**操作系统**

**实 验 报 告**

实验名称： shell命令解释器的实现

学 员： 侯华玮 学 号： 202102001015

培养类型： 无军籍学员 年 级： 21级

专业：计算机科学与技术（天河拔尖班） 所属学院： 计算机学院

指导教员： 文艳军 职 称： 教授

实 验 室： 实验日期： 2023.05.24

国防科技大学训练部制

《实验报告》填写说明

1．学员完成人才培养方案和课程标准要所要求的每个实验后，均须提交实验报告。

2．实验报告封面必须打印，报告内容可以手写或打印。

3．实验报告内容编排及打印应符合以下要求：

（1）采用A4（21cm×29.7cm）白色复印纸，单面黑字打印。上下左右各侧的页边距均为3cm；缺省文档网格：字号为小4号，中文为宋体，英文和阿拉伯数字为Times New Roman，每页30行，每行36字；页脚距边界为2.5cm，页码置于页脚、居中，采用小5号阿拉伯数字从1开始连续编排，封面不编页码。

（2）报告正文最多可设四级标题，字体均为黑体，第一级标题字号为4号，其余各级标题为小4号；标题序号第一级用“一、”、“二、”……，第二级用“（一）”、“（二）” ……，第三级用“1.”、“2.” ……，第四级用“（1）”、“（2）” ……，分别按序连续编排。

（3）正文插图、表格中的文字字号均为5号。

一、实验目的和内容

实验目的：

● 加深对进程同步与互斥概念的认识；

● 掌握信号量的使用，并应用它解决生产者——消费者问题；

● 掌握信号量的实现原理。

实验内容：

● 在 Ubuntu 下编写程序，用信号量解决生产者——消费者问题；

● 在 0.11 中实现信号量，用生产者—消费者程序检验之。

二、操作方法与实验步骤

**信号量系统调用**

sem\_t\* sys\_sem\_open(const char\* name, unsigned int value)  
int sys\_sem\_wait(sem\_t \*sem)  
int sys\_sem\_post(sem\_t \*sem)  
int sys\_sem\_unlink(const char\* name)

* sem\_open() 的功能是创建一个信号量，或打开一个已经存在的信号量。
  + sem\_t 是信号量类型，根据实现的需要自定义。
  + name 是信号量的名字。不同的进程可以通过提供同样的 name 而共享同一个信号量。如果该信号量不存在，就创建新的名为 name 的信号量；如果存在，就打开已经存在的名为 name 的信号量。
  + value 是信号量的初值，仅当新建信号量时，此参数才有效，其余情况下它被忽略。当成功时，返回值是该信号量的唯一标识（比如，在内核的地址、ID 等），由另两个系统调用使用。如失败，返回值是 NULL。
* sem\_wait() 就是信号量的 P 原子操作。如果继续运行的条件不满足，则令调用进程等待在信号量 sem 上。返回 0 表示成功，返回 -1 表示失败。
* sem\_post() 就是信号量的 V 原子操作。如果有等待 sem 的进程，它会唤醒其中的一个。返回 0 表示成功，返回 -1 表示失败。
* sem\_unlink() 的功能是删除名为 name 的信号量。返回 0 表示成功，返回 -1 表示失败。

**生产者消费者问题**

利用上面实现的信号量系统调用，编写一个应用程序“pc.c”来模拟经典的生产者 消费者之间的同步。

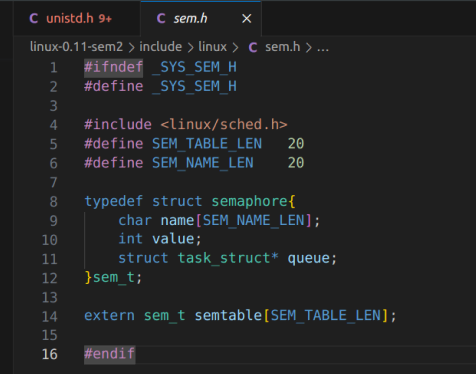
在这个程序中，要建立 1 个生产者进程，5 个消费者进程；用文件建立一个共享缓冲区；生产者进程依次向缓冲区写入整数 0,1,2,...,499；每个消费者进程从缓冲区中读取 100 个数，每读取一个数字就打印到标准输出上；缓冲区文件最多只能保存 10 个数。

**实验步骤**

**1 添加信号量系统调用**

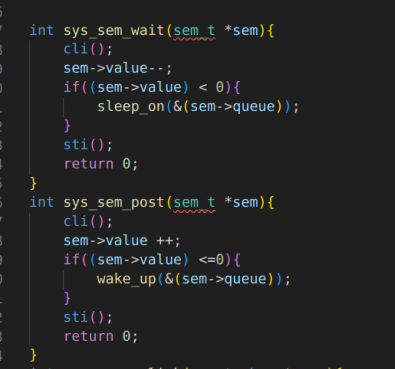
1. 修改unist.h与sys.h文件，添加有关信号量系统调用的信息。
2. 添加信号量头文件sem.h与信号量系统调用实现文件sem.c。

头文件中声明了信号量的结构体，包括name, value, queue三个成员。

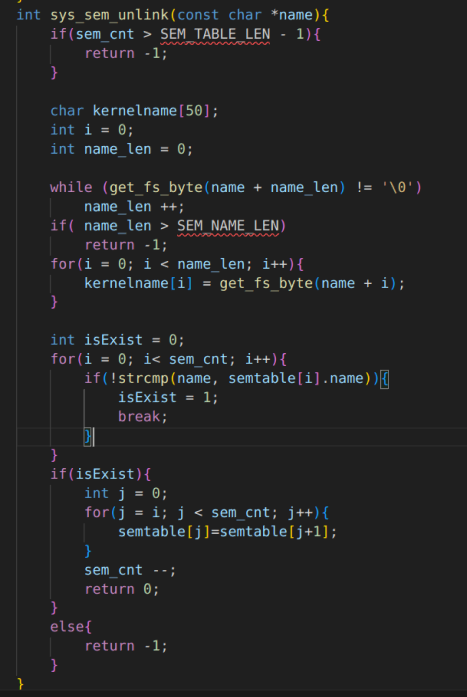


信号量系统调用实现：

sys\_sem\_open

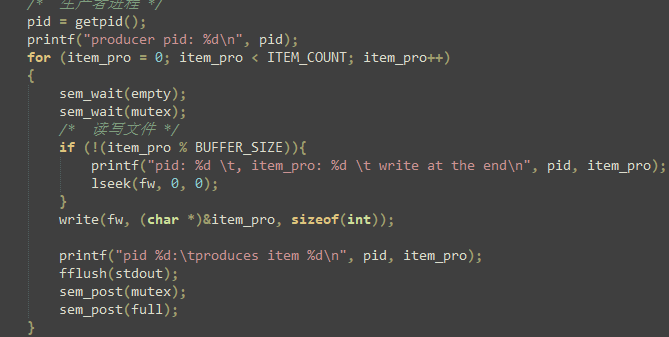


sys\_sem\_unlink

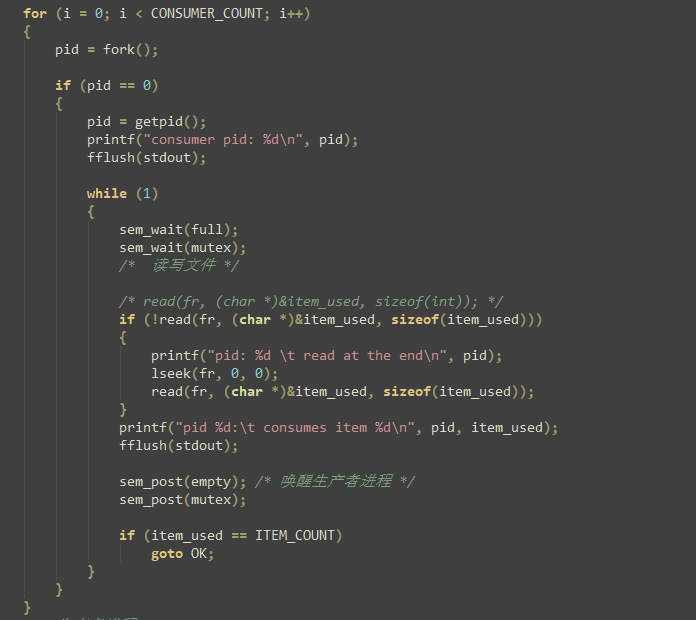


**2 编写生产者-消费者程序**

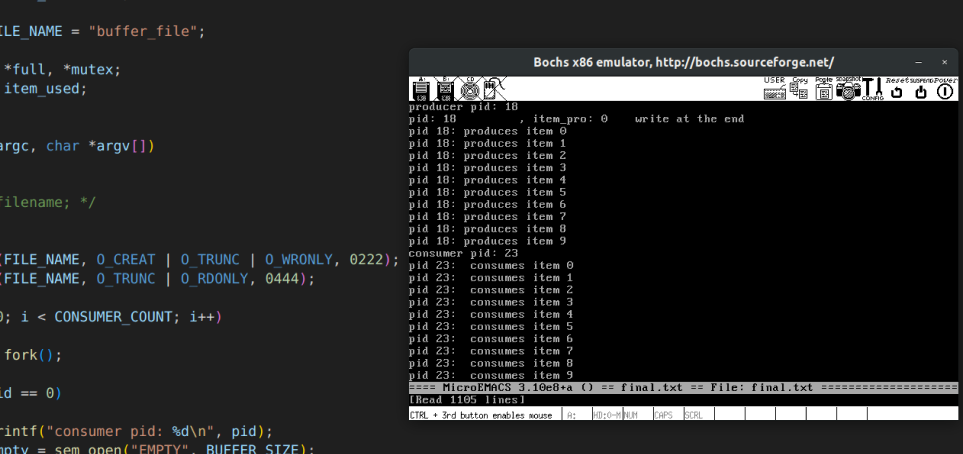
**生产者-父进程**



**消费者-子进程**



三、实验结果与分析



经测试，该生产者消费者程序成功实现了预定功能，说明信号量相关系统调用实现正确。

四、问题与建议