山东大学 计算机科学与技术 学院

操作系统 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202200130243 | 姓名：樊荣圣 | | 班级：4班 |
| 实验题目：实验一：进程控制 | | | |
| 实验学时：2 | | 实验日期：2024.10.9 | |
| 实验目的：加深对于进程并发执行概念的理解。实践并发进/线程的创建和控制方法。观 察和体验进程的动态特性。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，了解父子进程间的控制和协作关系。练习 Linux 系 统中进/线程创建与控制有关的系统调用的编程和调试技术。 | | | |
| 实验环境：VirtualBox、Ubuntu | | | |
| 源程序清单：  示例程序：pctl.h, pctl.c, pctl.out;  独立实验：pctl2.h, pctl2.c, pctl2.out | | | |
| 编译及运行结果：  在独立程序中设置了10次循环，这里就只展示第1次循环和第10次循环的结果，每一轮次都先创建ls子进程和ps子进程，但先执行ps命令后执行ls命令，如下： | | | |
| 示例程序运行结果： | | | |
| 问题及收获：   1. 对sleep(5)能否去掉的思考：sleep(5)的作用是让父进程等待5秒，以便给子进程足够的时间来准备处理SIGINT信号，如果去掉sleep(5)语句，可能会出现子进程调用pause（）后，父进程会立即发送SIGINT信号，子进程接收信号并退出，不会执行execve()调用。 2. 反映出的特征和功能：父进程和子进程可以并发执行，这体现了进程的并发性；每个进程都会有自己的地址空间和资源，子进程通过执行exec()系统调用加载一个新的程序，这体现了进程的独立性；子进程可以通过pause()系统调用暂停执行，体现了进程的异步性。 3. 关于进程的生命期：进程的生命期从创建开始，到终止结束。在实验中，我们通过fork()系统调用创建一个子进程，然后通过execve()调用执行新的程序，如果执行成功，它会正常结束子程序。我们也可以通过exit()调用结束进程。例如程序中的这两段代码：      1. 关于进程的实体：进程的实体包括PCB、程序计数器、寄存器集合、堆栈等。在实验中，由fork创建的子进程继承了父进程的地址空间，但是通过execve加载了新的程序，所以这改变了进程的实体内容。 2. 关于进程的状态控制：进程状态一般包括就绪态、运行态和终止态。在实验中，子进程通过pause调用进入阻塞态，等待信号的到来。当信号到来时，子进程被唤醒，进入就绪态，等待CPU调度。父进程则通过waitpid等待子进程的结束。这些都体现了进程状态的控制。 3. 对进程概念的和并发概念的新理解：通过本次实验，确实更深刻地理解了进程是操作系统进行资源分配和调度的一个独立单位。而并发性不仅仅是指多个程序同时运行，还应该包括它们能够在任意时刻被操作系统中断和恢复的能力。在微观层面上并发其实是进程快速交替执行。在独立程序中最开始并没有在执行ls命令进程中添加sleep(1)语句，     因此最后输出的结果很能体现出ls命令进程和ps命令进程的并发性，它们各自进程的信息输出是交替的，如下：    在运行ps进程和ls进程时应该是快速地交替执行，因此这里并没有实现ps命令总在ls命令之前执行。故而在第一个进程执行ls命令之前添加上了一条sleep（1）语句，让其暂停一会，就实现了ps命令的先执行。从这里就可以很好地看出进程并发执行在微观层面上的实质，其实就是交替执行的。   1. 子进程的创建和执行新程序：子进程通过fork系统调用创建的，它会复制父进程的地址空间；子进程通过execve系统调用执行新的程序，这个调用会替换当前进程的地址空间，加载新的程序。 2. 信号的机理：信号是操作系统提供的一种软件中断机制，用于处理异步事件。在实验中，信号用于唤醒处于等待中的子进程。信号可以被阻塞、捕捉和忽略。 3. 利用信号实现进程控制：信号可以用于多种进程控制，入终止进程、暂停进程执行、处理键盘中断等。在实验中，信号用于唤醒暂停的子进程，从而继续执行。例如示例代码中的这段程序： | | | |