

What is a process?

Application – program on disk, flash memory ... (static entity).

Process – state of a program when executing loaded in memory (active entity)

If the same program launched more than multiple processes will be created, but will have very different state. It doesn't mean necessarily that process running (may be waiting for input or waiting for another process executing on CPU).

What does a process look like?

Process encapsulates all of this data for running applications:

- Text (program instruction set)
- Data (global and static variables)
- Heap (dynamic allocated memory)
- Stack (local variables)

All of these segments must have unique address. OS uses an abstraction called 'address space' to encapsulate all of the process state.

Address space – range of addresses from some V_0 to V_{max} . Different parts of process state (code, data...) will appear in different region of address space.

- Code and data segments exists when process first loads.
- Heap dynamically get expanded and shrinked during execution.
- Stack is a LIFO.

Process address space.

Addresses that are used by a process are *virtual* because they don't correspond to actual locations in physical memory. Instead, *memory management hardware* and *OS components* (like page tables) maintain a mapping between virtual addresses and physical addresses.