进程同步实验（生产者-消费者问题）

基于 openEuler 操作系统，设计一个 C 语言程序，利用信号量机制解决有限缓冲的生产者-消费者问题。

1. 实验内容

具体内容参考教材《操作系统概念》第六章项目：生产者-消费者问题。

2. 实验要求

(1) 缓冲区

(a) 缓冲区存储结构建议采用固定大小的数组表示，并作为环形队列处理。

(b) 缓冲区的访问算法按照课本 6.6.1 节图 6.10、图 6.11 进行设计。

(2) 主函数 main()

(a) 主函数需要创建一定数量的生产者线程与消费者线程。线程创建完毕后，主函数将睡眠一段时间，并在唤醒时终止应用程序。

(b) 主函数需要从命令行接受三个参数：睡眠时长、生产者线程数量、消费者线程数量。

(3) 生产者与消费者线程

(a) 生产者线程：随机睡眠一段时间，向缓冲区插入一个随机数。

(b) 消费者线程：随机睡眠一段时间，从缓冲区去除一个随机数。

3. API 介绍

(1) int sem_init(sem_t *sem, int pshared, unsigned int value)

(a) 头文件：<semaphore.h>

(b) 功能：初始化信号量

(c) 参数：

sem_t *sem：所需初始化信号量 sem 的地址。

int pshared：表明该信号量是否被同一进程下的线程或其他进程共享。0 表示该信号量可以在同一进程下的线程所共享；如果不是 0 则表示该信

号量可以在进程间共享。

unsigned int value：信号量初始值。

(d) 返回值：初始化成功则返回 0，失败则返回-1。

(2) int sem_wait(sem_t *sem)

(a) 头文件：<semaphore.h>

(b) 功能：如果信号量的值大于零，则减量继续进行，函数立即返回。 如果信号量当前的值为零，则调用将阻塞，直到有可能执行减量操作为止。

(c) 参数：信号量 sem 的地址。

(d) 返回值：运行成功则返回 0，失败返回-1。

(3) int sem_post(sem_t *sem)

(a) 头文件：<semaphore.h>

(b) 功能：解锁信号量 sem。

(c) 参数：信号量 sem 的地址。

(d) 返回值：运行成功则返回 0，失败返回-1。

(4) int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void *(*start_routine)(void *), void *arg)

(a) 头文件：<pthread.h>

(b) 功能：在调用 pthread_create()函数的进程中创建一个新的线程。

(c) 参数：

pthread_t *thread：当线程创建成功时，存储新线程的 ID。

const pthread_attr_t *attr：attr 指向一个 pthread_attr_t 结构，该结构的内容用于决定新建的线程的属性。如果 attr 为 NULL，则新线程为默认属性。

void *(*start_routine)(void *)：新线程通过调用 strat_routine()函数开始执行。

void *arg：向函数 start_routine()传递的唯一参数。当 arg 为 NULL 时，表明没有参数传递。

(d) 返回值：函数运行成功返回 0，失败则返回一个错误编号。