remap_file_pages() system call

The remap_file_pages() system call is used to create a nonlinear mapping, that is, a mapping in which the pages of the file are mapped into a nonsequential order in memory. The advantage of using remap_file_pages() over using repeated calls to mmap(2) is that the former approach does not require the kernel to create additional VMA (Virtual Memory Area) data structures.

Supporting of nonlinear mapping requires significant amount of non-trivial code in kernel virtual memory subsystem including hot paths. Also to get nonlinear mapping work kernel need a way to distinguish normal page table entries from entries with file offset (pte_file). Kernel reserves flag in PTE for this purpose. PTE flags are scarce resource especially on some CPU architectures. It would be nice to free up the flag for other usage.

Fortunately, there are not many users of remap_file_pages() in the wild. It's only known that one enterprise RDBMS implementation uses the syscall on 32-bit systems to map files bigger than can linearly fit into 32-bit virtual address space. This use-case is not critical anymore since 64-bit systems are widely available.

The syscall is deprecated and replaced it with an emulation now. The emulation creates new VMAs instead of nonlinear mappings. It's going to work slower for rare users of remap_file_pages() but ABI is preserved.

One side effect of emulation (apart from performance) is that user can hit vm.max_map_count limit more easily due to additional VMAs. See comment for DEFAULT_MAX_MAP_COUNT for more details on the limit.