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：线程和进程/线程管道通信实验 | | | |
| 实验学时：2 | | 实验日期： 2016-04-12 | |
| 实验目的：  通过Linux系统中线程和管道通信机制的实验，加深对于线程控制和管道通信概念的理解，观察和体验并发/线程间的通信和协作效果，练习利用无名管道进行线程/进程的编程和调试技术。 | | | |
| 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：  需要完成的实验是完成二元函数的计算，建立3个并发的协作线程（或者进程），分别完成f（x，y） ， f（x），f（y），其中使用父进程（主线程）完成，f（x，y） = f（x） + f（y）。子进程（线程）分别完成不同任务。  此次实验我使用线程操作技术，因为线程操作所付出的开销对于操作系统而言，比进程小的多，并且在日后的具体使用中值得去学习线程的操作技术。所以此次实验使用两个函数分别使用递归计算来完成任务。然后父线程等待子线程结束从管道中读取到最后的值进行相加，完成任务。 | | | |
| 结论分析与体会：  此次实验体现出了线程协作和线程通信中的重要特征和功能，并且在Linux操作系统中更加突出的体现了出来。在多线程OS中，通常是在一个进程中包括多个线程，每个线程都是作为利用CPU的基本单位，是花费最小开销的实体。由于线程很“轻”，故线程的切换非常迅速且开销小（在同一进程中的）。  在同一进程中的各个线程，都可以共享该进程所拥有的资源，这首先表现在：所有线程都具有相同的[地址空间](https://www.baidu.com/s?wd=%E5%9C%B0%E5%9D%80%E7%A9%BA%E9%97%B4&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9rjc4rjRdmW9bnhRsmhDv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHnvrHT1PjfznWRsPWbdP10drf" \t "/home/elvis/Documents\\x/_blank)（进程的[地址空间](https://www.baidu.com/s?wd=%E5%9C%B0%E5%9D%80%E7%A9%BA%E9%97%B4&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9rjc4rjRdmW9bnhRsmhDv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHnvrHT1PjfznWRsPWbdP10drf" \t "/home/elvis/Documents\\x/_blank)），这意味着，线程可以访问该[地址空间](https://www.baidu.com/s?wd=%E5%9C%B0%E5%9D%80%E7%A9%BA%E9%97%B4&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9rjc4rjRdmW9bnhRsmhDv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHnvrHT1PjfznWRsPWbdP10drf" \t "/home/elvis/Documents\\x/_blank)的每一个虚地址；此外，还可以访问进程所拥有的已打开文件、定时器、信号量机构等。由于同一个进程内的线程[共享内存](https://www.baidu.com/s?wd=%E5%85%B1%E4%BA%AB%E5%86%85%E5%AD%98&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9rjc4rjRdmW9bnhRsmhDv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHnvrHT1PjfznWRsPWbdP10drf" \t "/home/elvis/Documents\\x/_blank)和文件，所以线程之间互相通信不必调用内核。  线程的划分尺度小于进程，使得多线程程序的并发性高。 另外，进程在执行过程中拥有独立的[内存单元](https://www.baidu.com/s?wd=%E5%86%85%E5%AD%98%E5%8D%95%E5%85%83&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9rjc4rjRdmW9bnhRsmhDv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHnvrHT1PjfznWRsPWbdP10drf" \t "/home/elvis/Documents\\x/_blank)，而多个线程[共享内存](https://www.baidu.com/s?wd=%E5%85%B1%E4%BA%AB%E5%86%85%E5%AD%98&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9rjc4rjRdmW9bnhRsmhDv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHnvrHT1PjfznWRsPWbdP10drf" \t "/home/elvis/Documents\\x/_blank)，从而极大地提高了程序的运行效率 | | | |

附件：