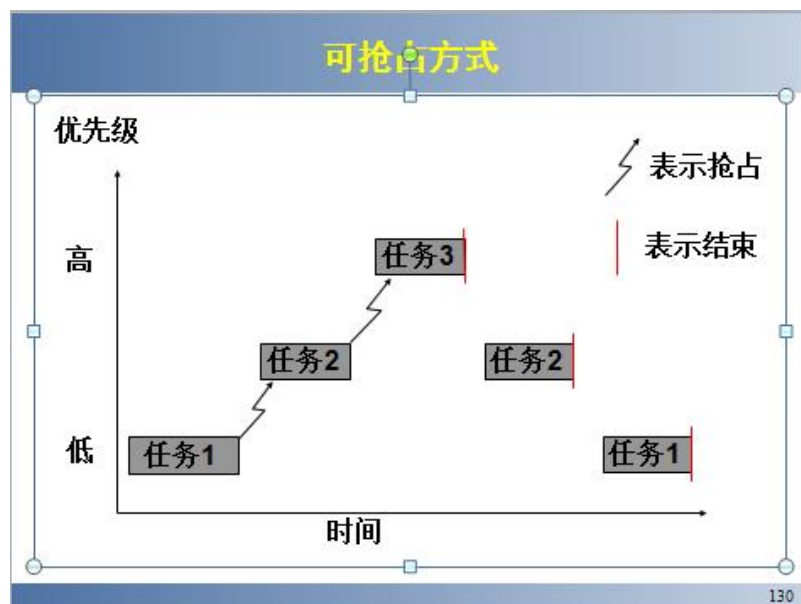
各个线程之间可以共享地址空间

μ COS-II 操作系统中的任务属于线程

20、简述占先式和非占先式操作系统内核特点？并画图说明他们之间的主要区别。

不可抢占调度方式：一个进程若被选中就一直运行下去，直到它被阻塞（I/O，或正在等待其他进程），或主动地交出 CPU。

可抢占调度方式：当一个进程在运行时，调度程序可以打断它。



21、画图说明 μ COS-II 操作系统各状态之间是如何切换的，并标明所有可能的切换函数。

见书上 118 页的图和下面的说明

22、什么是 μ COS-II 操作系统的时钟节拍？

时钟节拍是一种特殊的中断，相当于操作系统的心脏起搏器  
μC/OS 需要用户提供周期性信号源，用于实现时间延时和确认超时。节拍率应在 10 到 100Hz 之间，时钟节拍率越高，系统的额外负荷就越重；  
时钟节拍的频率取决于用户应用程序的精度。时钟节拍源可以是专门的硬件定时器，或是来自 50/60Hz 交流电源的信号。

23、有两个任务代码如下，其中 Task\_A、Task\_B 的优先级分别为：28、37。

分析 LCD 上的显示结果。

```
Void Task_A ()
{
    ClearScreen();
    LCD_Printf( "task1 is running!\n" );
    OSTimeDly(400);
}

Void Task_B ()
{
    ClearScreen();
    LCD_Printf( "hello task2! \n" );
    OSTimeDly(160);
}
```

在 LCD 上的显示结果为：

第一次：\_\_\_\_\_ task1 is running!\_\_\_\_\_

第二次：\_\_\_\_\_ hello task2!\_\_\_\_\_

第三次：\_\_\_\_\_ hello task2!\_\_\_\_\_

第四次：\_\_\_\_\_ task1 is running!\_\_\_\_\_

第五次：\_\_\_\_\_ hello task2! \_\_\_\_\_

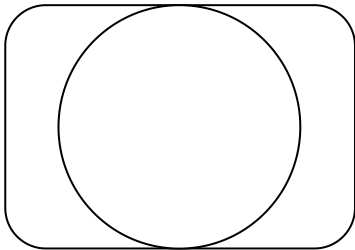
第六次：\_\_\_\_\_ hello task2! \_\_\_\_\_

24、在分辨率为 320x240 的 LCD 上，执行完下列程序后，画出在 LCD 上显示的图形并在划横线处添加注释。

```
void Main_Task(void *Id)
{
    int oldx, oldy;
    PDC pdc;
```

```
ClearScreen();
pdc=CreateDC();
SetDrawOrg(pdc, LCDWIDTH/2,LCDHEIGHT/2, &oldx, & oldy);
Circle(pdc,0, 0, 50); //以（0, 0）为圆心，50 为半径画圆
MoveTo(pdc, -50, -50); //将画笔起点移动到（-50,50）
LineTo(pdc, 50, -50); // 画一条起点为（-50,50）终点为（50,
-50）的直线
ArcTo(pdc, 80, -20, TRUE, 30); //画一条起点为（50,-50）终点
为（80, -20）的曲率为 30 的曲线

LineTo(pdc, 80, 20);
ArcTo(pdc, 50, 50, TRUE, 30);
LineTo(pdc, -50, 50);
ArcTo(pdc, -80, 20, TRUE, 30);
LineTo(pdc, -80, -20);
ArcTo(pdc, -50, -50, TRUE, 30);
OSTimeDly(3000);
}
```



320x240 的 LCD

五、程序分析题（本大题共 3 个小题，每小题 6 分，共 18 分）

得分	评阅人

25、下面程序是多任务中信号量的典型应用，阅读程序，分别画出各个任务的程序流程图，并说明程序要实现的功能什么。

```
OS_EVENT *math_Sem;

int source, dest;

math_Sem=OSSemCreate(0);

void task1()
{
    source = 8;
    OSSemPost(math_Sem);
}

void task2()
{
    U8 err;
    OSSemPend(math_Sem, 0, &err);
    dest = source;
    source =0;
}
```

Task1:

给 source 赋值

对于 math\_Sem 调用 OSSemPost 函数:

- 👉 检查是否有任务在等待该信号量，如果没有，将信号量的计数值加 1 并返回；
- 👉 如果有，将优先级最高的任务从等待任务列表中删除，并使它进入就绪状态；
- 👉 调用 OSSched()，判断是否需要进行任务切换。

Task2:

申请信号量资源，调用 OSSemPend 函数:

- 👉 如果信号量的计数值大于 0，将它减 1 并返回；
- 👉 如果信号量的值等于 0，则调用本函数的任务将被阻塞起来，等待另一个任务把它唤醒；



👉 调用 OSSched() 函数，调度下一个最高优先级的任务运行。

将 source 的值传给 dest，source 清零

该程序是一个操作系统中 PV 操作的演示程序，其中 Task1 为 V 操作，添加信号量的值，Task2 为 P 操作，等待一个信号量，将信号量减 1。程序的运行结果为 dest=8，source 值为 0。

26、下面程序是消息循环在多任务中的应用实例，仔细阅读程序，画出程序流程图，并说明程序要实现的功能是什么。（假设回车键的键值为 14）

```
void task1()
{
    POSMSG pMsg=0;
    for(;;)
    {
        pMsg=WaitMessage(0);
        switch(pMsg->Message) {
            case OSM_KEY:
                onKey(pMsg->WParam, pMsg->LParam);
                break;
        }
        DeleteMessage(pMsg);
    }
}

onKey(int nkey, int fnkey)
{
    switch(nkey) {
        case 14://OK
            LCD_printf("my name is XXX !\n");
            break;
        case 7://Cancel
            LCD_Cls();
    }
}
```



```

SetTextCtrlEdit(pText1, TRUE);
for(;;) {
    pMsg=WaitMessage(0);
    switch(pMsg->Message) {
        case OSM_KEY:
            onKey(pMsg->WParam, pMsg->LParam);
            break;
    }
    DeleteMessage(pMsg);
}

onKey(int nkey, int fnkey)
{
    switch(nkey) {
        case 1:
            AppendChar2TextCtrl(pTextCtrl, 0x31, TRUE);
            break;
        case 2:
            AppendChar2TextCtrl(pTextCtrl, 0x32, TRUE);
            break;
        case 10:
            TextCtrlDeleteChar(pTextCtrl, TRUE);
            break;
    }
}

```

与上题类似，只不过定义了一个文本控制框来输出文本。

1. 下面哪一种工作模式不属于 ARM 特权模式 ( A )。

A. 用户模式

B. 系统模式

C. 软中断模式

D. FIQ

模式

2. ARM7TDMI 的工作状态包括 ( D )。

- A. 测试状态和运行状态  
B. 挂起状态和就绪状态  
C. 就绪状态和运行状态  
D. ARM 状态和 Thumb

状态

3. 下面哪个 Linux 操作系统是嵌入式操作系统 ( B )。

- A. Red-hat Linux      B. uclinux      C. Ubuntu Linux      D. SUSE

Linux

4. 使用 Host-Target 联合开发嵌入式应用, ( B ) 不是必须的。

- A. 宿主机      B. 银河麒麟操作系统  
C. 目标机      D. 交叉编译器

5. 下面哪个不属于 Linux 下的一个进程在内存里的三部分的数据之一 ( A )。

- A. 寄存器段      B. 代码段  
C. 堆栈段      D. 数据段

选择题 (共 5 小题, 每题 2 分, 共 10 分)

1. 下面哪个系统属于嵌入式系统 ( D )。

- A. “天河一号” 计算机系统      B. IBMX200 笔记本电脑  
C. 联想 S10 上网本      D. Iphone 手机

2. 在 Makefile 中的命令必须要以 ( A ) 键开始。

- A. Tab 键      B. #号键  
C. 空格键      D. &键

3. Linux 支持多种文件系统, 下面哪种不属于 Linux 的文件系统格式 ( B )。

- A. Ext      B. FAT32  
C. NFS      D. Ext3

4. 下面哪种不属于 VI 三种工作模式之一 ( D )。

- A. 命令行模式      B. 插入模式  
C. 底行模式      D. 工作模式

5. 下面哪一项不属于 Linux 内核的配置系统的三个组成部分之一 ( C )。

- A. Makefile      B. 配置文件 (config.in)  
C. make menuconfig      D. 配置工具

1. 人们生活中常用的嵌入式设备有哪些? 列举 4 个以上(1) 手机, (2) 机顶

盒\_\_\_\_\_，(3) MP3\_\_\_\_\_，(4) GPS\_\_\_\_\_。(交换机、打印机、投影仪、无线路由器、车载媒体、PDA、GPS、智能家电等等。)

2. ARM9 处理器使用了五级流水线，五级流水具体指哪五级：(5) 取指\_\_\_\_\_，(6) 译码\_\_\_\_\_，(7) 执行\_\_\_\_\_，(8) 缓冲/数据\_\_\_\_\_，(9) 回写\_\_\_\_\_。

3. 在 Makefile 中的命令必须要以(10) Tab\_\_\_\_\_键开始。

4. Linux 支持多种文件系统，主要包括哪些（写出其中 4 中就行）(11) Ext\_\_\_\_\_，(12) VFAT\_\_\_\_\_，(13) JFS\_\_\_\_\_，(14) NFS\_\_\_\_\_。(JFS、ReiserFS、Ext、Ext2、Ext3、ISO9660、XFS、Minx、MSDOS、UMSDOS、VFAT、NTFS、HPFS、NFS、SMB、SysV、PROC 等)

5. VI 的工作模式有哪三种：(15) 命令行模式\_\_\_\_\_，(16) 插入模式\_\_\_\_\_，(17) 底行模式\_\_\_\_\_。

6. Linux 下的一个进程在内存里包括三部分的数据，它们是：(18) 代码段\_\_\_\_\_，(19) 堆栈段\_\_\_\_\_，(20) 数据段\_\_\_\_\_。

1. 目前使用的嵌入式操作系统主要有哪几种 (1) Linux\_\_\_\_\_，(2) Windows CE\_\_\_\_\_，(3) uc/OS II\_\_\_\_\_，(4) Android\_\_\_\_\_。(或 VxWorks, Palm OS , QNX, Symbian 等)

2. 如果按用途划分，嵌入式微处理器可分为哪类 (5) 嵌入式微控制器(MCU)，又称为单片机\_\_\_\_\_，(6) 嵌入式微处理器(EMPU)\_\_\_\_\_，(7) 嵌入式 DSP 处理器\_\_\_\_\_，(8) 嵌入式片上系统(SOC)\_\_\_\_\_。

3. ARM 核有两个指令集，分别是：(9) ARM 指令集\_\_\_\_\_和 (10) Thumb 指令集\_\_\_\_\_。

4. Makefile 里主要包括的语法规则有哪些：(11) 显式规则\_\_\_\_\_，(12) 隐晦规则\_\_\_\_\_，(13) 变量定义\_\_\_\_\_，(14) 文件指示\_\_\_\_\_，(15) 注释\_\_\_\_\_。

5. 大多数 Bootloader 都包含两种不同的操作模式，分别是：(16) 启动加载模式\_\_\_\_\_，(17) 下载模式\_\_\_\_\_。

6. Linux 内核的配置系统由三部分组成，它们分别是：(18) Makefile\_\_\_\_\_，(19) 配置文件 (config.in)\_\_\_\_\_，(2) 配置工具\_\_\_\_\_。

RISC: reduced instruction set computer, 精简指令集计算机	MCU: Micro Controller Unit, 微控制单元
DSP: Digital Signal Processor, 【电脑】数字信号处理器 【电脑】只读存储器	ROM: read-only memory,
SWI: Software Interrupt	PC: 程序计数器
IRQ: 外部中断请求	NFS: Network File System 网络文件系统
SOC: System On Chip, 片上系统	CPSR : 当前程序状态寄存器
ICE: In Circuit Emulator, 实时在线仿真器	MMU: Memory Management Unit, 内存管理单元
FIQ: 快速中断请求	VFS: 虚拟文件系统
IPC : Inter-Process Communication 内部进程间通信	IR: 指令寄存器

1、 嵌入式开发环境主要包括哪些组件？

答：嵌入式系统开发需要交叉编译和在线调试的开发环境，主要包括

- 1 宿主机
- 1 目标机（评估电路板）
- 1 基于 JTAG 的 ICD 仿真器、或调试监控软件、或在线仿真器 ICE
- 1 运行于宿主机的交叉编译器和链接器、以及开发工具链或软件开发环境
- 1 嵌入式操作系统

2、 什么是嵌入式系统，嵌入式系统同 PC 系统相比具有哪些特点？

根据电气工程师协会的定义，嵌入式系统（Embedded System）是用来控制或者监视机器、装置、工厂等大规模系统的设备。一般认为嵌入式系统是以应用为中心，以计算机技术为基础，其软/硬件可裁减，可满足应用系统对功能、可靠性、成本、体积、功耗的严格要求的专用计算机系统。

(1) 嵌入式系统功耗低、体积小、专用性强。嵌入式系统与 PC 系统的最大不同就是嵌入式 CPU 大多工作在为特定用户群设计的系统中，能够把 PC 系统中许多由板卡完成的任务集成在芯片内部，从而使系统设计趋于小型化。

(2) 嵌入式系统中的软件一般都固化在存储器芯片或单片机芯片中，以提高执行速度和系统可靠性。

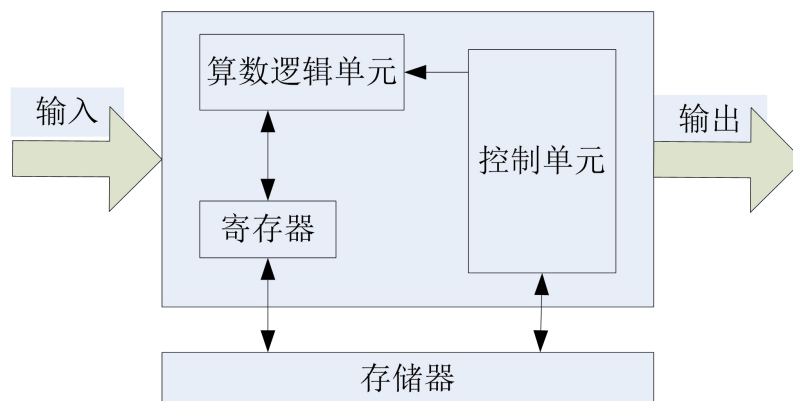
(3) 嵌入式系统的硬件和软件都经过精心设计，系统精简，其操作系统一般和应用软件集成在一起。

(4) 软件代码质量要求高。

(5) 嵌入式系统开发需要专门的开发工具和开发环境。

3、 嵌入式硬件系统一般由哪几部分组成？其核心是什么？核心由哪几部分组成？他们之间的关系是怎样的（请用图表示出来）？

嵌入式硬件系统一般由嵌入式微处理器、存储器和输入/输出部分组成。其中嵌入式微处理器是嵌入式硬件系统的核心，通常由三大部分组成：控制单元、算术逻辑单元和寄存器。



4. Boot Loader 在嵌入式系统中主要起什么作用？完成哪些主要的工作？

Boot Loader 是在操作系统运行之前执行的一段程序, 通过这段程序, 初始化硬件设备, 建立内存空间的映射表, 从而建立适当的系统软硬件环境, 为最终调用操作系统内核做好准备。

Boot Loader 的 stage1 通常包括以下工作

1. 硬件设备初始化
2. 加载 Bootloader 的 stage2 准备 ARM 空间
3. 拷贝 Bootloader 的 stage2 到 RAM 空间中
- 4 设置堆栈
- 5 跳转到 stage2 的 C 入口点

Boot Loader 的 stage2 通常包括以下工作

- 1 初始化本阶段要使用到的硬件设备
- 2 检测系统内存映射
- 3 将内核映像和根文件系统映像从 flash 设备上复制到 RAM 空间中
- 4 设置内核启动参数
- 5 调用启动内核

2、在进行嵌入式开发中，编写了一个多文件程序，其中包括 main.c, lan.c, lan.h, timer.c, timer.h, pic.c, pic.h, 主程序在 main.c 中；生成目标文件确定为 fzxy\_jk\_ks, 要求尽可能多的使用隐含规则，需要包括为目标命令，能够使用 make clean 进行中间文件的清理。

```
MAIN = jk_qz_test
```

```
OBJS = main.o lan.o timer.o pic.o
```

```
CC = gcc
```

```
INCDIRS = -I.
```

```
CFLAGS = -Wall -O2 -g $(INCDIRS)
```

```
LIBSLINK = -lpthread
```

```
DEBUG_FLAG =
.SUFFIXES= .o .c
$(MAIN): ${OBJS}
${CC} -o $@ $(CFLAGS) ${DEBUG_FLAG} ${OBJS} $(LIBSLINK)
.o:
$(CC) -g $(CFLAGS) -c $<
clean:
/bin/rm -f core *.o
```

---

- 2、通过创建两个线程来实现对一个数的递加。要求通过 `int main()` 函数分别调用 `void *thread1()`, `void *thread2()`, `void thread_create(void)`, `void thread_wait(void)` 实现，得到的运行结果如下：

```
#include <stdio.h>

#include <string.h>

#include <pthread.h>
#include <sys/time.h>

#define MAX 10
pthread_t thread[2];
pthread_mutex_t mut;
int number=0, i;

void *thread1()
{
    printf("thread1 : I'm thread 1\n");
    for (i = 0; i < MAX; i++){
        printf("thread1 : number = %d\n",number);
        pthread_mutex_lock(&mut);
        number++;
        pthread_mutex_unlock(&mut);
        sleep(2);
    }
    printf("thread1 : 主函数在等我完成任务吗? \n");
    pthread_exit(NULL);
}

void *thread2()
{
    printf("thread2 : I'm thread 2\n");
    for (i = 0; i < MAX; i++){
```



```

        printf("thread2 : number = %d\n",number);
        pthread_mutex_lock(&mut);
        number++;
        pthread_mutex_unlock(&mut);
        sleep(3);
    }
    printf("thread2 :主函数在等我完成任务吗? \n");
    pthread_exit(NULL);
}

void thread_create(void)
{
    int temp;
    memset(&thread, 0, sizeof(thread)); //comment1
    /*创建线程*/
    if((temp = pthread_create(&thread[0], NULL, thread1, NULL)) != 0) //comment2
        printf("线程 1 创建失败!\n");
    else printf("线程 1 被创建\n");
    if((temp = pthread_create(&thread[1], NULL, thread2, NULL)) != 0) //comment3
        printf("线程 2 创建失败");
    else printf("线程 2 被创建\n");
}

void thread_wait(void)
{ /*等待线程结束*/
    if(thread[0] !=0) { //comment4
        pthread_join(thread[0],NULL);
        printf("线程 1 已经结束\n");
    } if(thread[1] !=0) { //comment5
        pthread_join(thread[1],NULL);
        printf("线程 2 已经结束\n");
    }
}

int main()
{ /*用默认属性初始化互斥锁*/
    pthread_mutex_init(&mut,NULL);
    printf("我是主函数哦，我正在创建线程，呵呵\n");
    thread_create();
    printf("我是主函数哦，我正在等待线程完成任务阿，呵呵\n");
    thread_wait();
    return 0;
}

```