OpenHarmony内核开发—消息队列

前言

本节主要介绍：

- 消息队列的相关概念
- 消息队列的运作机制
- 如何利用消息队列实现任务间数据通信

目录

消息队列基本概念

消息队列的概念：

消息队列，是一种常用于任务间通信的数据结构，实现了接收来自任务或中断的不固定长度的消息，并根据不同的接口选择传递消息是否存放在自己空间。任务能够从队列里面读取消息，当队列中的消息是空时，挂起读取任务；当队列中有新消息时，挂起的读取任务被唤醒并处理新消息。

用户在处理业务时，消息队列提供了异步处理机制，允许将一个消息放入队列，但并不立即处理它，同时队列还能起到缓冲消息作用。

LiteOS中使用队列数据结构实现任务异步通信工作，具有如下特性：

- 消息以先进先出方式排队，支持异步读写工作方式。
- 读队列和写队列都支持超时机制。
- 发送消息类型由通信双方约定，可以允许不同长度（不超过队列节点最大值）消息。
- 一个任务能够从任意一个消息队列接收和发送消息。
- 多个任务能够从同一个消息队列接收和发送消息。
- 当队列使用结束后，如果是动态申请的内存，需要通过释放内存函数回收。

消息队列运作机制

运作原理

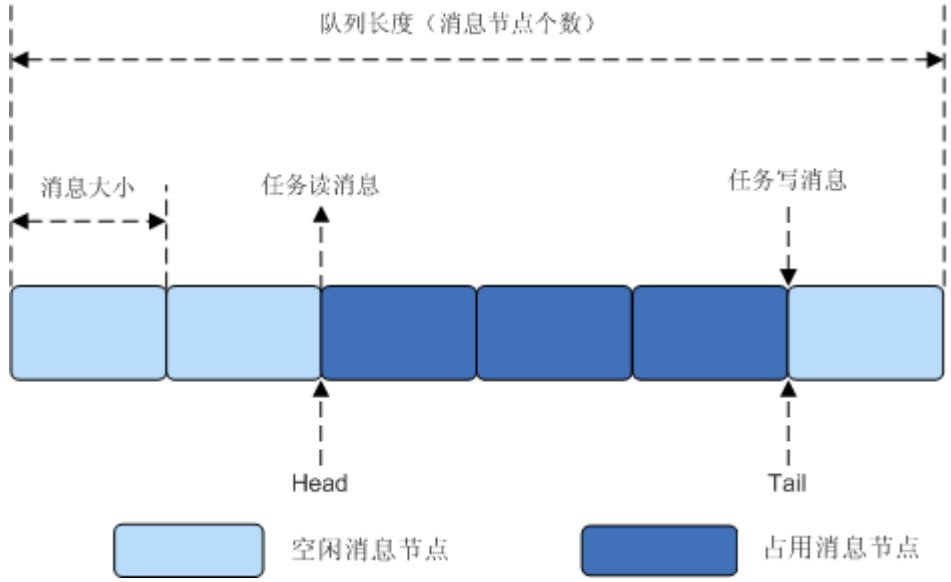
创建队列时，根据用户传入队列长度和消息节点大小来开辟相应的内存空间以供该队列使用，返回队列ID。

在队列控制块中维护一个消息头节点位置Head和一个消息尾节点位置Tail来表示当前队列中消息存储情况。Head表示队列中被占用消息的起始位置。Tail表示队列中被空闲消息的起始位置。刚创建时Head和Tail均指向队列起始位置。

写队列时，根据Tail找到被占用消息节点末尾的空闲节点作为数据写入对象。

读队列时，根据Head找到最先写入队列中的消息节点进行读取。

删除队列时，根据传入的队列ID寻找到对应的队列，把队列状态置为未使用，释放原队列所占的空间，对应的队列控制头置为初始状态。



实现消息队列功能

cmsis_os2的API消息队列接口简介：

接口名	功能描述
osMessageQueueNew	创建消息队列
osMessageQueuePut	发送消息
osMessageQueueGet	获取消息
osMessageQueueDelete	删除消息队列

创建消息队列：osMutexNew (const osMutexAttr_t *attr);

发送消息：osMutexAcquire (osMutexId_t mutex_id, uint32_t timeout);

获取消息：osMutexRelease (osMutexId_t mutex_id);

删除消息队列：osMutexDelete (osMutexId_t mutex_id);

消息队列扩展实验

扩展实验代码

```
void Thread_MsgQueue1(void *argument)
{
  (void)argument;
  uint8_t num = 0;
  //do some work...
  msg.Buf = "Hello BearPi-HM_Nano!";

  while (1)
  {
    msg.Idx = num;
    osMessageQueuePut(mid_MsgQueue, &msg, 0U, 0U);
    num++;

    //suspend thread
    osThreadYield();
    osDelay(100);
  }
}
```

```
void Thread_MsgQueue2(void *argument)
{
  (void)argument;
  osStatus_t status;
  uint32_t count;
  while (1)
  {
    //Insert thread code here...
    count = osMessageQueueGetCount(mid_MsgQueue);
    printf("message queue get count: %d\r\n",count);
    if(count == MSGQUEUE_OBJECTS)
    {
      osMessageQueueDelete(mid_MsgQueue);
    }
    //wait for message
    status = osMessageQueueGet(mid_MsgQueue, &msg, NULL, 0U);
    if (status == osOK)
    {
      printf("Message Queue Get msg idx:%d buf:%s\n", msg.Idx,msg.Buf);
    }
    osDelay(300);
  }
}
```

本节小结

- 1、了解消息队列的概念
- 2、掌握如何创建和删除消息队列
- 3、掌握如何利用消息队列实现任务间数据通信

谢谢观看

开 源 从 小 熊 派 开 始

OPEN-SOURCE STARTED WITH THE BEARPI