

第2章 操作系统硬件基础 习题参考答案

一、选择题

1. 对于一个单处理器系统来说, 允许若干个进程同时执行, 轮流占用处理器, 称它们为 (D) 的。
A 顺序执行 B 同时执行 C 并行执行 D 并发执行
2. 用户从终端上输入一条命令, 即产生了 (B)。
A 程序性中断事件 B 外部中断事件 C 输入输出中断事件 D 自愿性中断事件
3. 自愿性中断事件是由 (C) 引起的。
A 程序中使用了非法操作码 B 程序中访问地址越界 C 程序中使用了一条访管指令 D 程序中除数为“0”
4. 中断装置根据 (D) 判别有无强迫中断事件发生。
A.指令操作码为访管指令 B.基址寄存器 C.限长寄存器 D.中断寄存器
5. 中断处理程序在保护现场和分析中断原因后,一般会请求系统创建相应的处理事件进程,排入(D)。
A.等待队列 B.运行队列 C.后备队列 D.就绪队列
6. (B) 程序可执行特权指令。
A.同组用户 B.操作系统 C.特权用户 D.一般用户
7. 下列指令中, 只能在管态下执行的是 (C)。
A.读时钟日期 B.访管指令 C.屏蔽中断指令 D.取数指令
8. 下列 (B) 指令不在核心态下运行。
A.屏蔽所有的中断 B.读时钟 C.设置时钟日期 D.改变存储器映像图

9. 处理机处于管态时,处理机可以执行的指令应该是 (C).

A.仅限于非特权指令 B.仅限于特权指令 C.全部指令 D. 仅限于访

管指令映像图

10. 用户程序在目态下使用特权指令将引起的中断属于 (B)。

A.硬件故障中断 B.程序中断 C.外部中断 D. 访问中断

二、填空题

1. 强迫性中断事件有硬件故障中断、_程序性中断_、外部中断和_输入输出中断_事件.

2. 自愿性中断是由进程中执行一条_访管指令_引起的。

3. 中断响应的三项工作为检查是否有中断事件发生,_保护断点等信息_和_启动操作系统的中断处理程序_.

4. 中断装置要通过检查_中断寄存器_才能识别是否有强迫性中断事件发生。

5. 中断处理程序一般只做一些简单的处理,然后请求系统创建_相应的处理事件_的进程,排入_就绪_队列.

6. 用户进程从目态（常态）转换为管态（特态）的唯一途径是_中断_。

7. 为了赋予操作系统某些特权,使得操作系统更加安全可靠地工作,实际系统中区分程序执行的两种不同的运行状态是_管态_和 _目态_ ,其中_目态_程序不能执行特权指令。

8. 当中央处理器处于_管态_时可以执行包括特权指令在内的一切机器指令，当中央处理器处于_目态_时不允许执行特权指令。因此，操作系统占用中央处理器时，应让中央处理器在_管态_下工作，而用户程序占用中央处理器时，应让中央处理器在_目态_下工作。如果中央处理器在_目态_工作，却取得了一条

特殊指令，此时中央处理器将拒绝执行该指令，并形成“非法操作”事件，终端设备识别该事件之后，转交给操作系统处理，由操作系统通知用户“程序中有非法指令”。

三、问答题

1. 中断事件的处理应做哪几件事？

处理器执行指令时,根据操作码是否是访管指令,来确定是否有自愿性中断事件发生。若有,则通知中断装置;对于强迫性中断,中断装置只要检查中断寄存器的内容,若非“0,表示有中断事件发生。

2. 中断系统有何作用？

中断系统在实现进程并发执行,维护系统正常工作,进行故障处理及满足定时要求等方面起着重要的作用。

3. 什么是特权指令？举例说明之。如果允许用户进程执行特权指令会带来什么后果？

在现代计算机中,一般都提供一些专门供操作系统使用的特殊指令,这些指令只能在管态执行,称为特权指令。这些指令包括:停机指令、置 PSW 指令、中断操作指令(开中断、关中断、屏蔽中断)、输入输出指令等。用户程序不能执行这些特权指令。如果允许用户程序执行特权指令,有可能干扰操作系统的正常运行,甚至有可能使整个系统崩溃。

4. 在计算机系统中,为什么要区分管态和目态 (核心态与用户态)?

为了正确管理和控制系统资源,为了操作系统的软件不被用户系统资源破坏,对操作系统的状态进行分类,使用户不能轻易篡改系统,所以分成管态和目态。

5. 什么是系统调用？它与普通的函数调用有什么不同？请简要说明系统调用的

处理流程。

Linux 内核中设置了一组用于实现各种系统功能的子程序，称为系统调用。用户可以通过系统调用命令在自己的应用程序中调用它们。

区别在于系统调用由操作系统核心提供，运行于核心态;而普通的函数调用由函数库或用户自己提供，运行于用户态。

系统调用的原理是进程先用适当的值填充寄存器，然后调用一个特殊的指令，这个指令会跳到一个事先定义的内核中的一个位置(当然，这个位置是用户进程可读但是不可写的)。在 Intel CPU 中，这个由中断 0x80 实现。硬件知道了位置，你就不是在限制模式下运行的用户，而是作为操作系统的内核。进程可以跳转到的内核位置叫做 `sysenter_call`。这个过程检查系统调用号，这个号码告诉内核进程请求哪种服务。然后，它查看系统调用表(`sys_call_table`)找到所调用的内核函数入口地址。接着，就调用函数，等返回后，做一些系统检查，最后返回到进程(或到其他进程，如果这个进程时间用尽)。