山东大学 软件 学院

操作系统 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201400301291 | 姓名： 雷超 | | 班级： 四班 |
| 实验题目：进程同步实验 | | | |
| 实验学时：2 | | 实验日期： 2016-04-26 | |
| 实验目的：  加深对于并发协作进程同步与互斥概念的理解，观察和体验并发进程同步与互斥操作的效果，分析与研究经典进程同步与互斥问题的实际解决方案。了解Linux操作系统IPC进程同步工具的用法，练习并发协作进程的同步与互斥操作的编程与调试技术。 | | | |
| 硬件环境：  00:00.0 Host bridge: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor DRAM Controller (rev 06)  00:01.0 PCI bridge: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor PCI Express x16 Controller (rev 06)  00:02.0 VGA compatible controller: Intel Corporation 4th Gen Core Processor Integrated Graphics Controller (rev 06)  00:03.0 Audio device: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor HD Audio Controller (rev 06)  00:14.0 USB controller: Intel Corporation 8 Series/C220 Series Chipset Family USB xHCI (rev 05)  00:16.0 Communication controller: Intel Corporation 8 Series/C220 Series Chipset Family MEI Controller #1 (rev 04)  00:1a.0 USB controller: Intel Corporation 8 Series/C220 Series Chipset Family USB EHCI #2 (rev 05)  00:1b.0 Audio device: Intel Corporation 8 Series/C220 Series Chipset High Definition Audio Controller (rev 05)  00:1c.0 PCI bridge: Intel Corporation 8 Series/C220 Series Chipset Family PCI Express Root Port #1 (rev d5)  00:1c.1 PCI bridge: Intel Corporation 8 Series/C220 Series Chipset Family PCI Express Root Port #2 (rev d5)  00:1c.2 PCI bridge: Intel Corporation 8 Series/C220 Series Chipset Family PCI Express Root Port #3 (rev d5)  00:1c.4 PCI bridge: Intel Corporation 8 Series/C220 Series Chipset Family PCI Express Root Port #5 (rev d5)  00:1d.0 USB controller: Intel Corporation 8 Series/C220 Series Chipset Family USB EHCI #1 (rev 05)  00:1f.0 ISA bridge: Intel Corporation HM86 Express LPC Controller (rev 05)  00:1f.2 SATA controller: Intel Corporation 8 Series/C220 Series Chipset Family 6-port SATA Controller 1 [AHCI mode] (rev 05)  00:1f.3 SMBus: Intel Corporation 8 Series/C220 Series Chipset Family SMBus Controller (rev 05)  01:00.0 3D controller: NVIDIA Corporation GM107M [GeForce GTX 860M] (rev a2)  08:00.0 Network controller: Intel Corporation Wireless 3160 (rev 93)  09:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller (rev 10)  0a:00.0 Unassigned class [ff00]: Realtek Semiconductor Co., Ltd. RTS5249 PCI Express Card Reader (rev 01) | | | |
| 软件环境：  Linux EVA 4.6.2-gentoo-EVOLUTION x86\_64 Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz GenuineIntel GNU/Linux  dev-libs/gmp-6.0.0a  dev-libs/mpfr-3.1.3\_p4  dev-libs/mpc-1.0.2-r1  sys-devel/gcc-4.9.3 , 5.3.0 , 5.4.0  sys-libs/glibc-2.22-r4  sys-devel/clang-3.5.0-r100  app-editors/emacs-24.5-r1  sys-kernel/gentoo-sources-4.6.2  sys-kernel/linux-headers-4.6  sys-apps/systemd-226-r2  ABI\_X86=”32 64” | | | |
| 实验步骤与内容：  抽烟者问题。三个抽烟者进程，每个抽烟者不断的卷起烟草并且抽烟。抽烟者抽烟需要三种原材料，并且每个人只能自身生成其中的一种。这种产品是固定不能改变的。系统中有两个供应者，无限提供三种材料，但是每次只能提供三种中的两种，得到缺失的材料的抽烟者进行抽烟行为。并且通过消息队列向供应者发送消息要求提供另外的两种材料。重复进行。 | | | |
| 结论分析与体会：  此次实验并不困难。主要是熟悉IPC通信时候所使用的三大套件来完成任务。在代码层面，主要是通过信号灯来达成进程同步和进程互斥的目的来分别保证逻辑顺序和临界区资源的合理访问；使用消息队列来完成抽烟者发送消息和供应者接受消息；使用共享内存来完成资源的存取。  通过分析可以得到，对于供应者而言，他们两个是相互排斥的。不可以同时写入数据，这是因为可能在随机生产资源的时候，生成同一种资源在系统是是对资源先进行读取然后写入，这就存在临界区问题，所以使用一个互斥锁进行保护；消费者之间由于设计的时候认为抽烟者生成的两个产品是一种，共计三种，所以抽烟者之间不存在互斥的关系；但是抽烟者和同一时间的提供对应产品的供应者是互斥的，所以也需要加上一把互斥锁。这就是此次实验的主要分析。  进程同步也是进程之间直接的制约关系，是为完成某种任务而建立的两个或多个线程，这个线程需要在某些位置上协调他们的工作次序而等待、传递信息所产生的制约关系。进程间的直接制约关系来源于他们之间的合作。 | | | |

附件：