CLAIMS

1. A computer implemented method for blocking specific instructions from being executed by a processor the method comprising:

setting an instruction blocking value;

fetching an instruction to be executed by the processor, the instruction comprising an opcode, the instruction supported by the processor;

responsive to the instruction blocking value permitting execution of the instruction, executing the fetched instruction by the processor; and

responsive to the instruction blocking value not permitting execution of the instruction, blocking execution of the fetched instruction and causing a program exception event.

2. The method according to Claim 1, wherein the processor is a logical processor of a virtual machine, wherein the fetching is performed by the logical processor, the method further comprising:

determining the instruction blocking value of the virtual machine, wherein the instruction blocking value is set in the logical processor having one or more physical processors, wherein the instruction is supported by the one or more physical processors, wherein responsive to the instruction blocking value permitting execution of the instruction, the execution is performed by the logical processor.

3. The method according to Claim 1, wherein the processor is one or more physical processors of a logical processor of a virtual machine, wherein the instruction blocking value is set in the one or more physical processors, wherein the fetching is performed by the one or more physical processors.

4. The method according to Claim 2, further comprising:

the instruction blocking value having been defined for the virtual machine for blocking execution of the instruction, the setting the instruction blocking value responsive to the enabling the virtual machine to use the physical processor;

setting another instruction blocking value defined for another virtual machine having another logical processor, the setting the another instruction blocking value responsive to the enabling the another virtual machine to use the physical processor; and

responsive to the another instruction blocking value permitting execution of the instruction, permitting execution of the instruction by the another logical processor; and

responsive to the another instruction blocking value not permitting execution of the instruction, not permitting execution of the instruction by the another logical processor.

5. The method according to Claim 3, further comprising:

the instruction blocking value having been defined for the virtual machine for blocking execution of the instruction, the setting the instruction blocking value responsive to the enabling the virtual machine to use the physical processor;

setting another instruction blocking value defined for another virtual machine having another logical processor, the setting the another instruction blocking value responsive to the enabling the another virtual machine to use the physical processor; and

responsive to the another instruction blocking value permitting execution of the instruction, permitting execution of the instruction by the physical processor while the another virtual machine is enabled to use the physical processor; and

responsive to the another instruction blocking value not permitting execution of the instruction, not permitting execution of the instruction by the physical processor while the another virtual machine is enabled to use the physical processor.

6. The method according to Claim 1, wherein the instruction is the permitted instruction responsive to the instruction employing a permitted function code, wherein the instruction is the not permitted instruction responsive to the instruction employing a not permitted function code.

7. The method according to Claim 2, further comprising:

determining whether the instruction is the permitted instruction by associating the opcode of the instruction with the instruction blocking value.

8. The method according to Claim 6, wherein the instruction fetched specifies a function to be performed, further comprising:

using the opcode to index into a table to locate the instruction blocking value, the instruction blocking value comprising a permission field;

using the permission field to determine permitted functions; and

responsive to the function being a permitted function, determining the execution of the instruction is permitted; and

responsive to the function being a not permitted function, determining the execution of the instruction is not permitted.

9. A computer system for blocking specific instructions from being executed by a processor, comprising:

a memory;

a processor in communication with the memory, the processor comprising an instruction fetching element for fetching instructions from memory and one or more execution elements for executing fetched instructions;

wherein the computer system is configured to perform the method as claimed in any of claims 1 to 8.

10. A computer program product method for blocking specific instructions from being executed by a processor, the computer program product comprising a tangible storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing the method as claimed in any of claims 1 to 8.

CLAIMS

1. A computer implemented method for blocking specific functions to be performed by an instruction from being executed by a logical processor executing a virtual machine, the method comprising:

setting an instruction blocking value defined for the virtual machine, for blocking execution of an instruction, the instruction blocking value indicating permitted functions;

fetching, by said logical processor, an instruction to be executed by the logical processor, the instruction comprising an opcode and specifying a function code of a plurality of function codes, the function code being used to select the function to be performed by the instruction, the instruction being supported by one or more physical processors;

determining an instruction blocking value for the fetched instruction having the function code, by comparing the instruction blocking value defined for the virtual machine with the opcode of the fetched instruction;

responsive to the instruction blocking value permitting execution of the instruction having the function code, executing the fetched instruction by the logical processor; and

responsive to the instruction blocking value not permitting execution of the instruction having the function code, blocking execution of the fetched instruction and causing a program exception event.

2. The method according to Claim 1, further comprising:

the instruction blocking value having been defined for the virtual machine for blocking execution of functions to be performed by the instruction, the setting the instruction blocking value responsive to the enabling the virtual machine to run on the physical processor;

setting another instruction blocking value defined for another virtual machine executing on another logical processor, the setting the another instruction blocking value responsive to the enabling the another virtual machine to run on the physical processor; and

responsive to the another instruction blocking value permitting execution of the instruction having the function code, permitting execution of the instruction by the another logical processor; and

responsive to the another instruction blocking value not permitting execution of the instruction having the function code, blocking execution of the instruction by the another logical processor.

3. The method according to Claim 1, wherein the determining an instruction blocking value further comprises the steps of:

using the opcode to index into a table to locate the instruction blocking value defined for the virtual machine, the instruction blocking value comprising a permission field;

using the permission field to determine permitted functions by comparing with the function code of the fetched instruction; and

responsive to the function being a permitted function, determining that execution of the instruction is permitted; and

responsive to the function being a not permitted function, determining that execution of the instruction is not permitted.

4. The method according to Claim 1, further comprising the steps of:

determining whether the instruction is the permitted instruction by associating the opcode of the instruction with the instruction blocking value.

5. A computer system for blocking specific instructions from being executed by a processor, comprising:

a memory;

a processor in communication with the memory, the processor comprising an instruction fetching element for fetching instructions from memory and one or more execution elements for executing fetched instructions;

wherein the computer system is configured to perform the method as claimed in any of claims 1 to 4.

6. A computer program loadable into the memory of the computer system of claim 5, comprising software code portions for performing, when said program is run on the computer system, the steps of the method as claimed in claims 1 to 4.