|  |
| --- |
| 1. 实验目的及实验内容   （本次实验所涉及并要求掌握的知识；实验内容；必要的原理分析） |
| **实验目的：**掌握POSIX共享内存的概念，使用和相关信息输出  **实验内容：**参考3.10&3.11，编写一个具有父子进程的小程序，并能将共享内存段的信息进行正确输出。 |
| 1. 实验环境及实验步骤   （本次实验所使用的器件、仪器设备等的情况；具体的实验步骤） |
| **实验环境：**  1. vmware workstation 15.0.0   1. ubuntu-18.10-desktop 2. 内核版本：linux-4.18.14   **实验步骤：**   1. **参考3.10&3.11，学习基本知识：**   习题3.10主要运用了3.5.1小结介绍的POSIX共享内存方法，具体涉及的函数有shmget()，shmat()，shmdt()，shmctl()。在参考课本内容后，在网上查找相关资料，了解各函数原型与参数的意义。  习题3.11介绍了来自数据结构struct shmid\_ds的关于POSIX进程间通信机制的状态信息。可以通过shmctl()获得此结构。   1. **确定编程目标：**   在了解了相关信息后觉得仿照课本，编写父子进程，子进程中生成Fibonacci序列，父进程接收命令行参数。  父进程同时执行错误检查，创建共享内存段，将共享内存段附加到地址空间，将参数传递给子进程，等待其生产Fibonacci序列，并在父进程中输出序列，以及struct shmid\_ds的各种信息。   1. **代码实现：**   最终形成了Fibonacci.cpp。  **头文件以及数据结构部分：**  其中重要的函数及状态类型所需头文件见注释  define Max 10，定义fibonacci数列最长为10.若有需要可以更改。  fib数组存储fibonacci序列，fib\_size为序列长度    **主函数部分：**  argc为终端输入的参数个数，atoi()完成字符串到int的转换。  首先判断了输入参数个数及范围是否合法。  shmget()原型：int shmget(key\_t key, size\_t size, int shmflg)  在本程序中创建一个共享内存对象并返回共享内存标识符赋予segment\_id变量。  共享内存段属性既读又写。  shmat()原型：void \*shmat(int shmid, const void \*shmaddr, int shmflg)  连接共享内存标识符为shmid的共享内存，连接成功后把共享内存区对象映射到调用进程的地址空间，随后可像本地空间一样访问。  shmaddr这里取0，让内核自己决定一个合适的地址位置。  返回附加好的共享内存地址到指针shared  之后赋size值。    **创建子进程以及子进程部分：**  fork出子进程。id==0为子进程，输出共享内存地址以及计算fibonacci序列。  之后shmdt断开链接。    **主进程部分：**  先等待子进程结束，之后输出共享内存中的fibonacci序列。  然后shmctl将shmid\_ds赋值到shmbuffer中。  之后输出相关信息。 |
| 1. 实验过程分析   （实验分工，详细记录实验过程中发生的故障和问题，进行故障分析，说明故障排除的过程及方法。根据具体实验，记录、整理相应的数据表格等） |
| **遇到的问题及解决：**  1.编译时无法识别I\_IRUSR ,I\_IWUSR。  解决：加上<sys/stat.h>   1. 不清楚shmctl如何获得struct shmid\_ds   解决：查阅资料得知shmclt原型int shmctl(int shmid, int cmd, struct shmid\_ds \*buf)，cmd 为IPC\_SET时：改变共享内存的状态，把buf所指的shmid\_ds结构中的uid、gid、mode复制到共享内存的shmid\_ds结构内  实验结果记录：  在linux gcc编译运行，结果如下图所示。  IMG_256  用ipcs命令验证。结果正确  IMG_256 |