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：进程控制实验 | | | |
| 实验学时：2 | | 实验日期： 2016-04-05 | |
| 实验目的：  加深对于进程并发执行概念的理解。实践并发进程的创建和控制方法。观察和体验进程的动态控制特性。进一步理解进程生命期期间创建，变换，撤销状态变换的过程。掌握进程控制的方法，了解父子进程间等的控制和协作关系。练习Linux系统中进程创建与控制有关的系统调用的编程和调试技术。 | | | |
| 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：  根据示例实验进行实践，然后深入理解系统调用所体现的进程的特征和功能，了解真实的操作系统中进程的生命周期，进程实体控制。  根据对示例实验进程控制的新的理解和认识，设计并且分析独立实验的需求，根据需求的特性完成代码编写，主要使用的是信号对于父子进程间的操控和影响进行实验。  主要使用的系统调用函数有 fork()，waitpid()，exec()，signal()，kill()。完整代码见附件。  总体来说，此次实验通过两次fork() ，来创建两个子进程。通过signal()注册信号来注册关于SIGINT的函数，子进程进入主体函数后pause()，等待父进程有序唤醒来达到后创建的进程先执行命令的结果。 | | | |
| 结论分析与体会：  通过此次实验，我进一步加深对于进程并发执行概念的理解。初步认识并且实践并发进程的创建和控制方法。通过观察和体验进程的动态控制特性，进一步理解进程生命期期间创建，变换，撤销状态变换的过程，从而掌握进程控制的方法，了解父子进程间等的控制和协作关系。练习Linux系统中进程创建与控制有关的系统调用的编程和调试技术。  这次的实验相对简单的，主要涉及的设计方面的是使用父进程控制两个子进程，这一点是由于两个子进程无法直接得到兄弟进程的进程号码所决定的，当然也可以通过特殊的方式获得，但是那样会显得累赘，所以设计为父进程控制子进程。  其中涉及到waitpid()父亲进程回收子进程的问题。如果不使用waitpid()，父进程结束后若子进程没有结束则变成孤儿进程最后由init进程充当其父进程并且定期回收，若子进程先结束，父进程没有回收，则变成僵尸进程。 | | | |

附件：