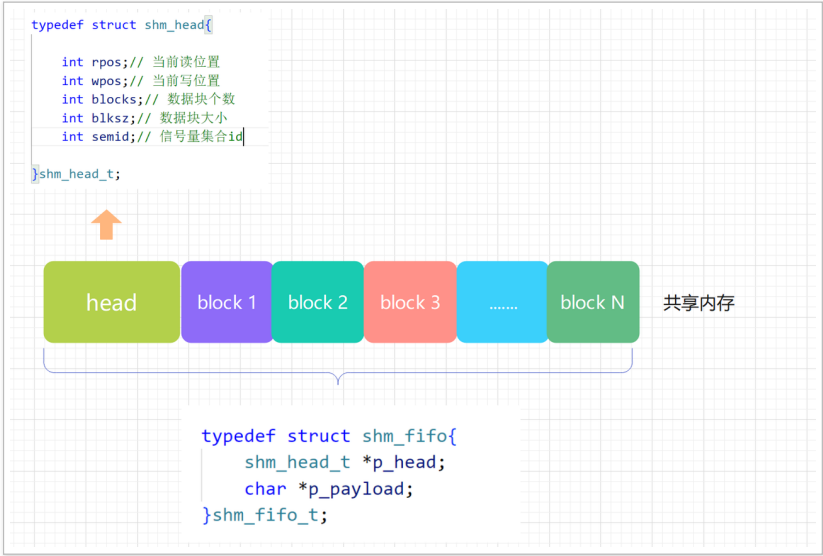


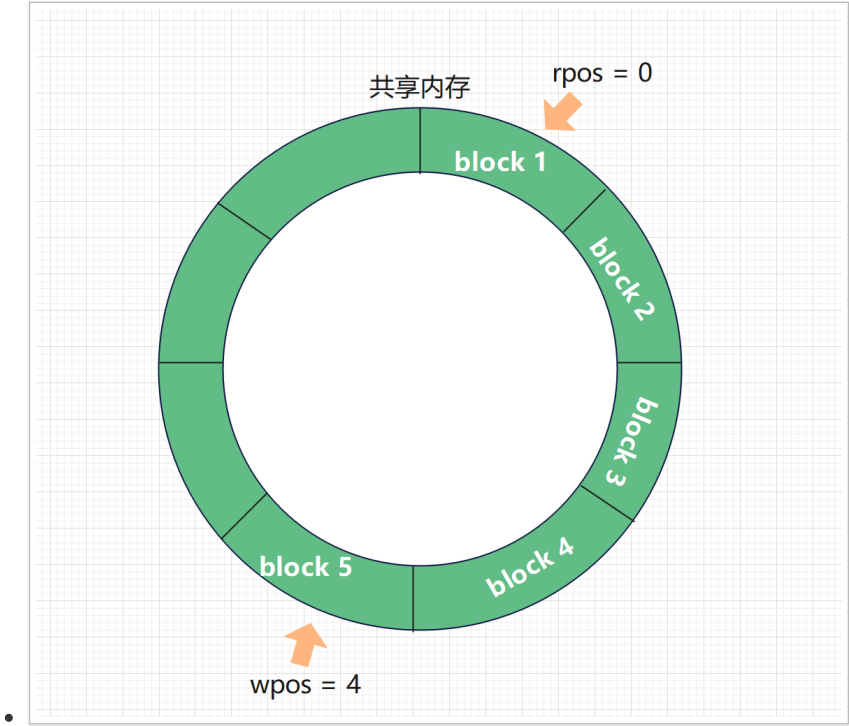
2.2 环形队列设计 (二)- 环形队列初始化与销毁_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 2.2 环形队列设计 (二)- 环形队列初始化与销毁涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

- 环形队列的基本数据结构主要分为两个区域

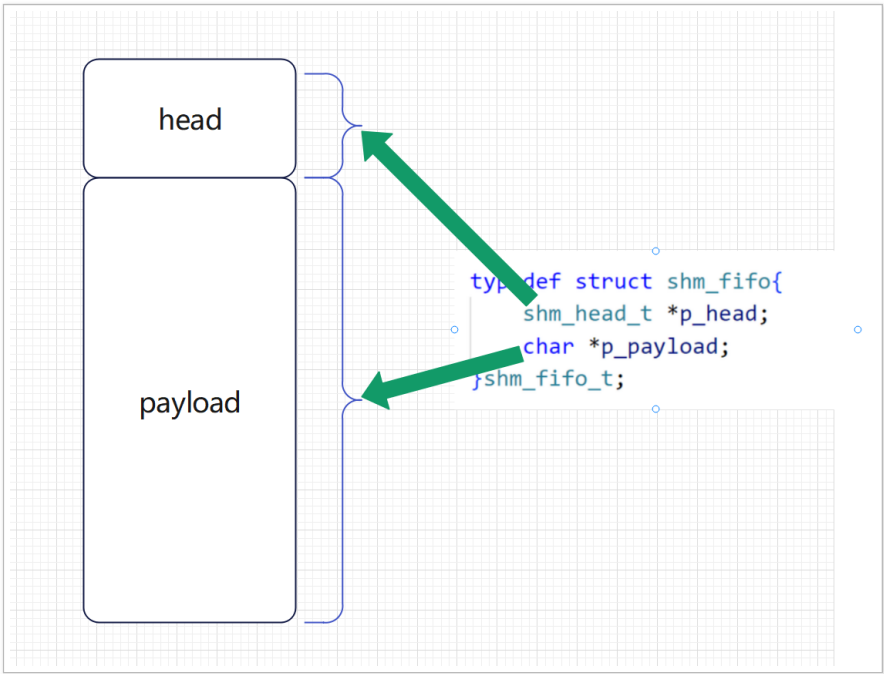


- head 区域: 存储读写位置等信息, 具体定义如下:
- ```
typedef struct shm_head{
 int rpos;
 int wpos;
 int blocks;
 int blksize;
 int semid;
}shm_head_t;
```
- payload 区域 : 存储传输的数据, 数据被分为多个 block, 每个 block 的大小为 blksize



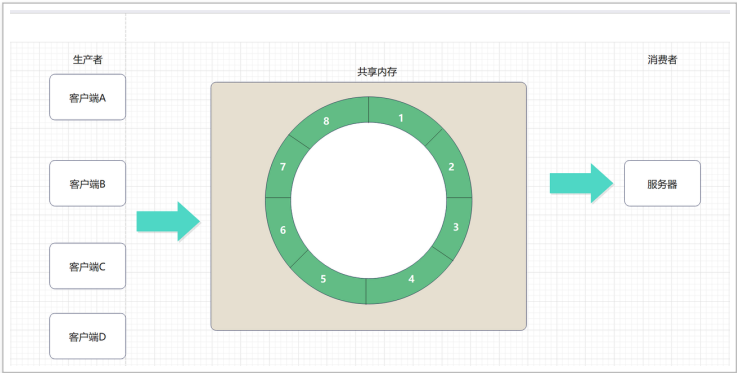
- 将两个区域合并定义到一个结构体，方便管理，具体如下:

```
typedef struct shm_fifo{
 shm_head_t *p_head;
 char *p_payload;
}shm_fifo_t;
```

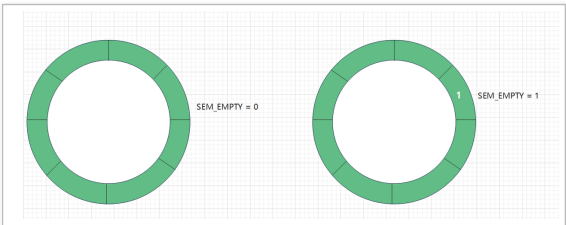


- 环形队列的初始化，主要做的事情如下:
  - 分配环形队列数据结构空间
  - `shm_fifo_t *p_shm_fifo = (shm_fifo_t *)malloc(sizeof(p_shm_fifo));`
  - 创建共享内存
  - `s_creat_type = shm_create(shm_sz,&g_shmid);`
  - 映射共享内存
  - `p_shm_fifo->p_head = (shm_head_t *)shm_at(g_shmid);`

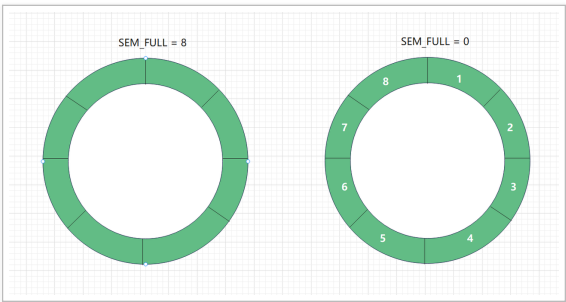
- 
- 对 head 区域进行初始化
- ```
p_shm_fifo->p_head->rpos = 0;
p_shm_fifo->p_head->wpos = 0;
p_shm_fifo->p_head->blocks = blocks;
p_shm_fifo->p_head->blksz = blksz;
p_shm_fifo->p_head->semid = sem_create(3,values);
```
- 在项目中，客户端相当于生产者，服务器相当于消费者



-
- 创建信号量集合，并包含 3 个信号量
 - 使用一个信号量 (SEM_MUTEX) 用于共享内存的互斥
 - 使用两个信号量 (SEM_EMPTY 与 SEM_FULL) 用于环形队列的同步
 - SEM_EMPTY = 0, 则表示队列为空 (没有资源可以访问), 消费者进程阻塞, 等待生产者进程存储数据到队列中, 并会让 SEM_EMPTY 的值增加, 阻塞消费者进程则可以唤醒消费



-
- SEM_FULL = blocks, 则表示队列为满
 - 每有一个数据节点存入队列中, 则 SEM_FULL--
 - 当 SEM_FULL = 0 时, 队列为满, 则生产者进程阻塞
 - 当消费者进程消费后, 会让 SEM_FULL 的值增加



-
-
-
-
- ```
p_shm_fifo->p_head->semid = sem_create(3,values);
```

## 完整实现如下:

- shm\_fifo\_t \*shmfifo\_init(int blocks,int blksz)  
{  
    int shm\_sz = 0;  
    enum shm\_creat\_status s\_creat\_type;  
    unsigned short values[] = {0,blocks,1};  
  
    shm\_fifo\_t \*p\_shm\_fifo = (shm\_fifo\_t \*)malloc(sizeof(p\_shm\_fifo));  
  
    if (!p\_shm\_fifo)  
        return NULL;  
  
    shm\_sz = blocks \* blksz + sizeof(shm\_head\_t);  
  
    s\_creat\_type = shm\_create(shm\_sz,&g\_shmid);  
  
    if (s\_creat\_type != SHM\_CREAT\_ERROR){  
        p\_shm\_fifo->p\_head = (shm\_head\_t \*)shm\_at(g\_shmid);  
  
        if (s\_creat\_type == SHM\_CREAT\_NEW){  
  
            p\_shm\_fifo->p\_head->rpos = 0;  
            p\_shm\_fifo->p\_head->wpos = 0;  
            p\_shm\_fifo->p\_head->blocks = blocks;  
            p\_shm\_fifo->p\_head->blksz = blksz;  
            p\_shm\_fifo->p\_head->semid = sem\_create(3,values);  
        }  
  
        p\_shm\_fifo->p\_payload = (char \*) (p\_shm\_fifo->p\_head + 1);  
  
        return p\_shm\_fifo;  
    }  
  
    return NULL;  
}

- 环形队列的销毁主要将信号量、共享内存、环形队列结构对象进行释放, 具体实现如下:

```
void shmfifo_destroy(shm_fifo_t *fifo)
{
 sem_del(fifo->p_head->semid);
 shm_dt(fifo->p_head);
 shm_del(g_shmid);
 free(fifo);
}
```

---

全文完

本文由 简悦 SimpRead 优化, 用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta, 点击查看详细说明

