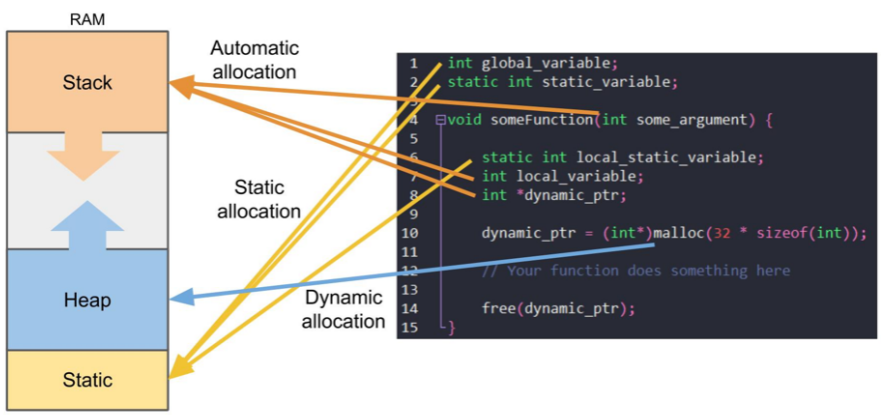
MEMORY MANAGEMENT

Volatile Memory
- static → stores global variables
- stack → automatic allocation of global variables
  - → organised as last-in-first-out (LIFO) system
- heap → allocated explicitly by programmer
  - ↳ eg. malloc() function
  - → aka dynamic allocation
  - → must manually deallocate heap for languages w/o garbage collecting system eg. C/C++
    - ↳ failure results in memory leak

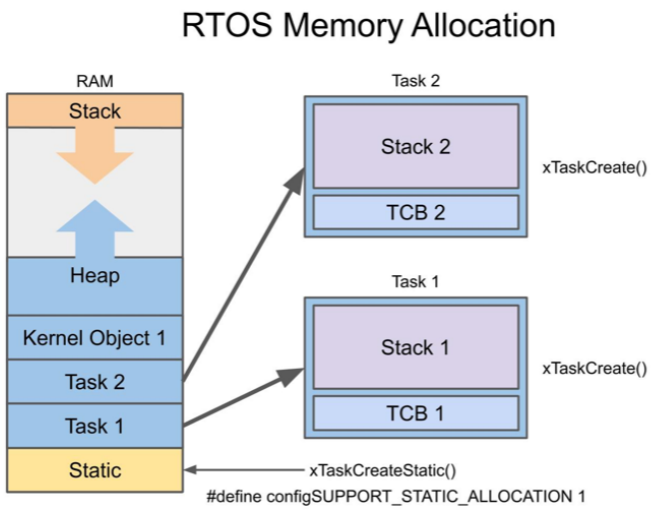
→ stack & heap grow towards each other
  ↳ can take up unallocated memory → if left unchecked, could collide & overwrite each other's data

Memory Allocation



Static memory is used for storing global variables and variables designated as

- Create task in FreeRTOS
  - ↳ the O.S. allocates a section of heap memory for the task



Task Control Block → stores info about task eg. its priority & local stack pointer (TCB)

→ local variables created during fn. calls within a task are pushed to the task's local stack

→ For FreeRTOS, malloc() & free() are not thread safe
(use pvPortMalloc() & vPortFree())