

ARM PROGRAMMING

Bùi Quốc Bảo

Standard malloc and free function

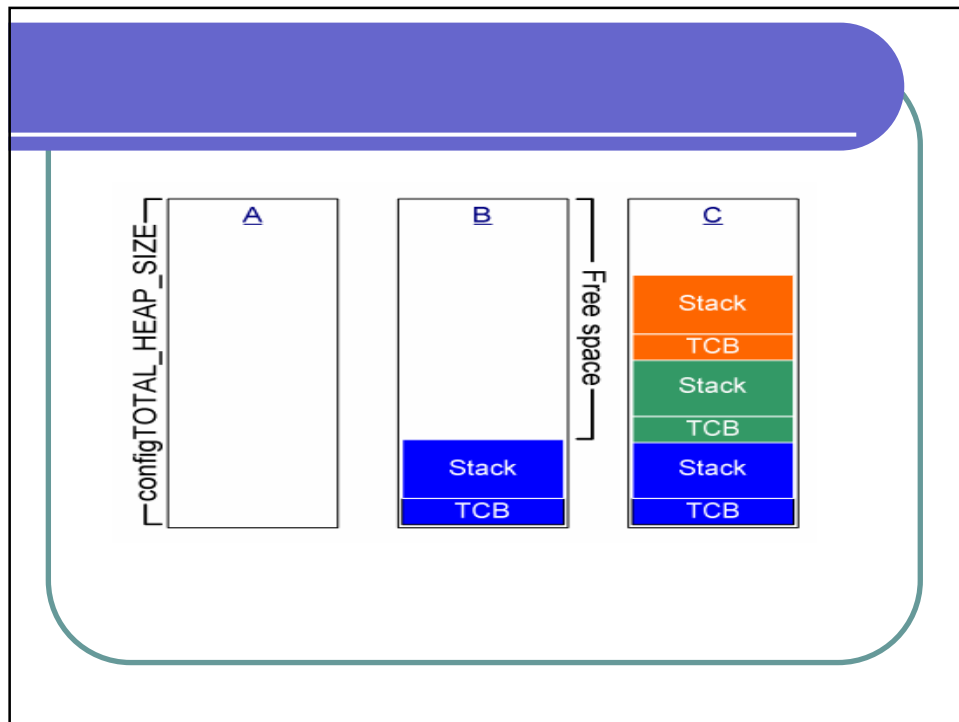
- They are not always available on small embedded systems.
- Their implementation can be relatively large so take up valuable code space.
- They are rarely thread safe.
- They are not deterministic. The amount of time taken to execute the functions will differ from call to call.
- They can suffer from memory fragmentation.
- They can complicate the linker configuration.

FreeRTOS pvPortMalloc() and vPortFree()

- FreeRTOS cung cấp 2 hàm pvPortMalloc() và vPortFree()
- User có thể viết lại hai hàm này
- Hai hàm này nằm trong các file heap_1.c, heap_2.c and heap_3.c, trong thư mục FreeRTOS\Source\Portable\MemMang

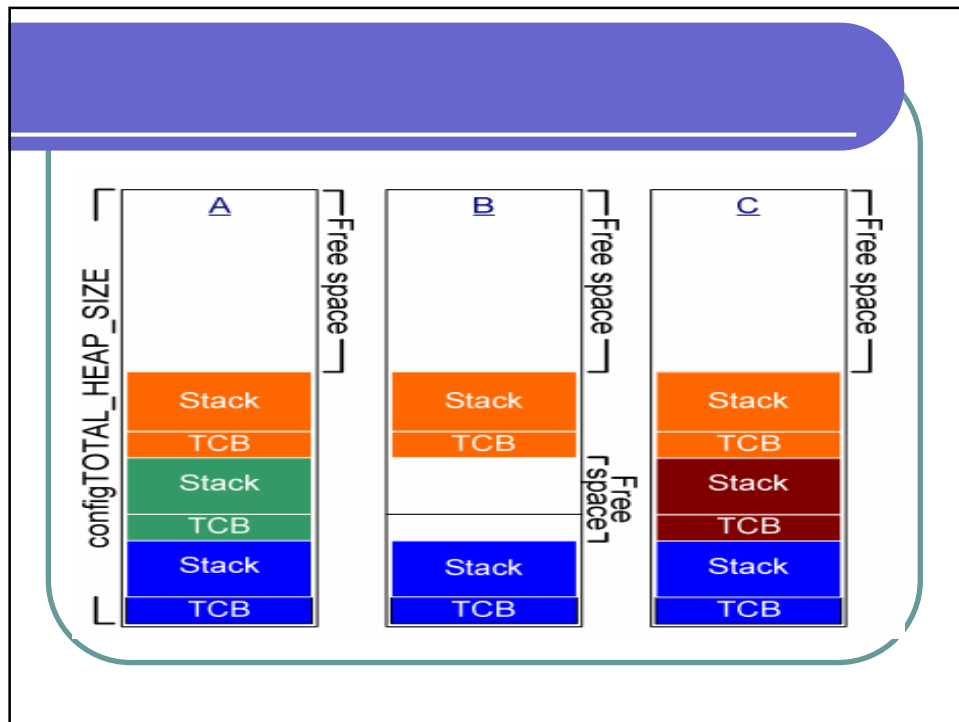
Heap_1.c

- Heap1.c cung cấp một hàm pvPortMalloc() cơ bản, không có hàm vPortFree()
- Các chương trình mà không có lệnh xóa task, semaphore, queue thì có thể dùng heap1.c



Heap2.c

- Heap1.c cung cấp hàm `pvPortMalloc()` và `pvPortFree()`



Heap3.c

- Heap3.c dùng các hàm malloc và free, đồng thời suspend scheduler

```
void *pvPortMalloc( size_t xWantedSize )
{
    void *pvReturn;

    vTaskSuspendAll();
    {
        pvReturn = malloc( xWantedSize );
    }
    xTaskResumeAll();

    return pvReturn;
}
```

```
void vPortFree( void *pv )
{
    if( pv != NULL )
    {
        vTaskSuspendAll();
        {
            free( pv );
        }
        xTaskResumeAll();
    }
}
```