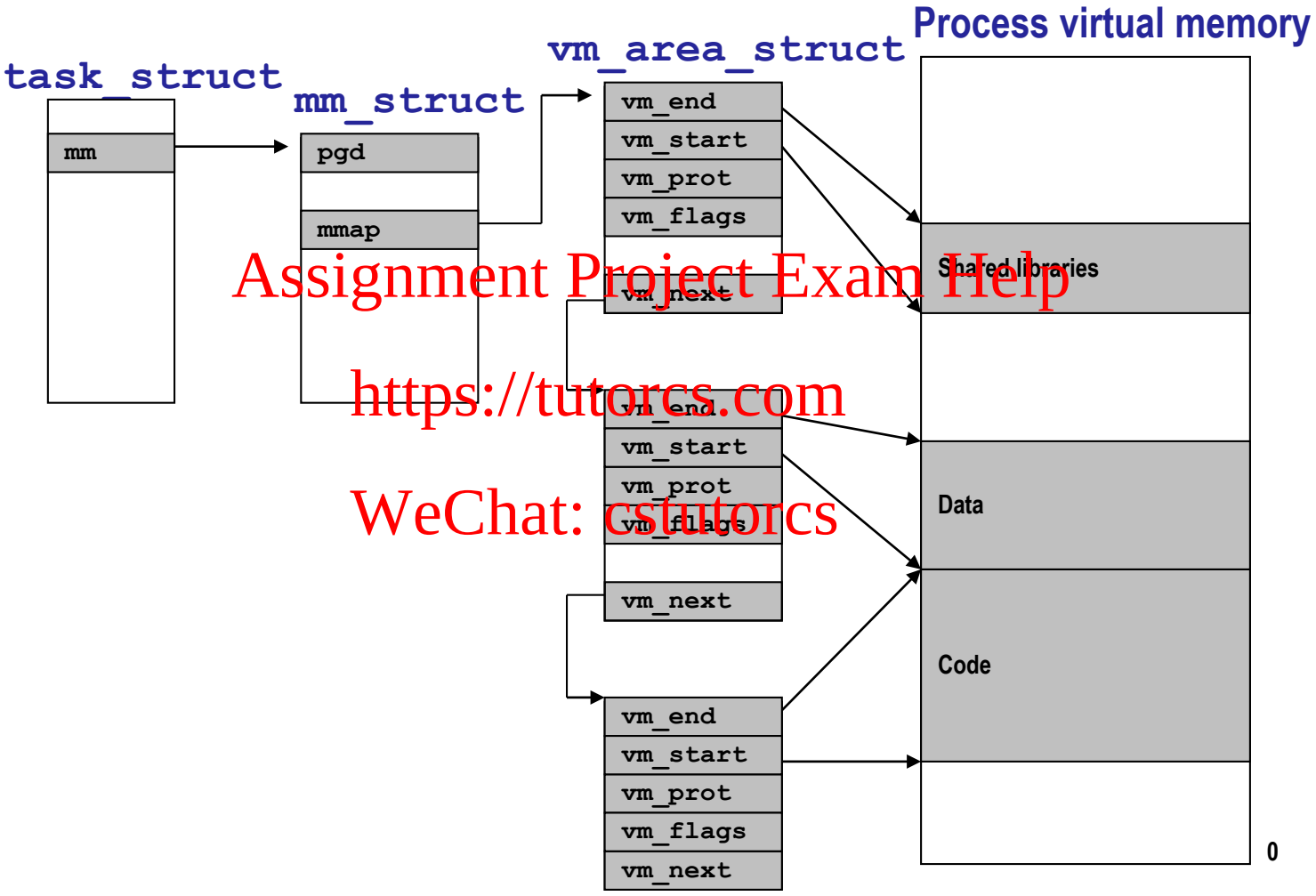
Key VM Structures in Linux

- **`vm_area_struct:`** describes the properties of a contiguous region of the virtual address space
  - Entire region may contain many pages
  - Entire region shares properties' permissions and associated operations
  - May be a file, library, group of libraries, etc.
- **`mm_struct:`** contains information about the process' entire virtual address space
  - Contains a list of `vm_area_struct` objects
  - Processes have distinct `mm_structs`, threads share an `mm_struct`

VM Structures in Linux



The `fork` Function Revisited

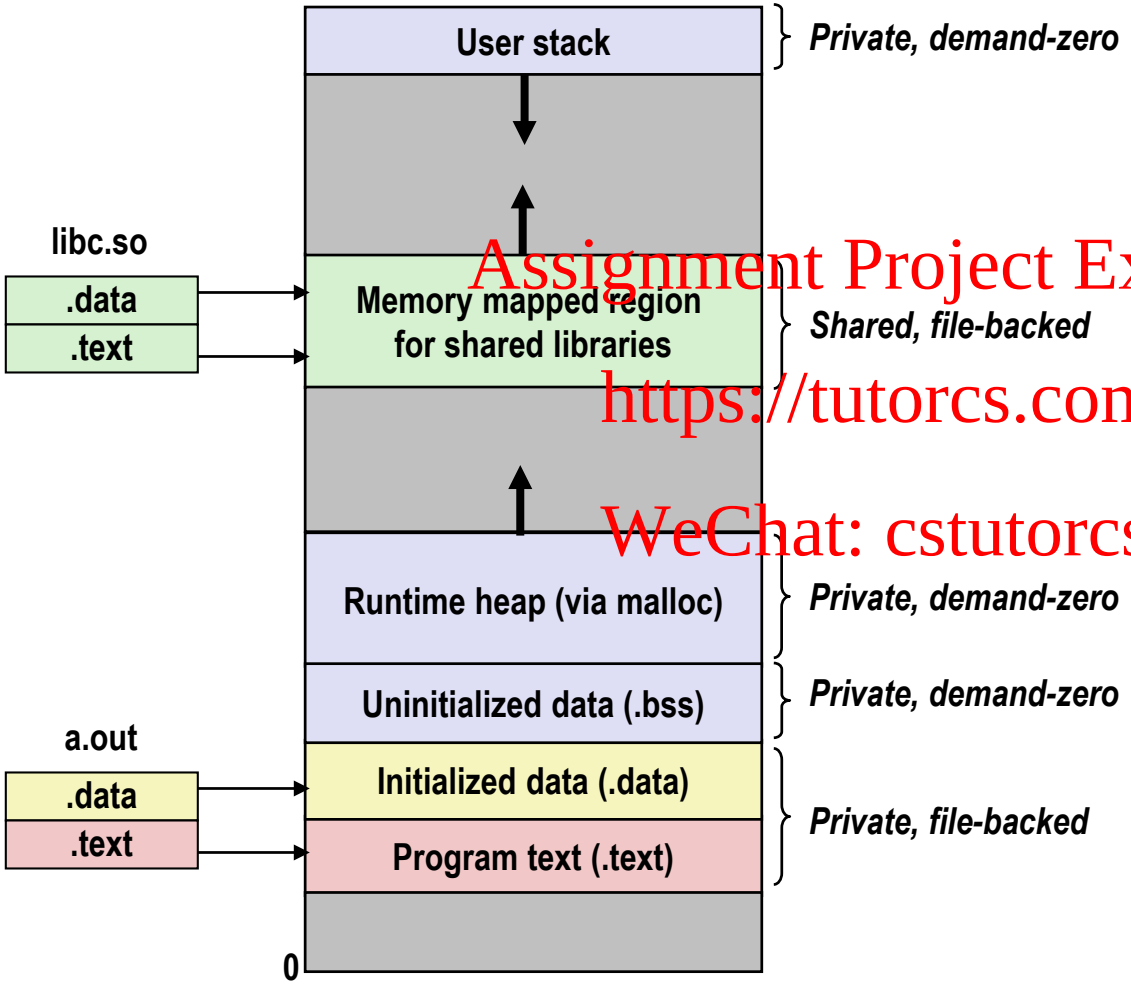
- **VM and memory mapping explain how `fork` provides private address space for each process.**

- **To create virtual address for new new process**
  - Create exact copies of current `mm_struct`, `vm_area_struct`, and page tables.
  - Flag each page in both processes as read-only
  - Flag each `vm_area_struct` in both processes as private COW

- **On return, each process has exact copy of virtual memory**

- **Subsequent writes create new pages using COW mechanism.**

The execve Function Revisited



- To load and run a new program `a.out` in the current process using `execve`:
- Free `vm_area_struct`'s and page tables for old areas
- Create `vm_area_struct`'s and page tables for new areas
  - Programs and initialized data backed by object files.
  - `.bss` and stack backed by anonymous files .
- Set PC to entry point in `.text`
  - Linux will fault in code and data pages as needed.