INFO-0940: Operating Systems

Project 2: Adding system calls - Report

Goffart Maxime   Joris Olivier
180521   182113

Academic year 2020 - 2021

1 Implementation

2 How the memory of a process is managed by the kernel ?

Processes are implemented in the kernel as instances of `task_struct`. This struct contains a *mm* field which is an instance of `mm_struct` that represents a summary of the process memory. This *mm* field contains a *mmap* that is an instance of `vm_area_struct` that represents a memory area. This *nmap* field contains 3 fields *vm_ start*, *vm_ end*, and *vm_ next*. The first one represents the logical address corresponding to the first address within the virtual memory area, the second one represents the first address outside the virtual memory area, and the third one is a reference to another `vm_area_struct` which contains higher segment begin and end addresses.

To translate a logical address to a physical address, a page table is used. This table allows to obtain the physical address corresponding to a given logical given address using multi-level paging in order to reduce the size of the table that is stored in physical memory. Moreover, the offset of the logical address is retained according to the page size and only the first bits of the logical address are used to look at the mapping in the table in order to reduce the size of the entries in the table. Some entries do not refer to any physical page : this means that they have the present flag clear. This could be because their contents have been swapped out or because they never have been touched.

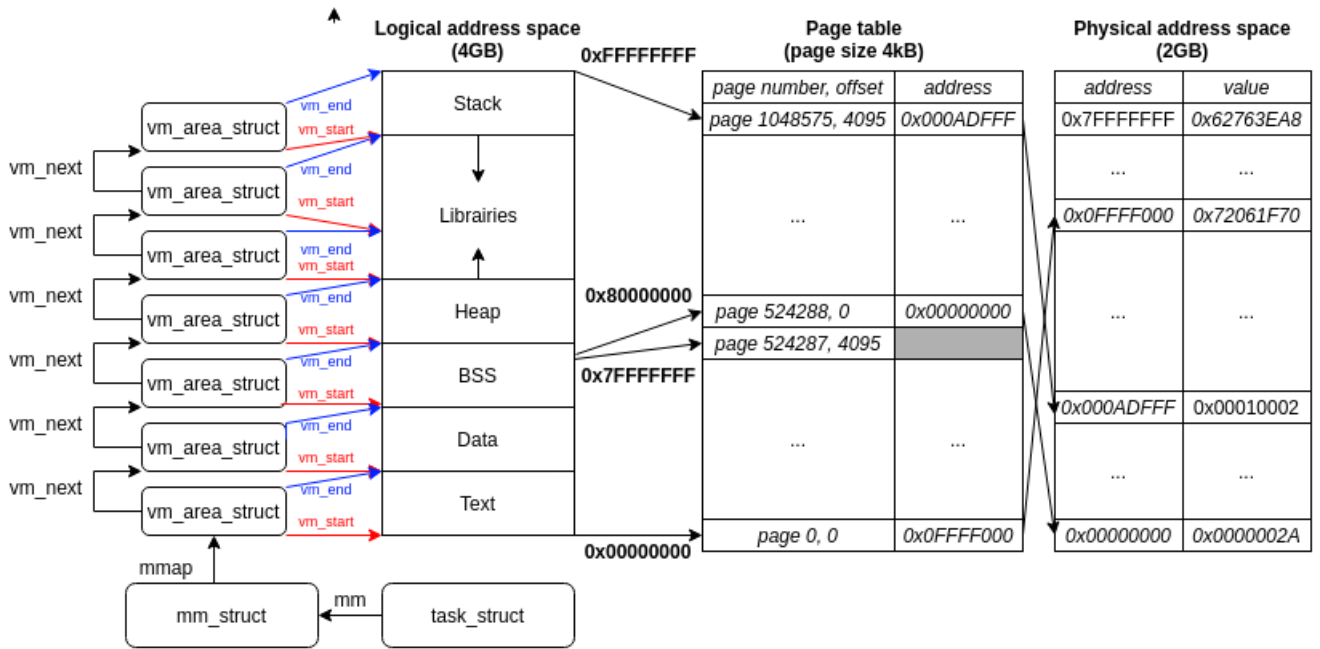This management of the process memory by the kernel is represented on the Figure 1.

Figure 1: Management of the process memory by the kernel.