CLAIMS

1. A method implemented in a computer comprising a processor, memory accessible by the processor, and computer program instructions stored in the memory and executable by the processor, the method comprising:

when a user process makes a system call, switching to kernel mode and using a system call page table for the user process to execute a system call handler;

when the system call handler attempts to access unmapped kernel space memory, generating a page fault; and

handling the page fault by determining whether the attempted access to unmapped kernel space memory is allowed;

wherein the system call page table comprises a plurality of kernel-space address ranges allowing execution of kernel code and access to kernel data, and wherein the system call page table is created by:

creating a template system call page table comprising mappings of kernel code and data required to perform the entry to the kernel mode upon execution of a system call, a table with the pointers to a system call handlers, and an entry code of each system call handler;

creating a copy of the template system call page table for the user process in a process state representation when the user process is created and populating the per-process entry page table with user-space mappings along with the original template; and

when a page fault is determined to be allowed, adding a mapping of a kernel-space address range allowing execution of the system call handler that attempted to access kernel-space memory.

2. The method of claim 1, wherein the computer system operating system is the Linux operating system.

3. The method of claim 2, further comprising: when a page fault is determined not to be allowed, terminating execution of the user process that made the system call that attempted to access kernel-space memory.

4. The method of claim 3, further comprising: when execution of the system call handler ends, removing the mapping of the kernel-space address range added during execution of the system call handler.

5. A system comprising a processor, memory accessible by the processor, and computer program instructions stored in the memory and executable by the processor to perform:

when a user process makes a system call, switching to kernel mode and using a system call page table for the user process to execute a system call handler;

when the system call handler attempts to access unmapped kernel space memory, generating a page fault; and

handling the page fault by determining whether the attempted access to unmapped kernel space memory is allowed;

wherein the system call page table comprises a plurality of kernel-space address ranges allowing execution of kernel code and access to kernel data, and wherein the system call page table is created by:

creating a template system call page table comprising mappings of kernel code and data required to perform the entry to the kernel mode upon execution of a system call, a table with the pointers to a system call handlers, and an entry code of each system call handler;

creating a copy of the template system call page table for the user process in a process state representation when the user process is created and populating the per-process entry page table with user-space mappings along with the original template; and

when a page fault is determined to be allowed, adding a mapping of a kernel-space address range allowing execution of the system call handler that attempted to access kernel-space memory.

6. The system of claim 5, wherein the computer system operating system is the Linux operating system.

7. The system of claim 6, further comprising: when a page fault is determined not to be allowed, terminating execution of the user process that made the system call that attempted to access kernel-space memory.

8. The system of claim 7, further comprising: when execution of the system call handler ends, removing the mapping of the kernel-space address range added during execution of the system call handler.

9. A computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith, the program instructions executable by a computer, to cause the computer to perform a method comprising:

when a user process makes a system call, switching to kernel mode and using a system call page table for the user process to execute a system call handler;

when the system call handler attempts to access unmapped kernel space memory, generating a page fault; and

handling the page fault by determining whether the attempted access to unmapped kernel space memory is allowed;

wherein the system call page table comprises a plurality of kernel-space address ranges allowing execution of kernel code and access to kernel data, and wherein the system call page table is created by: creating a template system call page table comprising mappings of kernel code and data

required to perform the entry to the kernel mode upon execution of a system call, a table with the pointers to a system call handlers, and an entry code of each system call handler;

creating a copy of the template system call page table for the user process in a process state representation when the user process is created and populating the per-process entry page table with user-space mappings along with the original template; and

when a page fault is determined to be allowed, adding a mapping of a kernel-space address range allowing execution of the system call handler that attempted to access kernel-space memory.

10. The computer program product of claim 9, wherein the computer system operating system is the Linux operating system.

11. The computer program product of claim 10, further comprising: when a page fault is determined not to be allowed, terminating execution of the user process that made the system call that attempted to access kernel-space memory.

12. The computer program product of claim 11, further comprising: when execution of the system call handler ends, removing the mapping of the kernel-space address range added during execution of the system call handler.