

```

#include <limits.h>
#include <assert.h>

#include "task.h"
#include "processor.h"
#include "my_scheduler.h"
#include "my_sem.h"

#define MAX_SEMS 1024
#define MAX_WAITS (MAX_MY_TASKS)
#define INVALID_SEM_COUNT INT_MAX

typedef struct _T_WaitItem {
    SemId    sem;
    TaskId   task;
} WaitItem;

int      _sems[MAX_SEMS];
WaitItem _waits[MAX_WAITS];

void initSem(void)
{
    int i;

    for ( i = 0; i < MAX_WAITS; ++i )
        _waits[i].sem = INVALID_SEM_ID;

    for ( i = 0; i < MAX_SEMS; ++i )
        _sems[i] = INVALID_SEM_COUNT;
}

SemId createSem(int count)
{
    int i;

    Tsk_lock();

    for ( i = 0; i < MAX_SEMS; ++i )
        if ( _sems[i] == INVALID_SEM_COUNT ) {
            _sems[i] = count;
            Tsk_unlock();
            return i;
        }

    Tsk_unlock();
    return INVALID_SEM_ID;
}

```