a卷

选择题

1、（B）目录存放着 Linux 的源代码。 ***B、 /usr/src***

2、设备驱动程序是内核的一部分，下面的描述不正确的是：(D ) ***D、设备驱动程序可以使用户更加仔细的了解硬件的细节。***

3、以下哪种方式不是 Linux 操作系统中任务之间通信方式。（C ） ***C、邮件***

4、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***

5、不属于服务器端的准备过程的是（D） *D.accept()*

6、通常，用户编写的程序中所使用的地址是( A)。 ***A. 逻辑地址***

7、设备的打开、关闭、读、写等操作是由( D)完成的。 ***D．设备驱动程序***

8、多个进程的实体能存在于同一内存中，在一段时间内都得到运行。这种性质称作进程 的( B)。 *B．并发性*

9、Linux 操作系统支持多种设备，这些设备的驱动程序不包括以下的那一项特点（ C） *C、当系统启动及设备驱动初始化后，驱动程序将维护其控制的设备。如果一个特有 的设备驱动程序所控制的物理设备不存在，将会影响整个系统的运行*

10、每个设备文件都对应有两个设备号，而标识该设备的种类，也标识了该设备所使用的 驱动程序的识别号是指( A) ***A、主设备号***

二、判断题。注意：只判断不改错不得分（每小题 2 分，共 10 分）

1. 普通进程可以通过管道机制同守护进程进行通信。错误

2、在应用程序中不可以采用中断的方法来访问外部 I/O 设备。正确

3、命名管道在完成进程间通信的任务后，系统就将资源回收，不能再被任何进程读或写 了。错误

4、在 Linux 系统中，当一个进程 fork（）一个新的子进程时，父进程产生子进程时，只分 配一个指向父进程的指针向量，当子进程执行时才从父进程那复制一份为自己所用 正确

5、进程 ID 为 1 通常是 init 进程，在自举过程结束时由内核调

用。init 进程绝不会终止， 它是一个系统进程，并且以超级用户特权运行 错误

三、填空题（每空 1 分，共 20 分） 1、任务总共有五个状态：休眠状态、（就绪状态）、（运行状态）、中断服务状态和 （等待状态），在任何时候，一个任务只会处于其中的某一个状态。

2、嵌入式软件的分类：（系统）软件、（支撑）软件和（应用）软件。

3、 LINUX 使用的进程间通信方式有：管道和命名管道、（ 消息队列）、（共享内存 ）、（信号 ）、 套接字、信号量。

4、（任务（进程和线））是实现多道程序并发执行的必要基础。多个任务间的关系可以 分为（同步）和（通信）两种。

5、加载驱动的命令是：（insmod ），创建设备结点的命令是：（ mknod）。

6、在 linux 网络编程中，主要有两种通信方式：（ tcp协议通信）和（UDP协议通信 ）。

7、Linux 进程中包含 3 个段（ 数据段）、（堆栈段 ）和（代码段 ）。

8、查看本机 IP 地址的命令是（ifconfig ）

四简答题

1、 试说明 wait()与 waitPid()的区别。

（1）在一个子进程终止前， wait 使其调用者阻塞，而 waitpid 有一选择项，可使调用者不阻塞。 （2）waitpid 并不等待第一个终止的子进程—它有若干个选择项，可以控制它所等待的特定进程。 （3）实际上 wait 函数是 waitpid 函数的一个特例。 waitpid 函数提供了 wait 函数没有提供的三个功能： (1) waitpid 等待一个特定的进程(而 w a i t 则返回任一终止子进程的状态)。 (2) waitpid 提供了一个 w a i t 的非阻塞版本。有时希望取得一个子进程的状态，但不想阻塞。 (3) waitpid 支持作业控制（以 WUNTRACED 选择项）。

2、 什么是守护进程？守护进程的特征有哪些？

(5) 其他初始化工作 2、守护进程（ daemon）是生存期长的一种进程。它们常常在系统引导装入时起动，在系统关闭时 终止。因为它们没有控制终端，所以说它们是在后台运行的。 守护进程特征： （1）所有守护进程都以超级用户（用户 I D 为 0）的优先权运行。 （2）没有一个守护进程具有控制终端—终端名称设置为问号（？）、终端前台进程组 I D 设置为－1。 缺少控制终端可能是精灵进程调用了 s e t s i d 的结果。 （3）除 u p d a t e 以外的所有精灵进程都是进程组的首进程，对话期的首进程，而且是这些进程组 和对话期中的唯一进程。u p d a t e 是它所在进程组和对话期（中的唯一进程，但是该进程组的首进 程（可能也是该对话期的首进程）已经终止。 （4）所有这些守护进程的父进程都是 i n i t 进程

3、 I/O 端口设备的访问方法有哪两种？请简要说明。

第一种方法是直接使用 I/O 端口操作函数：在设备打开或驱动模块被加载时就申请 I/O 端口区域， 之后用 inb（）、outb（）等进行端口访问，最后在设备关闭或驱动模块被卸载的时候释放 I/O 端口 范围。 第二种方法是将 I/O 端口映射为内存来进行访问：在设备打开或驱动模块被加载时，申请 I/O 端口 区域并使用 ioport\_map()映射到内存，之后使用 I/O 内存的函数进行端口访问，最后在最后在设备关 闭或驱动模块被卸载的时候释放 I/O 端口并释放映

4、简述用信号集的编程模型实现信号编程的基本

答：基本流程为： 1、 创建 timer，编写 function. 2、 为 time 的 expires,data,function 赋值 3、 调用 add\_time 将 timer 加入列表 4、 定时器到期后，function 被执行 5、 在涉及到 timer 的时候适当地调用 del\_timer。 删除 timer 或修改 timer 的 expires： Struct timer\_list mytimer; init\_timer(&mytimner); mytimer.function=mytimer\_ok; mytime.expires=jiffies+HZ/100; add\_timer(&mytimer); del\_timer(&mytimer); 各个系统调用可以根据情况放置于不同的内核模块函数中。

1、假设已经有如下文件：LED 驱动源文件（led.c），编译好的驱动目标文件（led.o）， LED 主设备号为 254，已编译好的驱动测试文件（testled）根文件系统（ramdisk.gz）, 内核文件（zImage）,试从开发模式和产品模式分别加载上述文件。

开发模式：开发正常启动系统后， （1）insmod led.o （2）mknod c 254 0 (3)./testled

产品模式： （1）解压根文件系统，挂载一个临时目录，把 led.o 拷贝到/dev，即 cp led.o /mnt/loop/dev (2) vi /etc/rcS 在最后一行添加 insmod led.o mknod c 254 0 （3）把 led.c 文件放入 linux 文件夹中的/driver 目录下，修改 config.m 文件，make menuconfig 编译内核，生成新的 zImage 文件