**深 圳 大 学 实 验 报 告**

**课程名称： 操作系统**

**实验项目名称： 实验2处理机调度**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 张滇**

**报告人： 梁润宇 学号： 2021220003 班级： 2班**

**实验时间： 2023年4月4日-2023年4月30日**

**实验报告提交时间： 2024年4月25日**

**教务部制**

|  |
| --- |
| **实验目的与要求：**  **实验目的：**  掌握进程同步与通信实验  加深对进程同步于通信操作的直观认识；  掌握Linux操作系统的进程、线程机制和编程接口；  掌握Linux操作系统的进程和线程间的同步和通信机制；  掌握经典同步问题的编程方法；  **实验要求：**  操作部分（参考）：   1. 借助google工具查找资料，学习使用Linux进程间通信：管道、消息队列、共享内存； 2. 设计编写以下程序，着重考虑其同步问题：    1. 一个程序（进程）从客户端读入按键信息，一次将“一整行”按键信息保存到一个共享存储的缓冲区内并等待读取进程将数据读走，不断重复上面的操作；    2. 另一个程序（进程）生成两个进程，用于显示缓冲区内的信息，这两个进程并发读取缓冲区信息后将缓冲区清空（一个进程的两次显示操作之间可以加入适当的时延以便于观察）。    3. 在两个独立的终端窗口上分别运行上述两个程序，展示其同步与通信功能，要求一次只有一个任务在操作缓冲区。    4. 运行程序，记录操作过程的截屏并给出文字说明。   要求使用posix信号量来完成这里的生产者和消费者的同步关系。  实验报告要求：   1. 按学校统一格式 2. 需要给出具体命令和自行编写的程序的源代码 3. 程序的设计需要给出设计思路或流程框图 4. 实验操作的截图需要有必要的说明文字   **说明：**  （1）本次实验课作业满分为100分，占总成绩的比例（待定）。  （2）本次实验课作业截至时间2024年4月30日（周二）23:59。  （3）报告正文：请在指定位置填写，本次实验不需要单独提交源程序文件。  （4）个人信息：WORD文件名中的“姓名”、“学号”，请改为你的姓名和学号；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”等信息。  （5）提交方式：请在BLACKBOARD平台中按时提交；延迟提交不得分。  （6）发现抄袭（包括复制&粘贴整句话、整张图），该次作业记零分。  （7）期末考试阶段补交无效。 |

|  |
| --- |
| **1. 头文件shm\_com\_sem.h**  图1-头文件代码  头文件内定义了一个共享内存结构体shared\_mem\_st，用于在进程间共享数据。同时，使用了信号量来实现进程间的同步和互斥操作。  **2.生产者producer.c**  由于已经给了代码框架，实验报告内只进行填空处的分析。  填空1：获取共享内存区，并挂入内存  图2-获取共享内存  shmget 函数用于创建一个共享内存段或获取一个已经存在的共享内存段的标识符。它接受三个参数：共享内存的 key（这里是 2024），内存段的大小（sizeof(struct shared\_mem\_st)），以及权限标志（0666|IPC\_CREAT 表示创建或打开一个所有用户可读可写的共享内存段）。如果操作失败，shmget 返回 -1。  检查 shmget 的返回值，如果返回 -1，说明共享内存创建或获取失败，此时程序输出错误信息并调用 exit(EXIT\_FAILURE) 终止程序。  如果 shmget 成功，返回一个共享内存标识符 shmid，接着使用 shmat 函数将共享内存连接到当前进程的地址空间。shmat 函数的第一个参数是 shmget 返回的共享内存标识符 shmid，第二个参数是希望共享内存连接到进程中的地址（通常是 (void \*) 0 表示让系统自动选择合适的地址），第三个参数是标志位（通常为 0）。  如果 shmat 返回值是 (void\*) -1，说明共享内存连接失败，程序输出错误信息并调用 exit(EXIT\_FAILURE) 终止程序。  在终端内通过ipcs -m可以查看共享内存情况。  图3-共享内存情况  键(key)的值为7e8，转换成十进制就是我设置的key = 2024。连接数是3，表示有3个进程正在访问此共享内存，分别是1个生产者进程，2个消费者进程。  填空2：创建信号量  此处就是按照头文件里已经定义好的信号量名字进行创建。  图4-信号量创建  sem\_open函数定义如下：  sem\_open(const char \*name, int oflag, mode\_t mode, unsigned int value);  其中name是信号量的名字，必须是以/开头的路径名，在系统中唯一标识一个信号量；oflag是打开标志，用于指定信号量的操作，可以是O\_CREAT（如果信号量不存在则创建），O\_EXCL（如果信号量已存在则返回错误），O\_RDWR（可读可写方式打开）等；mode是权限位，指定信号量的访问权限；value是信号量的初始值。  (1) sem\_queue：互斥信号量，初值为1，用于保护对缓冲区的访问。在生产者和消费者进程访问缓冲区之前，会使用 sem\_wait(sem\_queue) 来尝试获取这个信号量，如果获取成功（即信号量的值大于 0），则表示可以访问缓冲区；访问完成后使用 sem\_post(sem\_queue) 来释放信号量，以便其他进程可以访问缓冲区。  (2)sem\_queue\_empty：空缓冲区信号量，初值为16，用于表示还有多少空的缓冲区可以写入数据。在生产者进程向缓冲区写入数据之前，会使用 sem\_wait(sem\_queue\_empty) 来尝试获取一个空缓冲区，获取成功后写入数据，并使用 sem\_post(sem\_queue\_empty) 来增加空缓冲区的数量，表示已经写入了数据。  (3)sem\_queue\_full：满缓冲区信号量，初值为0，用于表示已经有多少缓冲区被写满了。在消费者进程从缓冲区读取数据之前，会使用 sem\_wait(sem\_queue\_full) 来尝试获取一个已满的缓冲区，获取成功后读取数据，并使用 sem\_post(sem\_queue\_full) 来增加满缓冲区的数量，表示已经读取了数据。  填空3：将输入的行写入缓冲区，要有信号量操作  首先等待空缓冲区信号量，然后等待互斥信号量。完成写入之后，释放互斥信号量，释放满缓冲区信号量。  图5-写入缓冲区  填空4：清理信号量，结束共享内存在本进程的挂载映像，删除共享内存区域  图6-生产者收尾工作  图7-生产者流程图  **3.消费者customer.c**  按实验要求，消费者需要使用2个进程并发操作缓冲区。那么使用fork创建子进程来实现2个进程的要求。  填空1：获取共享内存区，并挂入内存  填空2：获得producer创建的3个信号量  填空3：信号量操作，打印消费内容及进程号，发现quit退出  填空4：释放信号量  填空5，6为父进程内操作，与子进程完全一致  最后，使用kill(0, SIGKILL) 向当前进程组中的所有进程发送 SIGKILL 信号，而父子进程属于同一个进程组。这样就实现了结束消费者程序。  填空1：和生产者完全一致  填空2：获取producer创建的3个信号量，还是用sem\_open函数。当oflag = O\_CREAT时，若name指定的信号量不存在时，则会创建一个，而且后面的mode和value参数必须有效。若name指定的信号量已存在，则直接打开该信号量，同时忽略mode和value参数。由于这3个信号量已经存在了，就不必写mode和value  图8-打开信号量  填空3：  图9-消费者信号量操作  等待满缓冲区信号量，等待互斥信号量，从共享内存的读位置获取数据，输出，然后更新到下一行。其中在同一个进程的2次输出间加入3秒钟时延，便于观察并发。释放互斥信号量，释放空缓冲区信号量。发现quit，则break，退出循环。  填空4：清理信号量，使用sem\_unlink函数，和生产者处代码一样  填空5，6：父进程操作，除了输出部分有所不同，其他都和填空3，4一样  收尾工作：使用kill(0, SIGKILL) 向当前进程组中的所有进程发送 SIGKILL 信号，结束消费者程序。  图10-运行结果  运行结果中可以看到，生产者从终端读入按键信息，消费者生成的两个进程，并发地显示缓冲区内的信息。  ++++++++++++++++++++++++++++++++++++++++++++++++++++++  **其他（例如感想、建议等等）。**  感想：本实验中，学习了共享内存、有名信号量的操作，对进程间通信、同步有了更多了解。 |

深圳大学学生实验报告用纸

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：张滇  2024年 4 月 30 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。