

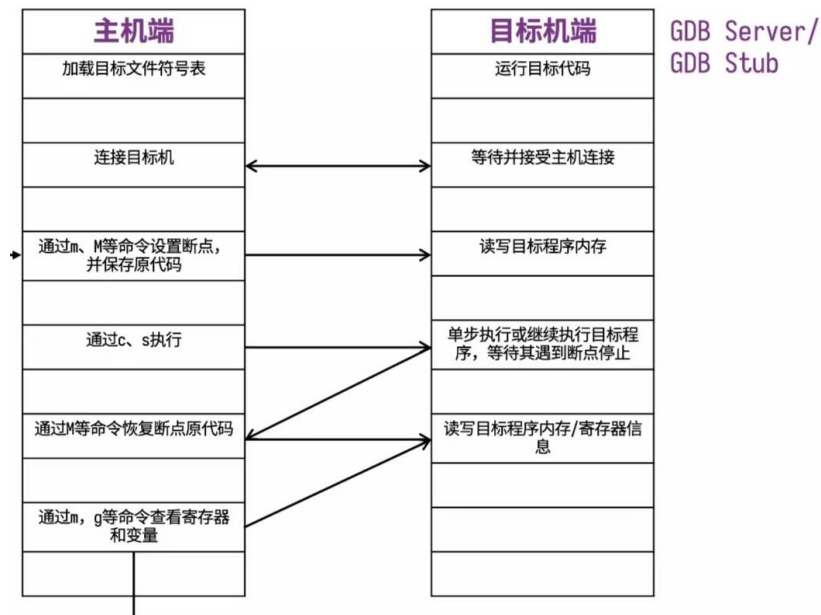
【嵌入式系统】习题 2

1. 简述使用交叉编译的主要原因（5 分）

采用交叉编译的主要原因有两个：首先，在嵌入式系统开发的起始阶段，目标平台尚未建立，因此需要做交叉编译，来生成所需的 **BootLoader** 以及操作系统内核；其次，当目标平台能够启动之后，由于目标机资源的限制，不能提供足够的资源供编译过程使用，因而只好将编译工程转移到高性能的宿主机中进行。

2. ARM 微处理器的寄存器在 Thumb 状态与 ARM 状态下相同的是（**R0-R7**）（5 分）

3. 如何使用调试器 gdb 进行代码调试,具体的调试方法有哪些？（10 分）



查看文件、设置和查看断点、运行代码、查看变量值、单步运行、恢复程序运行

4. 开发嵌入式系统时，需要构建一个宿主机-目标机的开发环境。若目标机是裸机，那么为了调试和下载软件需要将调试仿真器连接到目标机的哪一种接口？为什么？（10 分）

- A. SPI 接口
- B. 以太网接口
- C. JTAG 接口
- D. USB 接口

正确答案：C

【解析】由于程序调试时不占用目标机的资源，因此目标机的初始启动程序调试、硬件接口的调试，一般都采用 **JTAG**。至于裸机状态的目标机当然更需要采用 **JTAG** 调试方式了。故本题选择 C。

5. 下列关于 **Bootloader** 的陈述中，不正确的是（**B**）。（5 分）

- A. **Bootloader** 主要完成内存加电自检、外设存在自检、初始化外围设备、加载和启动操作系统等功能
- B. **QNX** 是支持多种嵌入式 CPU 的 **Bootloader** 程序
- C. 大多数从 **Flash** 存储器上启动的 **bootloader** 采用 **stage1** 和 **stage2** 两个阶段完成操作系统

的引导加载

D. Bootloader 的实现依赖于 CPU 的体系结构

6. 在基于 ARM 微处理器为核心的硬件平台上, 开发其 Linux 环境下的应用程序 exp1.c。若需要编译后生成 exp1.c 对应的汇编程序, 应使用的命令是 (A)。(5 分)

- A. arm-linux-gcc -S exp1.c
- B. arm-linux-gcc -oexp1 exp1.c
- C. arm-linux-gcc -c exp1.c
- D. arm-linux-gcc -g -oexp1 exp1.c

7. ARM Linux 中, 在进入中断响应之前, CPU 将依次进行哪些操作, 从下列选项中选出正确的进行排序。(10 分)

- a. 将 sp 指向 0x18。
- b. 将 cpsr 原来的内容装入 spsr_irq, 即中断模式的 spsr; 同时改变 cpsr 的内容使 CPU 运行于中断模式, 并关闭中断。
- c. 将进入中断响应前的内容装入 r14_irq, 即中断模式的 lr, 使其指向中断点。
- d. 从中断请求寄存器获取中断源。
- e. 将 pc 指向 0x18。
- f. 将堆栈指针 sp 切换成中断模式的 sp_irq。
- g. 从中断控制寄存器获取中断源。

答案:cbfe

8. 内存碎片分为: 内部碎片和外部碎片。内部碎片就是已经被分配出去 (能明确指出属于哪个进程) 却不能被利用的内存空间; 外部碎片指的是还没有被分配出去 (不属于任何进程), 但由于太小了无法分配给申请内存空间的新进程的内存空闲区域。请分别阐述内部碎片和外部碎片产生的原因。Buddy 方案对应的是哪种碎片? 简要说明 Buddy 算法的工作原理。(15 分)

【内部碎片】

内部碎片就是已经被分配出去 (能明确指出属于哪个进程) 却不能被利用的内存空间;

内部碎片是处于 (操作系统分配的用于装载某一进程的内存) 区域内部或页面内部的存储块。占有这些区域或页面的进程并不使用这个存储块。而在进程占有这块存储块时, 系统无法利用它。直到进程释放它, 或进程结束时, 系统才有可能利用这个存储块。

单道连续分配只有内部碎片。多道固定连续分配既有内部碎片, 又有外部碎片。

【外部碎片】

外部碎片指的是还没有被分配出去 (不属于任何进程), 但由于太小了无法分配给申请内存空间的新进程的内存空闲区域。

外部碎片是处于任何两个已分配区域或页面之间的空闲存储块。这些存储块的总和可以满足当前申请的长度要求, 但是由于它们的地址不连续或其他原因, 使得系统无法满足当前申请。

产生外部碎片的一种情况:

举个例子, 在内存上, 分配三个操作系统分配的用于装载进程的内存区域 A、B 和 C。假设, 三个内存区域都是相连的。故而三个内存区域不会产生外部碎片。现在假设 B 对应的进程执行完毕了操作系统随即收回了 B, 这个时候 A 和 C 中间就有一块空闲区域了。多道可变连续分配只有外部碎片。

总之，在内存上，外部碎片是位于任何两个操作系统分配的用于装载进程的内存区域或页面之间的空闲区域，内部碎片是位于一个操作系统分配的用于装载进程的内存区域或页面内部空闲区域。

Buddy 算法：(解决外碎片)

把内存中所有页面按照 2^n 划分，其中 $n=0\sim5$ ，每个内存空间按 1 个页面、2 个页面、4 个页面、8 个页面、16 个页面、32 个页面进行六次划分。划分后形成了大小不等的存储块，称为页面块，简称页块，包含一个页面的页块称为 1 页块，包含 2 个页面的称为 2 页块，依次类推。

每种页块按前后顺序两两结合成一对 Buddy“伙伴”。系统按照 Buddy 关系把具有相同大小的空闲页面块组成页块组，即 1 页块组、2 页块组.....32 页块组。每个页块组用一个双向循环链表进行管理，共有 6 个链表，分别为 1、2、4、8、16、32 页块链表。分别挂到 `free_area[]` 数组上。

内存分配时，系统按照 Buddy 算法，根据请求的页面数在 `free_area[]` 对应的空闲页块组中搜索。若请求页面数不是 2 的整数次幂，则按照稍大于请求数的 2 的整数次幂的值搜索相应的页面块组。

当相应页块组中没有可使用的空闲页面块时就查询更大一些的页块组，在找到可用的页块后分配所需要的页面。当某一空闲页面被分配后，若仍有剩余的空闲页面，则根据剩余页面的大小把他们加入到相应页面组中。内存页面释放时，系统将其作为空闲页面看待，检查是否存在与这些页面相邻的其他空闲页块，若存在，则合为一个连续的空闲区按 Buddy 算法重新分组。

9.Ext3 日志文件系统的特点是：（ABCD）（5 分）

- A.高可用性
- B.数据的完整性
- C.数据转换快
- D.多日志模式

10. ARM Linux 系统中，中断处理程序进入 C 代码以后，ARM 的处于（ A ）工作模式。（5 分）

- A. 超级用户（SVC）
- B. 中断（IRQ）
- C. 快速中断（IRQ）
- D. 和进入中断之前的状态有关系

11. 在 ARM Linux 体系中，用来处理外设中断的异常模式是（ C ）（5 分）

- A. 软件中断（SWI）
- B. 未定义的指令异常
- C. 中断请求（IRQ）
- D. 中止中断请求

12. 简述 Ext3 的日志模式与工作原理（10 分）

Ext3 有多种日志模式，一种工作模式是对所有的文件数据及 metadata（定义文件系统中数据的数据,即数据的数据）进行日志记录（`data=journal` 模式）；另一种工作模式则是只对 metadata 记录日志，而不对数据进行日志记录，也即所谓 `data=ordered` 或者 `data=writeback`

模式。系统管理人员可以根据系统的实际工作要求，在系统的工作速度与文件数据的一致性之间作出选择。

13.简述 ARM Linux 的各个进程状态的转变关系，试举例每种转变如何触发。（10 分）

运行态：进程占有处理器正在运行的状态。进程已获得 CPU，其程序正在执行。在单处理机系统中，只有一个进程处于执行状态；在多处理机系统中，则有多个进程处于执行状态。

就绪态：进程具备运行条件，等待系统分配处理器以便运行的状态。当进程已分配到除 CPU 以外的所有必要资源后，只要再获得 CPU，便可立即执行，进程这时的状态称为就绪状态。在一个系统中处于就绪状态的进程可能有多个，通常将它们排成一个队列，称为就绪队列。

等待态：又称阻塞态或睡眠态，指进程不具备运行条件，正在等待某个时间完成的状态。也称为等待或睡眠状态，一个进程正在等待某一事件发生（例如请求 I/O 而等待 I/O 完成等）而暂时停止运行，这时即使把处理机分配给进程也无法运行，故称该进程处于阻塞状态。

新建态：对应于进程被创建时的状态，尚未进入就绪队列。创建一个进程需要通过两个步骤：**1.为新进程分配所需要的资源和建立必要的管理信息。2.设置该进程为就绪态，并等待被调度执行。**

终止态：指进程完成任务到达正常结束点，或出现无法克服的错误而异常终止，或被操作系统及有终止权的进程所终止时所处的状态。处于终止态的进程不再被调度执行，下一步将被系统撤销，最终从系统中消失。终止一个进程需要两个步骤：**1.先对操作系统或相关的进程进行善后处理（如抽取信息）。2.然后回收占用的资源并被系统删除。**

NULL→新建态：执行一个程序，创建一个子进程。

新建态→就绪态：当操作系统完成了进程创建的必要操作，并且当前系统的性能和虚拟内存的容量均允许。

运行态→终止态：当一个进程到达了自然结束点，或是出现了无法克服的错误，或是被操作系统所终结，或是被其他有终止权的进程所终结。

运行态→就绪态：运行时间片到；出现有更高优先权进程。

运行态→等待态：等待使用资源；如等待外设传输；等待人工干预。

就绪态→终止态：未在状态转换图中显示，但某些操作系统允许父进程终结子进程。

等待态→终止态：未在状态转换图中显示，但某些操作系统允许父进程终结子进程。

终止态→NULL：完成善后操作。