

Abstract: This thesis deals with virtual memory management in the Linux kernel. Files or devices can be mapped into virtual memory using the `mmap()` system call, which is also capable of creating so-called anonymous mappings. Those do not map anything and serve only as a memory allocation method that can specify memory protection and flags for the created anonymous virtual memory area (VMA). A mapping can be merged or split depending on its flags, memory protection, location in the virtual memory and other parameters. However, in some cases even when two VMAs have practically identical parameters, they may not merge successfully because of implementation limitations. This thesis concentrates on anonymous VMAs and their page offset and `anon_vma` parameters, which in some cases prevent a merge in the current mainline kernel. It is demonstrated that in most of the cases with reasonable amount of effort the obstacles can be removed and the merges happen.