



厦门大学《嵌入式系统》课程期末试卷

信息学院 软件工程系 2018 级 软件工程专业

主考教师: 曾文华 试卷类型: (A 卷) 考试时间: 2020.12.29

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. 嵌入式系统的前身通常称为 (1)。
2. ARM 处理器的特权模式是指除 (2) 外的其他六种模式。
3. ARM 指令有两种状态, 分别是: (3) 和 (4)。
4. RT-Linux 中的 RT 是指 (5)。
5. μ CLinux 是专门针对没有 (6) 的处理器设计的。
6. make 编译工具读取的文件名称为 (7) 文件。
7. Bootloader 是 (8)。
8. 使用 mmap 系统调用, 可以将 (9) 空间的地址映射到 (10) 空间。
9. Boot Loader 的阶段 1 主要包含依赖于 CPU 的体系结构硬件初始化的代码, 通常都用 (11) 语言来实现; Boot Loader 的阶段 2, 通常用 (12) 语言完成, 以便实现更复杂的功能, 也使程序有更好的可读性和可移植性。
10. U-boot 2014 的目标结构中的 arch 子目录, 存放的是与 (13) 相关的代码。
11. Flash Memory (闪存) 有两种技术, 分别是 (14) Flash 和 (15) Flash。
12. IMX6 实验箱 Linux 系统挂载的第一个文件系统是 (16)。
13. 设备的控制操作是通过调用 file_operations 结构体中的 (17) 函数完成的。
14. 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 (18) 完成的。
15. 网络设备驱动程序在/dev 目录下没有对应的设备文件, 对网络设备的访问必须使用 (19), 而非读写设备文件。
16. IMX6 实验箱打开电源 (或按 Reset 键) 后, 通常需要重新设置 IP 地址, 并执行挂载命令“mount -t nfs 59.77.5.122:/imx6 /mnt”。该挂载命令中的 nfs 是指 (20), 59.77.5.122 是指 (21) 的 IP 地址。
17. 假设某个 make 命令的执行结果为“gcc -O2 -pipe -g -feliminate-unused-debug-types -c -o hello.o hello.c”, 该结果里“-c”中的 c 是 (22) 的意思, “-o”中的 o 是 (23) 的意思。
18. 如果需要将实验箱的 IP 地址设置为 192.168.1.34, 其命令是: (24)。
19. 假设虚拟机的/tftpboot 目录下有文件 inittab, 可以在 IMX6 实验箱的超级终端上执行有关命令, 通过 TFTP 方式, 将 inittab 文件下载到实验箱中, 这条命令为 (假设虚拟机的 IP 地址为 192.168.1.56) (25)。
20. 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): (26)。
21. Atlas 200 DK 是华为公司生产的 AI 开发套件, 其核心是海思公司生产的 (27) 处理器。
22. ModelArts 是面向 (28) 开发者的一站式开发平台。
23. NDK 集成了交叉编译器, 可以将用 C、C++书写的代码, 编译为 (29) 格式的文件, 在 Android 程序中可以用 Java 语言调用这些代码。
24. Android 的 NDK 开发工具集, 是 Android 为了方便 Android 程序开发者, 通过 (30) 的机制, 达到 Java 和本地 C/C++代码相互沟通的强有力武器。

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）

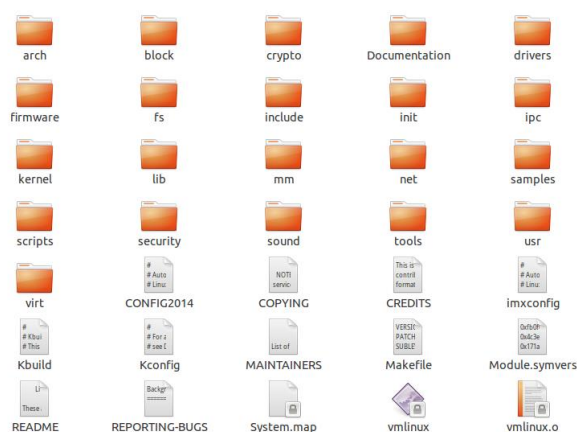
1. CAN
2. CPSR
3. DVPP
4. GPIO
5. JNI
6. JFFS
7. OBS
8. Ramfs
9. TBE
10. TFTP

三、简答题（11 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. 常见的嵌入式操作系统有哪些？（3 分）
2. ARM Cortex-A、ARM Cortex-R、ARM Cortex-M 系列处理器分为针对什么应用场合？（3 分）
3. IMX6 嵌入式教学科研平台（实验箱）的主 CPU 和从 CPU 分别采用哪个 ARM 处理器的产品？（2 分）
4. 请简述设备驱动程序与应用程序的区别。（2 分）
5. Linux 设备驱动程序开发调试有哪两种方法？（2 分）
6. IMX6 嵌入式教学科研平台（实验箱）开机后（打开电源或者按 Reset 键后），将完成哪些任务？（2 分）
7. IMX6 嵌入式教学科研平台（实验箱）程序的执行方式有二种，一种是下载方式，另一种是挂载方式。请举例说明这两种方式的区别。（2 分）
8. 宿主机（PC 机）与目标板（IMX6 实验箱）的连接方式有哪些？（2 分）
9. 什么是 Android NDK？简述 Android NDK 程序的开发过程，包括开发环境的搭建、Hello Jni 程序的编译和运行过程。（4 分）
10. IMX6 嵌入式教学科研平台（实验箱）从 CPU（第二个 CPU）的软件开发工具是什么？请简述该软件开发工具的安装过程。从 CPU 的 LED 灯程序是怎么执行的？（4 分）
11. “云+端协同实验 —— 猫狗图像识别”，云部分完成什么任务？端部分完成什么任务？（4 分）

四、综合题（9 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. ARM-Linux 内核的代码目录结构如下（位于/home/uptech/fsl-6dl-source/kernel-3.14.28/目录下）：



请问，该目录结构中的 arch、block、drivers、fs、mm、net 子目录中分别存放什么内容？（3 分）

2. Android Hello World 程序的工程文件夹如下：

DATA1 (D:) > UP-Tech > Android > Android-application > HelloWorld				
名称	修改日期	类型	大小	
.gradle	2019/11/22 9:25	文件夹		
.idea	2019/11/26 8:50	文件夹		
app	2019/11/23 9:33	文件夹		
build	2019/11/22 9:28	文件夹		
gradle	2019/11/22 9:24	文件夹		
.gitignore	2019/11/22 9:24	GITIGNORE 文件	1 KB	
build.gradle	2019/11/22 9:24	GRADLE 文件	1 KB	
gradle.properties	2019/11/22 9:24	PROPERTIES 文件	1 KB	
gradlew	2019/11/22 9:24	文件	5 KB	
gradlew	2019/11/22 9:24	Windows 批处理文件	3 KB	
HelloWorld.iml	2019/11/22 9:29	IML 文件	1 KB	
local.properties	2019/11/22 9:24	PROPERTIES 文件	1 KB	
settings.gradle	2019/11/22 9:24	GRADLE 文件	1 KB	

请问该文件夹的 5 个子文件夹中分别存放什么内容？（3 分）

3. 实验箱从 CPU 的 LED 灯程序的工程文件夹如下：

此电脑 > DATA1 (D:) > UP-Tech > STM32 > Exp > LED >	
名称	
App	
Board	
Chip	
Document	
Project	

请问该文件夹的 5 个子文件夹中分别存放什么内容？（3 分）

4. 以下为 CAN 总线双机通信接收程序的主函数，请问该程序中第 10)、12)、15)、18)、21) 行的作用是什么？（5 分）

```
1)  int main(int argc, char *argv[])
2)  {
3)      int s, nbytes, nbytes_send;
4)      struct sockaddr_can addr;
5)      struct ifreq ifr;
6)      struct can_frame frame_rev;
7)      struct can_frame frame_send;
8)      struct can_filter rfilter[1];
9)      int len = sizeof(addr);
10)     s = socket(PF_CAN, SOCK_RAW, CAN_RAW);
11)     strcpy(ifr.ifr_name, "can0");
```

```

12)    ioctl(s, SIOCGIFINDEX, &ifr);
13)    addr.can_family = AF_CAN;
14)    addr.can_ifindex = ifr.ifr_ifindex;
15)    bind(s, (struct sockaddr *)&addr, sizeof(addr));
16)    rfilter[0].can_id = 0x00;
17)    rfilter[0].can_mask = CAN_SFF_MASK;
18)    setsockopt(s, SOL_CAN_RAW, CAN_RAW_FILTER, &rfilter, sizeof(rfilter));
19)    while(1)
20)    {
21)        nbytes = read(s, &frame_rev, sizeof(frame_rev));
22)        if(nbytes > 0)
23)        {
24)            printf("ID=0x%X DLC=%d\n", frame_rev.can_id, frame_rev.can_dlc, frame_rev.data[0]);
25)        }
26)    }
27)    close(s);
28)    return 0;
29) }

```

5. 以下为 LED 灯驱动程序的头文件和全局变量，请问该程序的第 15)、16)、17) 行分别是做什么事情？(3 分)

```

1)    #include <linux/module.h>
2)    #include <linux/kernel.h>
3)    #include <linux/fs.h>
4)    #include <linux/init.h>
5)    #include <linux/miscdevice.h>
6)    #include <linux/delay.h>
7)    #include <linux/device.h>
8)    #include <linux/cdev.h>
9)    #include <linux/platform_device.h>
10)   #include <asm/irq.h>
11)   #include <linux/of.h>
12)   #include <linux/of_device.h>
13)   #include <linux/of_gpio.h>
14)   MODULE_LICENSE("GPL");
15)   #define DEVICE_NAME        "ledtest"
16)   #define DEVICE_MAJOR       231
17)   #define DEVICE_MINOR       0
18)   struct cdev *mycdev;
19)   struct class *myclass;
20)   dev_t devno;
21)   static unsigned int led_table [4] = {};

```

6. 以下是 RS-485 驱动程序的模块初始化和模块退出函数，请填写该程序中 2 个空格（第 11）、12）行）中的内容。（2 分）

```
1) static int __init gpio_uart485_init(void)
2) {
3)     printk("\n\nkzkuan__%s\n\n",__func__);
4)     return platform_driver_register(&gpio_uart485_device_driver);
5) }
6) static void __exit gpio_uart485_exit(void)
7) {
8)     printk("\n\nkzkuan__%s\n\n",__func__);
9)     platform_driver_unregister(&gpio_uart485_device_driver);
10) }
11) _____(1)_____(gpio_uart485_init);
12) _____(2)_____(gpio_uart485_exit);
```

7. 以下为按键（小键盘）程序的主函数，请说明该程序第 14）、16）、17）行的具体功能是什么？（3 分）

```
1) int main(int argc,char *argv[])
2) {
3)     int keys_fd;
4)     char ret[2];
5)     struct input_event t;
6)     keys_fd = open(argv[1], O_RDONLY);
7)     if(keys_fd<=0)
8)     {
9)         printf("open %s device error!\n",argv[1]);
10)        return 0;
11)    }
12)    while(1)
13)    {
14)        if(read(keys_fd, &t, sizeof(t)) == sizeof(t))
15)        {
16)            if(t.type == EV_KEY)
17)                if(t.value == 0 || t.value == 1)
18)                    printf("key %d %s\n",t.code,(t.value?"Pressed":"Released"));
19)        }
20)    }
21)    close(keys_fd);
22)    return 0;
23) }
```

8. 以下为步进电机程序的主函数，请说明该程序第 6)、7)、14)、16) 行的具体功能是什么？（4 分）

```
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;
15)    }
16)    munmap(cpld,0x04);
17)    close(mem_fd);
18)    return 0;
19) }
```

9. 以下为 Atlas 200 DK “创建首个 AI 应用” 中 graph.config 文件的内容，请说明该文件中第 8)、12)、25)、46) 行的作用。（4 分）

```
1) graphs {
2)     priority: 0
3)     engines {
4)         id: 611
5)         engine_name: "DataInput"
6)         side: HOST
7)         thread_num: 1
8)         so_name: "./libHost.so"
9)         ai_config {
10)            items {
11)                name: "path"
12)                value: "/home/ascend/AscendProjects/MyApp/resource/data/"
13)            }
14)            items {
15)                name: "target"
16)                value: "RC"
17)            }
18)        }
19)     }
```

```

18)     }
19) }
20) engines {
21)     id: 814
22)     engine_name: "ImagePreProcess"
23)     side: DEVICE
24)     thread_num: 1
25)     so_name: "./libDevice.so"
26)     ai_config {
27)         items {
28)             name: "resize_width"
29)             value: "224"
30)         }
31)         items {
32)             name: "resize_height"
33)             value: "224"
34)         }
35)     }
36) }
37) engines {
38)     id: 226
39)     engine_name: "MindInferenceEngine"
40)     side: DEVICE
41)     thread_num: 1
42)     so_name: "./libDevice.so"
43)     ai_config {
44)         items {
45)             name: "model_path"
46)             value: "/home/ascend/MindStudio-ubuntu/samples/modelfile/resnet18.om"
47)         }
48)     }
49) }
50) engines {
51)     id: 368
52)     engine_name: "SaveFilePostProcess"
53)     side: HOST
54)     thread_num: 1
55)     so_name: "./libHost.so"
56)     ai_config {
57)     }
58) }
59) connects {
60)     src_engine_id: 611
61)     src_port_id: 0
62)     target_engine_id: 814

```

```
63)      target_port_id: 0
64)      }
65)      connects {
66)          src_engine_id: 814
67)          src_port_id: 0
68)          target_engine_id: 226
69)          target_port_id: 0
70)      }
71)      connects {
72)          src_engine_id: 226
73)          src_port_id: 0
74)          target_engine_id: 368
75)          target_port_id: 0
76)      }
77)  }
```