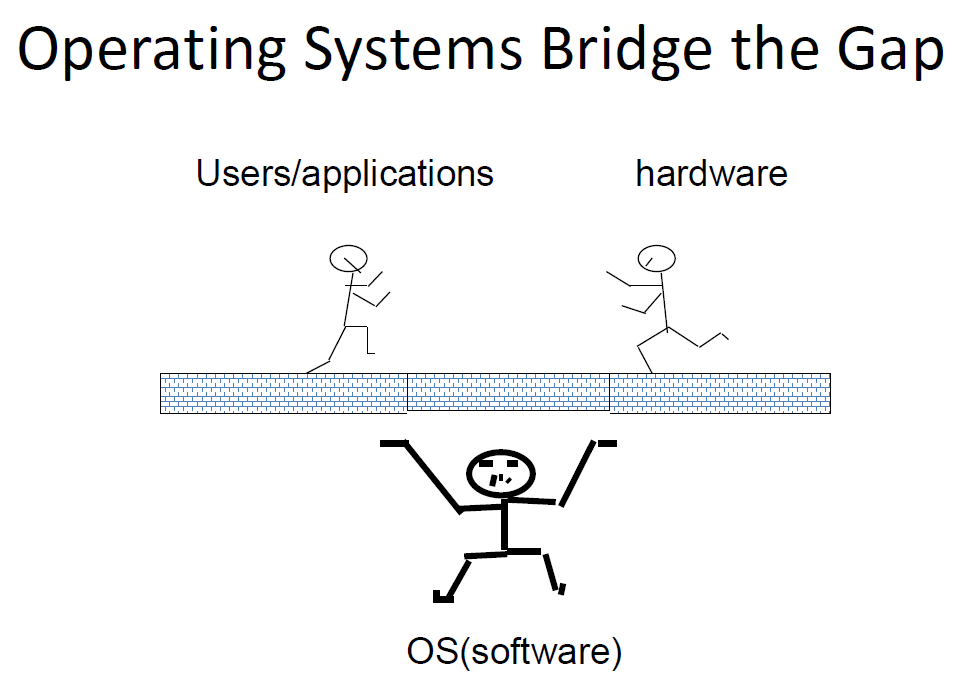
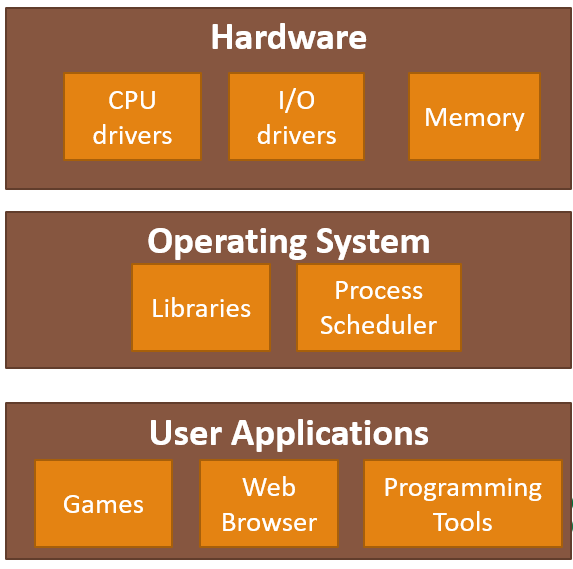
| **GDPARCM Lecture – Introduction to Operating Systems** | Instructor: Neil Patrick Del Gallego |
| --- | --- |

**Background on Operating Systems**





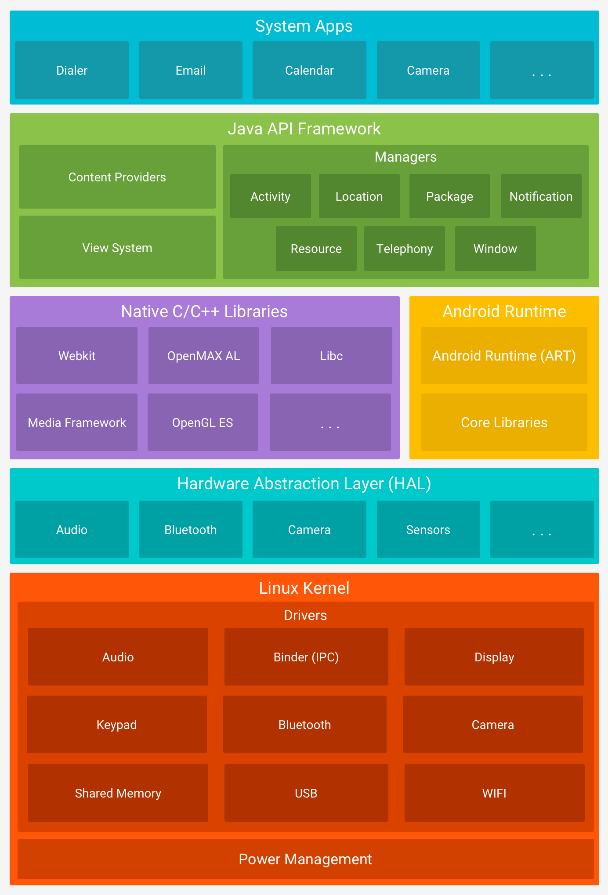
A computer system can be summarized into four components:

* Hardware components – physical components and drivers.
* OS – coordinates use of hardware with user applications
* Applications – typical applications used by a computer user

The primary tasks of an OS:

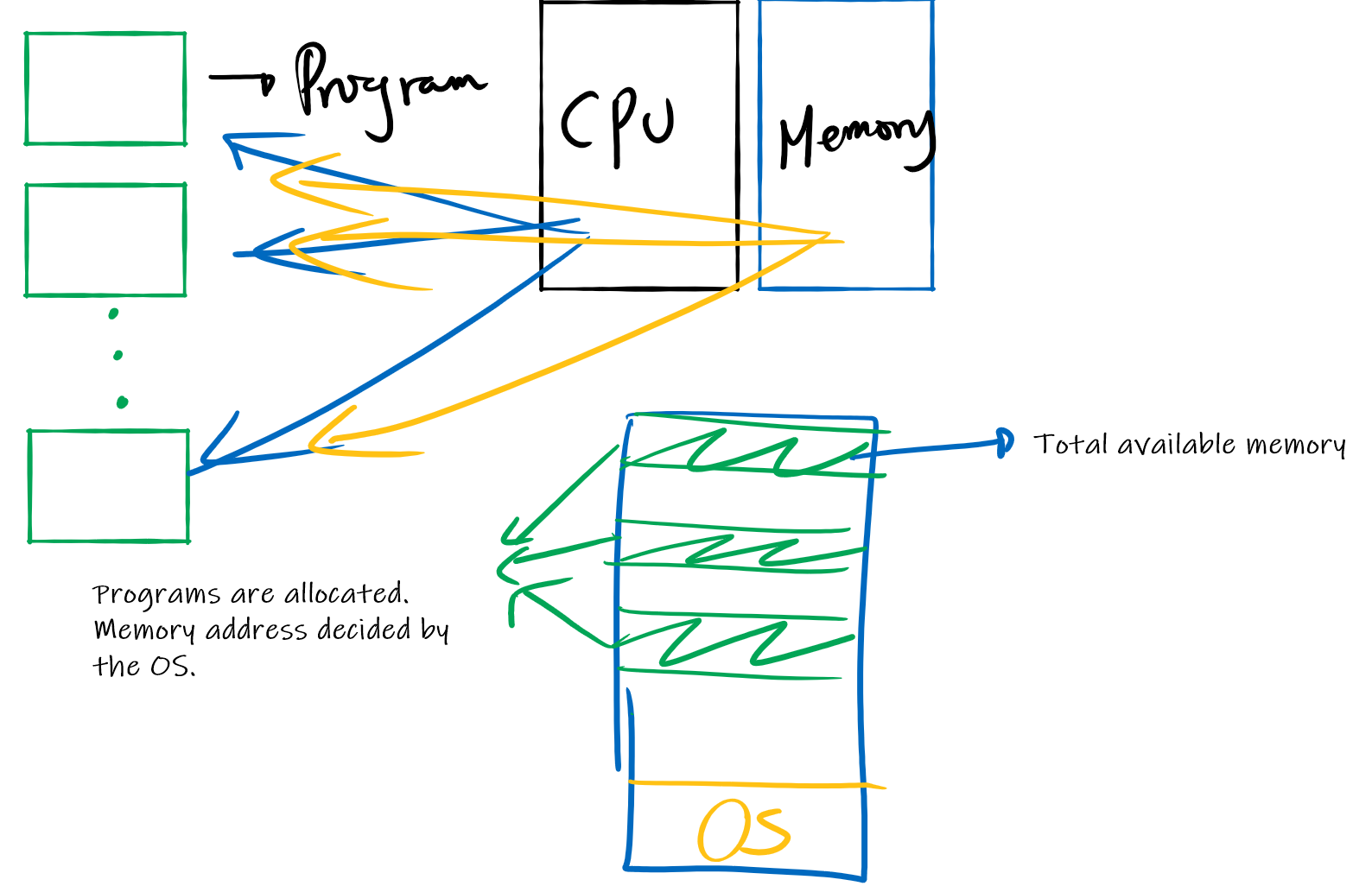
* Resource allocator – allocates resources such as CPU and memory to an application. Decides how to share resources among other user programs.
* Control program – controls behavior of user programs to avoid improper use of computer resources (CPU and memory hoggers, programs with infinite loops, etc).
* Security - Must protect application’s data from one another (unless designed to share the data). Allocate resources fairly and efficiently. Settles conflicting requests for resources. Prevent errors and improper use of the hardware
* Improvisation – find a way to improvise when hardware resources are scarce. Provides an illusion of dedicated machine with infinite memory and processors.

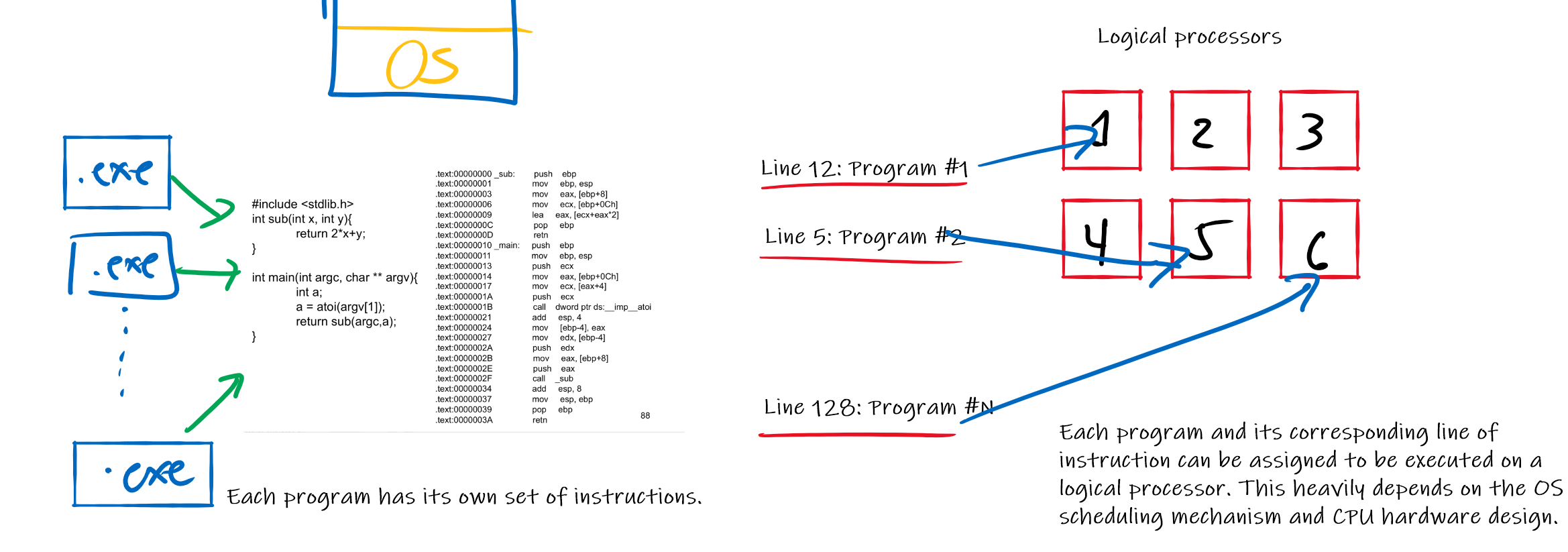
Architecture of android



An example of primary tasks in action:

**Resource allocator**

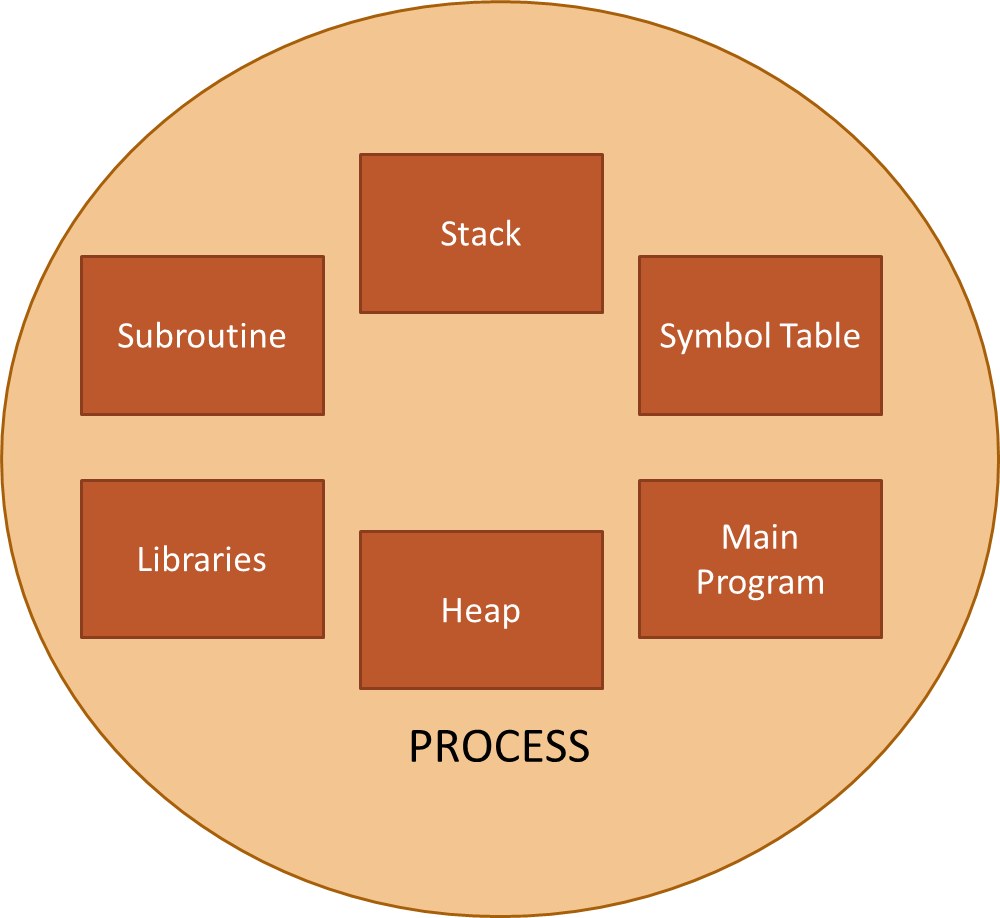




**Control program**

The OS ensures that all applications receive a fair number of resources. Each program has components where allocations are already pre-defined.

Components of a Program

* Main Program – points to the main function
* Subroutine – contains all functions declared in code
* Libraries – external functions or dependencies required
* Symbol table – holds variables
* Stack – holds temporary variables for function calls. Also stores the return address of a given function call.
* Heap – memory for dynamically allocated objects

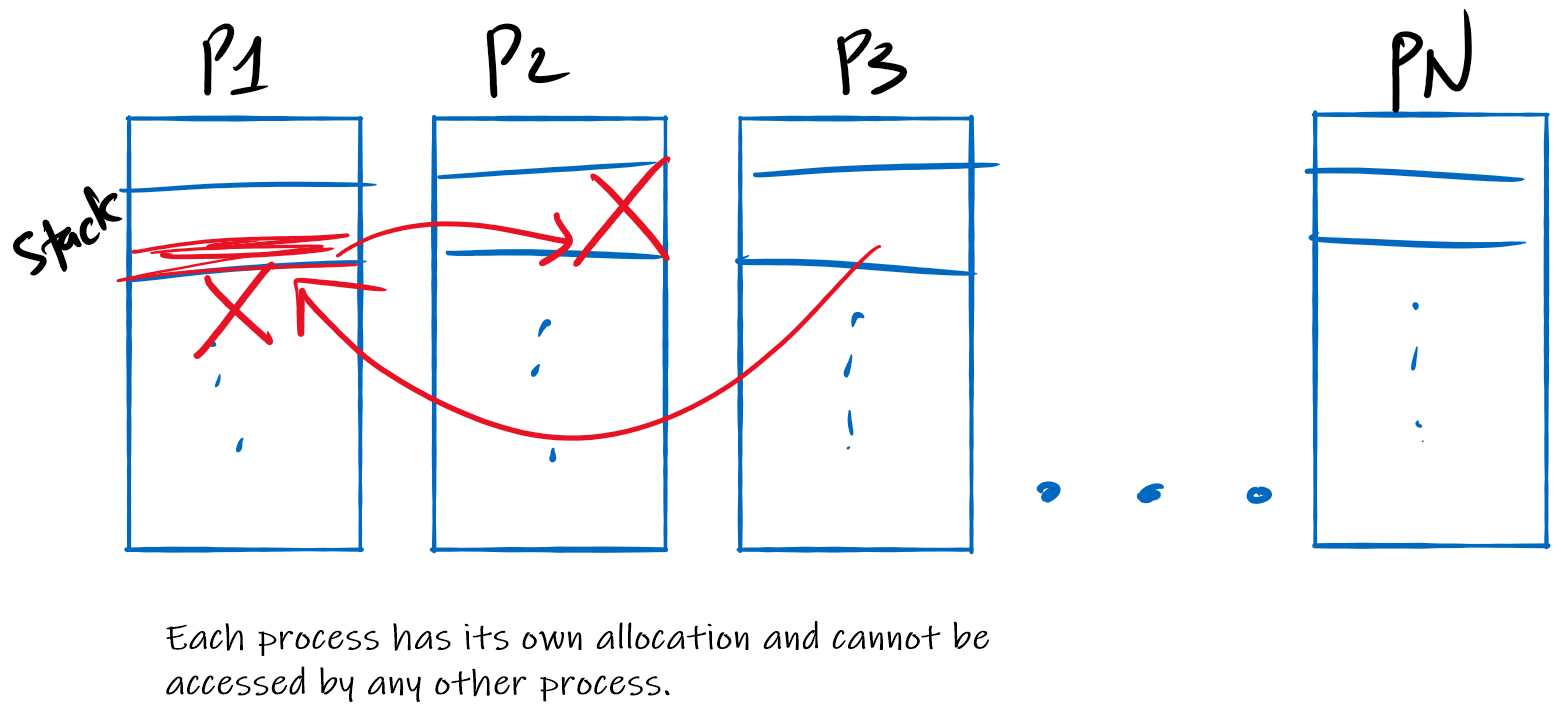
**ACTIVITY:**

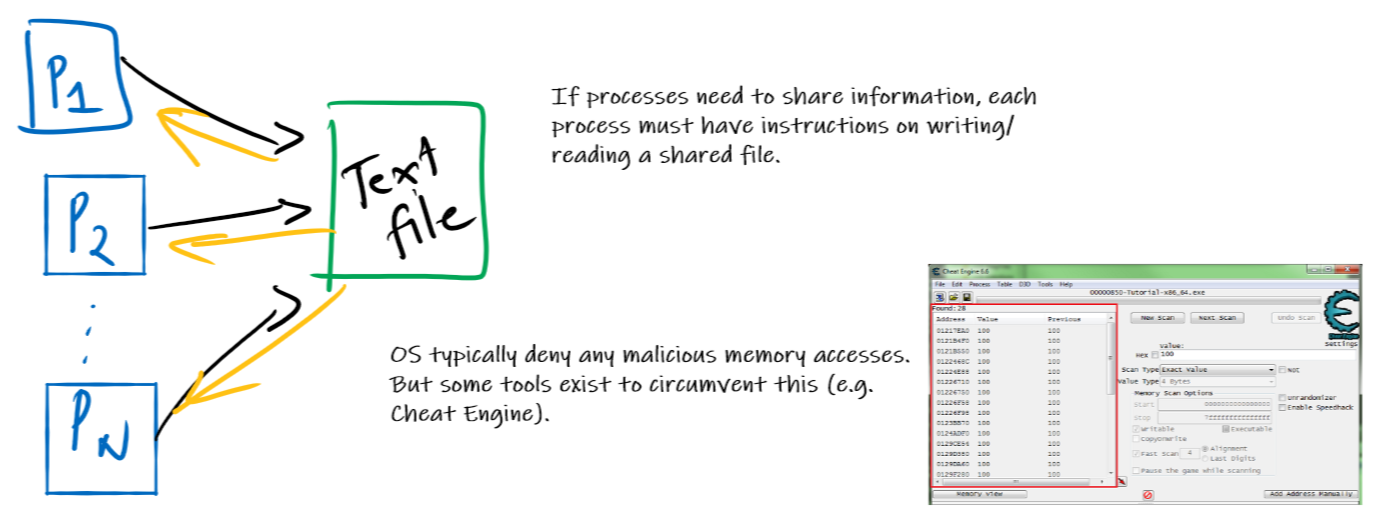
1. Provide a scenario that causes stack to overflow.
2. Provide a scenario that causes the heap to overflow.

|  |  |
| --- | --- |
|  |  |
|  |  |

The OS pre-allocates memory on each application. Each application has a certain boundary on their components. Overstepping these boundaries throw errors.

**Security**





**Improvisation**

In memory management, all OS perform a technique called **virtual memory**, where the OS uses secondary storages (such as HDD/SDD) as main memory whenever RAM is used up. This gives the illusion of a very large main memory.

