

# Resumen Memory Api - SistOp

Lautaro Bachmann

## Contents

<b>Interlude: Memory API</b>	<b>3</b>
Types of Memory . . . . .	3
stack memory, . . . . .	3
heap memory, . . . . .	3
Common Errors . . . . .	3
Underlying OS Support . . . . .	3
malloc() and free() aren't system calls, . . . . .	3
brk system call . . . . .	3
sbrk . . . . .	4
mmap() call. . . . .	4

## Interlude: Memory API

### Types of Memory

#### **stack memory,**

allocations and deallocations of it are managed implicitly by the compiler for you, for this reason it is sometimes called **automatic** memory.

#### **heap memory,**

all allocations and deallocations are explicitly handled by you,

#### **A couple of notes**

both stack and heap allocation occur when using the heap

first the compiler knows to make room for a pointer when it sees your declaration of said pointer. Subsequently, when the program calls `malloc()`, it requests space on the heap;

heap memory presents more challenges to both users and systems.

### Common Errors

Forgetting To Allocate Memory

Not Allocating Enough Memory

Forgetting to Initialize Allocated Memory

Forgetting To Free Memory

Freeing Memory Before You Are Done With It

Freeing Memory Repeatedly

Calling `free()` Incorrectly

### Underlying OS Support

#### **`malloc()` and `free()` aren't system calls,**

but rather library calls. Thus the `malloc` library manages space within your virtual address space, but itself is built on top of some system calls which call into the OS to ask for more memory or release some back to the system.

#### **`brk` system call**

is used to change the location of the program's **break**: the location of the end of the heap.

It increases or decreases the size of the heap based on whether the new break is larger or smaller than the current break.

**sbrk**

is passed an increment but otherwise serves a similar purpose as brk.

**mmap() call.**

mmap() can create an anonymous memory region within your program  
This memory can then also be treated like a heap and managed as such.