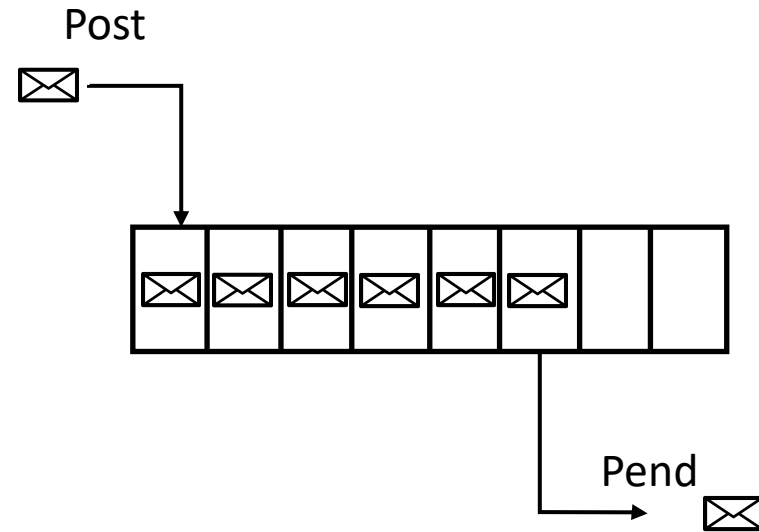


RTOS: Queue example



Message queue example for uC/OS-II

```
#include <stdio.h>
#include "includes.h"

/* Definition of Task Stacks */
#define TASK_STACKSIZE 2048
OS_STK task1_stk[TASK_STACKSIZE];
OS_STK task2_stk[TASK_STACKSIZE];
OS_STK task3_stk[TASK_STACKSIZE];

/* Definition of Task Priorities */
#define TASK1_PRIORITY 6
#define TASK2_PRIORITY 7
#define TASK3_PRIORITY 8

//Semaphore to protect jtag uart
OS_EVENT *shared_jtag_sem;

//Message queue OS_EVENT structure
OS_EVENT *MSG_Q;
#define QSIZE 10
//Storage for message pointers
void *queue_data[QSIZE];

typedef struct msg_data{
    int time;
    int taskid;
    int msg_no;
}MSG_DATA;

MSG_DATA msg_t1;
MSG_DATA msg_t2;
```

Message queue example for uC/OS-II

Similar for task2

```
void task1(void* pdata)
{
    INT8U error_code = OS_NO_ERR;
    int t1;
    int cnt=1;

    while (1)
    {
        t1 = OSTimeGet();
        OSSemPend(shared_jtag_sem,0,&error_code);
        printf("Task1 sending message: time=%d ms, id=%d, msg_no=%d\n",t1,1,cnt);
        OSSemPost(shared_jtag_sem);
        msg_t1.time = t1;
        msg_t1.taskid = 1;
        msg_t1.msg_no = cnt;
        error_code = OSQPost(MSG_Q,(void *)&msg_t1);
        cnt=cnt+1;
        OSTimeDlyHMSM(0, 0, 1, 0);
    }
}
```

Message queue example for uC/OS-II

```
void task3(void* pdata)
{
    INT8U error_code = OS_NO_ERR;
    int t1;
    MSG_DATA *msg_rx;
    MSG_DATA msg_local;
```

```
    while (1)
    {
        msg_rx = (MSG_DATA*)OSQPend(MSG_Q,0,&error_code);
        msg_local = *msg_rx;
        t1 = OSTimeGet();
        OSSemPend(shared_jtag_sem,0,&error_code);
        printf("Task3 received message: time=%d ms, id=%d, msg_no=%d (at %d ms)\n",
            msg_local.time,msg_local.taskid,msg_local.msg_no,t1);
        OSSemPost(shared_jtag_sem);
    }
}
```

```
Task1 sending message: time=1071 ms, id=1, msg_no=2
Task2 sending message: time=1075 ms, id=2, msg_no=2
Task3 received message: time=1071 ms, id=1, msg_no=2 (at 1080 ms)
Task3 received message: time=1075 ms, id=2, msg_no=2 (at 1082 ms)
Task1 sending message: time=2075 ms, id=1, msg_no=3
Task2 sending message: time=2079 ms, id=2, msg_no=3
Task3 received message: time=2075 ms, id=1, msg_no=3 (at 2084 ms)
Task3 received message: time=2079 ms, id=2, msg_no=3 (at 2086 ms)
```