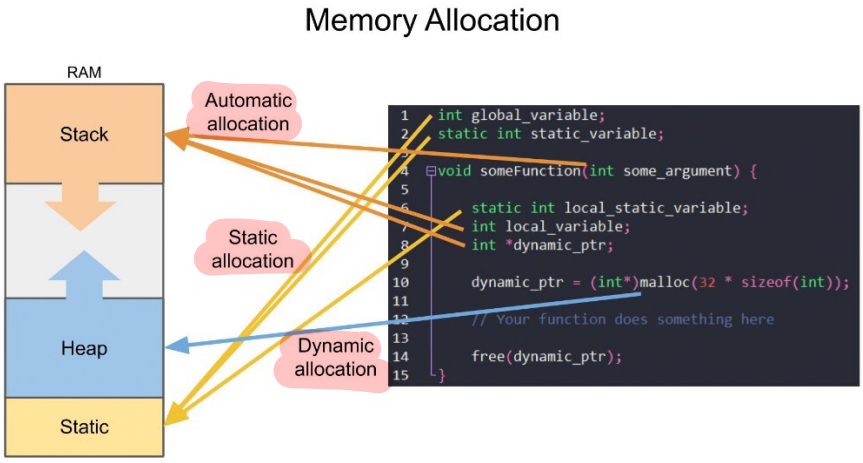
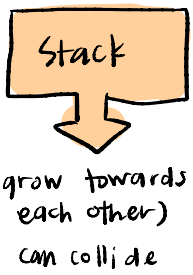
MEMORY MANAGEMENT

CONCEPTS



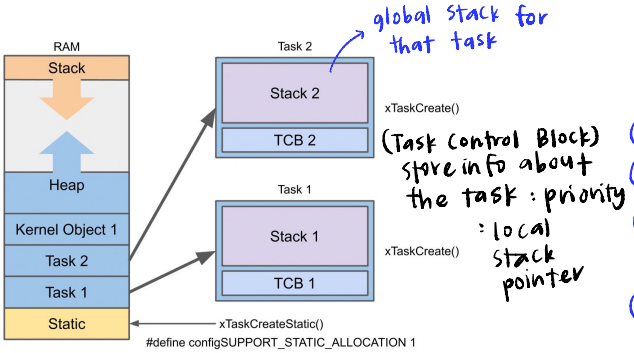


① Automatic allocation for global variables.
② last-in-first out (LIFO) system so that the variables of one function can "be pushed" to the stack when a new function is called.
③ Upon returning to the first function, that function's variable can be "popped off", which the function can use to continue running where it left.

① Must be allocated explitly by user. malloc()
② Heap memory must be deallocate when it is no longer used for system without garbage collection (C and C++)
③ If not, it can cause memory leak, undefined effects, corrupting memory.

① Used for storing global variables and variables designed as "static" (between function calls).

RTOS MEMORY ALLOCATION



① Local variables created during fx calls are pushed to the task's local stack.
② The stack size parameter included in xTaskCreate().
③ malloc() and free() are not thread safe in FreeRTOS.
④ Use pvPortMalloc() and vPortFree() to allocate the system's global heap
⑤ Static tasks use only static memory (allocating their own local stack and TCB in static memory instead of the heap.

↓
useful when heap memory is not used to prevent heap flow.