operating system, an external device exclusively used by the load operating system, and the physical resource set further includes an external device of the host machine. The external device may be a network adapter or a magnetic disk, or the like.

[0072] A specific embodiment is used as an example. Still refer to FIG. 1. In FIG. 1, a black area in the physical resource pool is a physical resource that is being used, and a physical resource, other than the black area, is an available physical resource. There are two load operating systems, which are referred to as a first load operating system and a second load operating system. The management operating system performs the first-time division on the available physical resources, and obtains two physical resource sets. In a specific implementation process, the management operating system performs physical division on the physical resources, and the division may be performed in the following manner:

[0073] Processor core: A processor core is used as a minimum functional unit for the division, or a processor may be used as a unit. For a heterogeneous platform, a heterogeneous processor core and a coprocessor core may also be classified as global computing resources.

[0074] Physical memory: A continuous or discontinuous segment of a physical memory is used as a unit for the division, and a specific size of a memory block may be a size of a signal block or a size of a hybrid block.

[0075] With respect to an external device, the division may mainly be performed in the following manner:

[0076] Network: A physical network adapter is used as a unit for the division. When a physical network adapter is not sufficient, a physical network adapter having a multi-array function or supporting server virtualization (for example, Single-Root I/O Virtualization, SR-IOV for short) is used, and is divided according to an array or a virtual function (Virtual Function, VF for short). Only when neither a physical network adapter nor an independent physical function of the network adapter is sufficient, it may be considered to share a physical network adapter and provide multiple virtual network adapters for an operating system instance, in a manner of virtualizing front and back ends of the network adapter.

[0077] Magnetic disk: A physical magnetic disk is used as a unit for the division. When a physical magnetic disk is not sufficient, a magnetic disk that may support an SR-IOV technology in the future may be divided by using a disk virtualization function. A single

physical magnetic disk is shared in a manner of logically partitioning the magnetic disk, and finally technologies such a network file system (Network File System, NFS for short) and a random access memory disk (RAM) may be used to replace a function of a conventional disk file system.

[0078] Terminal: Because using of a terminal device (a monitor or a keyboard), a serial port, or the like consumes no resource, in a premise that system performance is not affected, multiple sets of device resources such as serial ports and terminals may be obtained by virtualization in a hardware abstraction layer virtualization manner, to meet an upper-layer need.

[0079] With respect to the resource division, in this embodiment, it is ensured that the processor cores and the physical memory are divided in physical space, and a physical unit division manner is also used on a peripheral, but an alternative solution in which time division multiplexing is performed when physical division is inadequate is also supported.

[0080] The management operating system constructs, according to types of an operating system instance running on the first load operating system and an operating system instance running on the second load operating system, correspondences between the physical resource sets and the load operating systems. When the management operating system needs, according to the type of the operating system instance running on the first load operating system, more physical resources, the management operating system allocates, to the first load operating system, a physical resource corresponding to a shadow area; correspondingly, the management operating system allocates, to the second load operating system, a physical resource corresponding to a white area.

[0081] After the management operating system allocates, to each load operating system, a physical resource exclusively used by each load operating system, the management operating system pre-constructs, for each load operating system, a startup mirror used to start up each load operating system. The startup mirror refers to a program that configures hardware required for core running and constructs a data structure required for the core running, for a load operating system when the hardware and the core data structure that are required for running of the load operating system have not been configured or have not been sufficiently prepared. The startup mirror is specifically implemented in a manner of a segment of

program so that a startup processor core can update a status of the startup processor core, where the program that is configured for the load operating system after the hardware and the core data structure have been prepared and that starts up each core service and loads each core module, so as to finally start up the load operating system.

[0082] Further, the management operating system further sets, for each load operating system, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing a respective startup mirror of each load operating system. Persons skilled in the art can understand that, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror and involved in this embodiment specifically refer to a mapping relationship that is from the virtual memory address in virtual memory addresses corresponding to the startup mirror to the physical memory address, and the management operating system may implement the mapping relationship in a manner of a page table.

[0083] In a specific implementation process, a startup processor core of a load operating system first acquires a mapping relationship that is from a virtual memory address to a physical memory address and that is required for executing a startup mirror and are set by the management operating system for the load operating system, and the load operating system may directly access the physical memory address according to the mapping relationship and does not need to perform address translation. After the startup mirror of the load operating system has been started up, the startup mirror is configured to start up the load operating system. In a running process, the startup mirror directly accesses the physical memory address according to the mapping relationship that is from the virtual memory address to the physical memory address and that is corresponding to the startup mirror. Persons skilled in the art can understand that, the management operating system may set, for the startup mirror, some mapping relationships that are from virtual memory addresses to physical memory addresses, or may set, for the startup mirror, all mapping relationships that are from virtual memory addresses to physical memory addresses. When the management operating system sets, for the startup mirror, some mapping relationships that are from the virtual memory addresses to the physical memory addresses, and when the startup processor core that has been allocated to the load operating system is in a process of executing startup of the startup mirror, the startup

flowchart of Embodiment 3 of a resource processing method according to the present invention. As shown in FIG. 4, the resource processing method provided in this embodiment of the present invention includes:

[0113] Step 401: The management operating system acquires available processor cores of a second load operating system.

[0114] The second load operating system is a logged-out load operating system, other than the first load operating system, in the multiple load operating systems, the available processor cores include a processor core that has been started up, and a physical resource set of the first load operating system includes the available processor cores.

[0115] Step 402: When a physical resource set of the first load operating system includes the available processor cores, the management operating system determines, in the available processor cores, the startup processor core that starts up the first load operating system.

[0116] Step 403: The management operating system presets, for the first load operating system, a daemon on the startup processor core, and presets, in the daemon, a location that is of the startup mirror corresponding to the first load operating system and that is in the physical memory address.

[0117] Step 404: The management operating system sets, for the first load operating system and in the daemon, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror.

[0118] Step 405: The management operating system sends an inter-core interrupt message to the daemon running on the startup processor core, where the location that is of the startup mirror corresponding to the first load operating system, that is in the physical memory address, and that is preset in the daemon by the management operating system instructs the startup processor core to jump to the location that is in the physical memory address and in which the startup mirror is located, so as to execute the startup mirror of the first load operating system.

[0119] The inter-core interrupt message instructs the startup processor core to read, from the daemon, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror and is pre-constructed by the management operating system for the first load operating system.

[0120] In step 401, that the management operating system acquires available processor

the physical resource set.

[0125] When the management operating system determines the startup processor core in the available processor cores, the following manners may be used for implementation: the management operating system determines the startup processor core in the available processor cores according to a preset policy; the management operating system randomly selects, from the available processor cores, one processor core as the startup processor core; the management operating system acquires a user instruction, and determines the startup processor core in the available processor cores according to the user instruction.

[0126] In step 403, the management operating system sets, for the first load operating system, the daemon on the startup processor core; after the setting is complete, the daemon runs all the time on the startup processor core that has been started up, and waits for various system commands sent by the management operating system. The daemon is preset by the management operating system for the first load operating system. The management operating system further presets, in the daemon, the location that is of the startup mirror corresponding to the first load operating system and that is in the physical memory address.

[0127] Further, in step 404, the management operating system presets, in the daemon, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror of the first load operating system.

[0128] In step 405, the management operating system sends the inter-core interrupt message to the daemon running on the startup processor core, and an instruction corresponding to the inter-core interrupt message is agreed on by the management operating system and the first load operating system in advance. After the daemon running on the startup processor core receives the inter-core interrupt message sent by the management operating system, the startup processor core reads, from the daemon, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror and is pre-constructed by the management operating system for the first load operating system; in a process of executing the startup mirror by the startup processor core, the startup processor core accesses the physical memory address according to the mapping relationship.

[0129] The management operating system instructs the startup processor core to jump to

the first load operating system as the execution body. The first load operating system may be specifically any load operating system in the load operating systems shown in FIG. 1.

[0139] In step 501, after the management operating system determines the startup processor core for the first load operating system, the startup processor core of the first load operating system first acquires the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror and is set by the management operating system for the first load operating system, and the first load operating system may directly access the physical memory address according to the mapping relationship and does not need to perform address translation.

[0140] In step 502, the startup processor core of the first load operating system receives the instruction that is preset by the management operating system for the first load operating system and that is for executing the startup mirror jump, the first load operating system jumps, according to the instruction that is set by the management operating system and that is for executing the startup mirror jump, to the location that is of the startup mirror corresponding to the first load operating system and that is in the physical memory address; when the startup processor core of the first load operating system executes the startup mirror, the first load operating system directly accesses the physical memory address according to the mapping relationship that is from the virtual memory address to the physical memory address and that is corresponding to the startup mirror, so as to complete a startup process of the startup mirror.

[0141] In step 503, the startup processor core of the first load operating system acquires, in a process of executing the startup mirror of the first load operating system, the startup parameter of the first load operating system from the startup mirror of the first load operating system, where the startup parameter includes the identifiers of the processor cores allocated by the management operating system to the first load operating system and the identifier of the physical memory address allocated by the management operating system to the first load operating system.

[0142] Optionally, when the physical resource set includes some external devices of the host machine, the startup parameter further includes an identifier of an external device allocated by the management operating system to the first load operating system and a device

identifiers of the processor cores and the identifier of the physical memory address of the first load operating system, a physical resource set that is allocated by the management operating system to the first load operating system and that is exclusively used by the first load operating system. The first load operating system reads the startup parameter from the startup mirror, so as to acquire the physical resource set that is allocated by the management operating system to the first load operating system and that is exclusively used by the load operating system, without requiring involvement of a hypervisor, thereby simplifying a process of allocating a physical resource by the management operating system to the load operating system. Further, each load operating system can exclusively use a physical resource, so as to construct an independent page table, clock, runtime environment, and the like on the physical resource exclusively used by each load operating system, thereby improving overall performance of the load operating systems.

[0146] The following uses several specific embodiments to describe in detail the technical solutions of the method embodiment shown in FIG. 5.

[0147] A specific embodiment may be that shown in FIG. 6. FIG. 6 is a schematic flowchart of Embodiment 5 of a resource processing method according to the present invention. As shown in FIG. 6, the resource processing method provided in this embodiment of the present invention includes:

[0148] Step 601: The startup processor core of the first load operating system receives an inter-core interrupt message sent by the management operating system, where the startup processor core executes, according to an instruction of the inter-core interrupt message, an initialization program that is configured to initialize the startup processor core and that is corresponding to the first load operating system.

[0149] Step 602: The startup processor core of the first load operating system executes the initialization program, reads, from the initialization program, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror and is pre-constructed by the management operating system for the first load operating system, reads a jump instruction that is preset in the initialization program by the management operating system, jumps to the location that is of the startup mirror corresponding to the first load operating system and that is in the physical memory

mirror, acquires a physical resource set according to the startup parameter, and completes a startup process of the first load operating system. The management operating system allocates, to a load operating system, a physical resource exclusively used by the load operating system, without requiring involvement of a hypervisor, thereby simplifying a process of allocating the physical resource by the management operating system to load operating system; further, each load operating system can exclusively use a physical resource, thereby improving overall performance of the load operating systems.

[0160] Another specific embodiment is as follows: In this embodiment, the startup processor core of the first load operating system is a processor core that has been started up and that is determined by the management operating system, a daemon runs on the startup processor core of the first load operating system, the daemon is preset by the management operating system for the first load operating system, the location that is of the startup mirror corresponding to the first load operating system and that is in the physical memory address and the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror are set in the daemon. FIG. 7 is a schematic flowchart of Embodiment 6 of a resource processing method according to the present invention. As shown in FIG. 7, the resource processing method provided in this embodiment of the present invention includes:

[0161] Step 701: The daemon running on the startup processor core of the first load operating system receives an inter-core interrupt message sent by the management operating system, and the startup processor core reads, from the daemon according to an instruction of the inter-core interrupt message, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror and is pre-constructed by the management operating system for the first load operating system.

[0162] Step 702: The startup processor core of the first load operating system reads a jump instruction that is preset in the daemon by the management operating system, jumps to the location that is of the startup mirror corresponding to the first load operating system and that is in the physical memory address, and executes, according to the mapping relationship, the startup mirror corresponding to the first load operating system.

that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror.

[0248] The management operating system provided in this embodiment may execute the technical solutions of the foregoing resource processing method embodiments, and has a similar implementation principle and technical effect, and no detail is repeatedly described herein in this embodiment.

[0249] FIG. 13 is a schematic structural diagram of Embodiment 3 of a management operating system according to the present invention. This embodiment is implemented based on the embodiment in FIG. 11, and is specifically as follows:

[0250] Optionally, the management operating system further includes an available-processor-core acquiring module 1106, configured to acquire available processor cores of a second load operating system before the startup processor core determining module determines, in the processor cores allocated to the first load operating system, the startup processor core that starts up the first load operating system, where the second load operating system is a logged-out load operating system, other than the first load operating system, in the multiple load operating systems, the available processor cores include a processor core that has been started up, and a physical resource set of the first load operating system includes the available processor cores.

[0251] The startup processor core determining module 1103 is specifically configured to: when the physical resource set of the first load operating system includes the available processor cores, determine, in the available processor cores, the startup processor core that starts up the first load operating system.

[0252] Optionally, the management operating system further includes a daemon configuring module 1107, configured to preset, for the first load operating system, a daemon on the startup processor core after the startup processor core determining module determines, in the available processor cores, the startup processor core that starts up the first load operating system, and preset, in the daemon, a location that is of the startup mirror of the first load operating system and that is in the physical memory address.

[0253] The mapping relationship setting module 1102 is specifically configured to set, for the first load operating system and in the daemon, the mapping relationship that is from the

to the first load operating system and a device description information table corresponding to the external device.

[0259] Based on the embodiments in FIG. 12 and FIG. 13, when the management operating system monitors running of the first load operating system, a structure of the management operating system includes two possible implementation manners.

[0260] One possible implementation manner is that the management operating system further includes:

a first monitoring module, configured to monitor, after the instructing module instructs the startup processor core to execute the startup mirror pre-constructed for the first load operating system, running of the first load operating system; and

a first resource managing module, configured to determine that a physical resource needs to be reduced for the first load operating system if a monitoring result of the first monitoring module is that utilization of a physical resource allocated to the first load operating system is less than a first preset threshold, and determine a type identifier of the physical resource that needs to be reduced, where the type identifier of the physical resource is used to distinguish whether the physical resource is a processor core resource, a physical memory resource, or an external device resource.

[0261] The resource allocating module is further configured to perform, according to the type identifier of the physical resource that needs to be reduced, second-time division on the available physical resources that are in the physical resource pool and that are corresponding to the identifiers of the physical resources in the list of the available physical resources in the resource pool, and reduce the physical resource of the first load operating system, where the reduced physical resource is a physical resource corresponding to the type identifier of the physical resource that needs to be reduced.

[0262] The other possible implementation manner is that the management operating system further includes:

a second monitoring module, configured to monitor, after the instructing module instructs the startup processor core to execute the startup mirror pre-constructed for the first load operating system, running of the first load operating system; and

a second resource managing module, configured to determine that a physical resource needs to be added for the first load operating system if a monitoring result of the second monitoring module is that utilization of a physical resource allocated to the first load operating system is greater than a second preset threshold, and determine a type identifier of the physical resource that needs to be added, where the type identifier of the physical resource is used to distinguish whether the physical resource that needs to be added is a processor core resource, a physical memory resource, or an external device resource.

[0263] The resource allocating module is further configured to perform, according to the type identifier of the physical resource that needs to be added, second-time division on the available physical resources that are in the physical resource pool and that are corresponding to the identifiers of the physical resources in the list of the available physical resources in the resource pool, and add the physical resource of the first load operating system, where the added physical resource is a physical resource corresponding to the type identifier of the physical resource that needs to be added.

[0264] Based on the embodiments in FIG. 12 and FIG. 13, when the management operating system monitors running of the first load operating system, a structure of the management operating system includes two possible implementation manners.

[0265] One possible implementation manner is that the management operating system further includes:

a first receiving module, configured to receive, after the instructing module instructs the startup processor core to execute the startup mirror pre-constructed for the first load operating system, a resource reducing request message sent by the first load operating system, where the resource reducing request message includes a type identifier of a physical resource that needs to be reduced, and the type identifier of the physical resource is used to distinguish whether the physical resource is a processor core resource, a physical memory resource, or an external device resource.

[0266] The resource allocating module is further configured to perform, according to the type identifier, of the physical resource that needs to be reduced, in the resource reducing request message, second-time division on the available physical resources that are in the

physical resource pool and that are corresponding to the identifiers of the physical resources in the list of the available physical resources in the resource pool, and reduce the physical resource of the first load operating system, where the reduced physical resource is a physical resource corresponding to the type identifier of the physical resource that needs to be reduced.

[0267] The other possible implementation manner is that the management operating system further includes:

a second receiving module, configured to receive, after the instructing module instructs the startup processor core to execute the startup mirror pre-constructed for the first load operating system, a resource adding request message sent by the first load operating system, where the resource adding request message includes a type identifier of a physical resource that needs to be added, and the type identifier of the physical resource is used to distinguish whether the physical resource is a processor core resource, a physical memory resource, or an external device resource.

[0268] The resource allocatingmodule is further configured to perform, according to the type identifier, of the physical resource that needs to be added, in the resource adding request message, second-time division on the available physical resources that are in the physical resource pool and that are corresponding to the identifiers of the physical resources in the list of the available physical resources in the resource pool, and add the physical resource of the first load operating system, where the added physical resource is a physical resource corresponding to the type identifier of the physical resource that needs to be added.

[0269] Based on the foregoing embodiments, if the physical resource that needs to be added is a processor core, the resource allocating module is further specifically configured to:

mark the processor core, which needs to be added for the first load operating system, as a physical resource that belongs to the first load operating system, and send a resource adding message to the first load operating system, where the resource adding message includes an identifier of the processor core that needs to be added; and

send a jump instruction to the added processor core, where the jump instruction is used to instruct the added processor core to execute a kernel program of the first load operating system, and instruct the added processor core to read, from the kernel program, all

mapping relationships that are from virtual memory addresses to physical memory addresses and that are required for executing the first load operating system.

[0270] Based on the foregoing embodiments, if the physical resource that needs to be reduced is a processor core, the resource allocating module is further specifically configured to:

send a resource reducing message to the first load operating system, where the resource reducing message includes an identifier of the processor core that needs to be released; and

acquire the identifier of the processor core released by the first load operating system, and add the identifier of the released processor core to the list of the available physical resources in the resource pool, where the released processor core is a processor core corresponding to the identifier of the processor core that needs to be reduced.

[0271] Based on the foregoing embodiments, if the physical resource that needs to be added is a physical memory resource, the resource allocating module is further specifically configured to: mark a physical memory address, which needs to be added for the first load operating system, as a physical resource that belongs to the first load operating system, and send a resource adding message to the first load operating system, where the resource adding message includes an identifier of the physical memory address that needs to be added.

[0272] Based on the foregoing embodiments, if the physical resource that needs to be reduced is a physical memory resource, the resource allocating module is further specifically configured to: send a resource reducing message to the first load operating system, where the resource reducing message includes an identifier of a physical memory address that needs to be released; and

acquire an identifier of the physical memory address released by the first load operating system, and add the identifier of the released physical memory address to the list of the available physical resources in the resource pool, where the released physical memory address is a physical memory address corresponding to the identifier of the physical memory address that needs to be released.

[0273] Based on the foregoing embodiments, if the physical resource that needs to be added is an external device resource, the resource allocating module is further specifically

configured to: mark an external device, which needs to be added for the first load operating system, as a physical resource that belongs to the first load operating system, and send a hot-insertion message to the first load operating system, where the hot-insertion message is used to instruct the first load operating system to acquire an identifier of the external device that needs to be added and a device description information table corresponding to the added external device.

[0274] Based on the foregoing embodiments, if the physical resource that needs to be reduced is an external device resource, the resource allocating module is further specifically configured to: send a hot-removal message to the first load operating system, where the hot-removal message includes an identifier of an external device that needs to be reduced; and

acquire the identifier of the external device reduced for the first load operating system, and add the identifier of the reduced external device to the list of the available physical resources in the resource pool, where the reduced external device is an external device corresponding to the identifier of the external device that needs to be reduced.

[0275] The management operating system provided in this embodiment may execute the technical solutions of the foregoing resource processing method embodiments, and has a similar implementation principle and technical effect, and no detail is repeatedly described herein in this embodiment.

[0276] Based on the embodiments in FIG. 11 to FIG. 13, the management operating system may further include another module required for running, for example:

an inter-instance communication module, configured to implement bottom-layer communication between the management operating system and another load operating system, where notification and response are generally performed not by using a network but in an inter-processor interrupt manner;

a resource runtime assisting module, where because of a limitation of a hardware function, when an interrupt of an external device cannot be directly routed to a load operating system, the resource runtime assisting module of the management operating system may forward the interrupt, and help to implement normal response to the interrupt; or when the hardware function fully supports automatic routing of an interrupt and free configuration of direct memory access (Direct Memory Access, DMA for short), that is, when an interrupt of an

mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror and is pre-constructed by the management operating system for the first load operating system.

[0285] The instruction executing module 1402 is specifically configured to: execute the initialization program, read, from the initialization program, a jump instruction that is preset in the initialization program by the management operating system, and jump to the location that is of the startup mirror corresponding to the first load operating system and that is in the physical memory address.

[0286] Optionally, the startup processor core of the first load operating system is a processor core that has been started up and that is determined by the management operating system, a daemon runs on the startup processor core of the first load operating system, the daemon is preset by the management operating system for the first load operating system, the location that is of the startup mirror corresponding to the first load operating system and that is in the physical memory address and the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror are set in the daemon.

[0287] Optionally, the mapping relationship acquiring module 1401 is specifically configured to read, from the daemon and according to an instruction of an inter-core interrupt message after the daemon running on the startup processor core receives the inter-core interrupt message sent by the management operating system, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the startup mirror and is pre-constructed by the management operating system for the first load operating system.

[0288] The instruction executing module 1402 is specifically configured to: read a jump instruction that is preset in the daemon by the management operating system, and jump to the location that is of the startup mirror corresponding to the first load operating system and that is in the physical memory address.

[0289] Optionally, each physical resource set further includes some external devices of the host machine; and

the startup parameter further includes an identifier of an external device allocated

the management operating system if a monitoring result of the first monitoring module is that utilization of the physical resource allocated by the management operating system to the first load operating system is less than a first preset threshold, where the resource reducing request message includes a type identifier of a physical resource that needs to be reduced, and the type identifier of the physical resource is used to distinguish whether the physical resource that needs to be reduced is a processor core resource, a physical memory resource, or an external device resource.

[0294] Optionally, the load operating system further includes:

a second monitoring module, configured to monitor a physical resource of the first load operating system after the physical resource set acquiring module determines the physical resource set that is allocated by the management operating system to the first load operating system and that is exclusively used by the first load operating system; and

a second sending module, configured to send a resource adding request message to the management operating system if a monitoring result of the first monitoring module is that utilization of the physical resource allocated by the management operating system to the first load operating system is greater than a second preset threshold, where the resource adding request message includes a type identifier of a physical resource that needs to be added, and the type identifier of the physical resource is used to distinguish whether the physical resource that needs to be added is a processor core resource or a physical memory resource.

[0295] Optionally, if the physical resource that needs to be added is a processor core, the physical resource set acquiring module is further specifically configured to: after the second sending module sends the resource adding request message to the management operating system,

receive a resource adding message sent by the management operating system, where the resource adding message includes an identifier of the processor core that needs to be added; and

determine, according to the identifier of the processor core that needs to be added, the processor core added by the management operating system for the first load operating system; and

when the added processor core of the first load operating system receives a jump

CLAIMS

What is claimed is:

1. A resource processing method, wherein the method is applied to a multi-kernel operating system, the multi-kernel operating system comprises a management operating system and multiple load operating systems that run on a host machine including a plurality of processors, wherein the management operating system runs on a first processor in the plurality of processors and a first load operating system runs on a second processor in the plurality of processors, wherein, the host machine comprises a physical resource pool, the physical resource pool comprises processor cores and a physical memory that are of the host machine, and the method comprises:

determining, by the management operating system and in processor cores allocated to the first load operating system, a startup processor core that starts up the first load operating system, wherein the first load operating system is any load operating system in the multiple load operating systems, the management operating system pre-allocates, to the first load operating system, a physical resource set exclusively used by the first load operating system, and the physical resource set comprises some of the processor cores and a part of the physical memory that are of the host machine;

instructing, by the management operating system, the startup processor core to read a mapping relationship that is from a virtual memory address to a physical memory address and that is required for executing a boot image of the first load operating system and is pre-constructed by the management operating system, wherein the physical memory address comprises a location of the boot image corresponding to the first load operating system; and

instructing, by the management operating system, the startup processor core to execute the boot image pre-constructed for the first load operating system, so as to start up the first load operating system, wherein the boot image comprises a startup parameter of the first load operating system, and the startup parameter comprises identifiers of the processor cores allocated to the first load operating system and an identifier of the physical memory address allocated to the first load operating system;

accessing, by the first load operating system without the involvement of the management operating system, a physical resource in the physical resource set allocated to the first load operating system after the first load operating system is started up, wherein the accessing comprises the processor cores of the first load operating system convert, according to mapping relationships from virtual memory addresses to physical memory addresses, the virtual memory addresses into the physical memory addresses directly, and access the physical memory addresses, wherein the mapping relationships are constructed by the first load operating system.

2. The method according to claim 1, wherein the method further comprises:

constructing, by the management operating system, the boot image for the first load operating system; and

setting, by the management operating system for the first load operating system, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the boot image.

3. The method according to claim 1 or 2, wherein the instructing, by the management operating system, the startup processor core to read a mapping relationship that is from a virtual memory address to a physical memory address and that is required for executing a boot image of the first load operating system and is pre-constructed by the management operating system comprises:

sending, by the management operating system, an inter-core interrupt message to the startup processor core, wherein the inter-core interrupt message instructs the startup processor core to execute an initialization program that is configured to initialize the startup processor core and that is corresponding to the first load operating system; and

instructing, by the management operating system by instructing the startup processor core to execute the initialization program corresponding to the first load operating system, the startup processor core to read, from the initialization program, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the boot image of the first load operating system; and

the instructing, by the management operating system, the startup processor core to execute the boot image pre-constructed for the first load operating system comprises:

instructing, by the management operating system, the startup processor core to jump, after running of the initialization program ends, to the location that is of the boot image and is in the physical memory, so as to execute the boot image of the first load operating system.

4. The method according to claim 1 or 2, wherein before the determining, by the management operating system and in processor cores allocated to a first load operating system, a startup processor core that starts up the first load operating system, the method further comprises:

acquiring, by the management operating system, available processor cores of a second load operating system, wherein the second load operating system is a logged-out load operating system, other than the first load operating system, in the multiple load operating systems, the available processor cores comprise a processor core that has been started up; and

the determining, by the management operating system and in processor cores allocated to a first load operating system, a startup processor core that starts up the first load operating system comprises:

when the physical resource set of the first load operating system comprises the available processor cores, determining, by the management operating system and in the available processor cores, the startup processor core that starts up the first load operating system.

5. The method according to claim 1, wherein after the instructing, by the management operating system, the startup processor core to execute the boot image pre-constructed for the first load operating system, the method further comprises:

monitoring, by the management operating system, running of the first load operating system; and

when utilization of a physical resource allocated to the first load operating system is less than a first preset threshold, determining, by the management operating system, a type identifier of a physical resource that needs to be reduced for the first load operating system, and reducing the physical resource of the first load operating system according to the type identifier of the physical resource that needs to be reduced; or

when utilization of a physical resource allocated to the first load operating system is greater than a second preset threshold, determining, by the management operating system, a type identifier of a physical resource that needs to be added for the first load operating system,

and adding the physical resource of the first load operating system according to the type identifier of the physical resource that needs to be added, wherein

the type identifier of the physical resource is used to distinguish whether the physical resource is a processor core resource or a physical memory resource.

6. The method according to claim 1, wherein

the physical resource set further comprises some external devices of the host machine; the startup parameter further comprises an identifier of an external device allocated to the first load operating system and a device description information table corresponding to the external device; and

after the instructing, by the management operating system, the startup processor core to execute the boot image pre-constructed for the first load operating system, the method further comprises:

receiving, by the management operating system, a resource reducing request message sent by the first load operating system, wherein the resource reducing request message comprises a type identifier of a physical resource that needs to be reduced, and the type identifier of the physical resource is used to distinguish whether the physical resource is a processor core resource, a physical memory resource, or an external device resource; and

reducing, by the management operating system, the physical resource of the first load operating system according to the type identifier of the physical resource that needs to be reduced.

7. The method according to claim 1, wherein the physical resource set further comprises some external devices of the host machine; the startup parameter further comprises an identifier of an external device allocated to the first load operating system and a device description information table corresponding to the external device; and

after the instructing, by the management operating system, the startup processor core to execute the boot image pre-constructed for the first load operating system, the method further comprises:

receiving, by the management operating system, a resource adding request message sent by the first load operating system, wherein the resource adding request message comprises a type identifier of a physical resource that needs to be added, and the type identifier of the

physical resource is used to distinguish whether the physical resource is a processor core resource, a physical memory resource, or an external device resource; and

adding, by the management operating system, the physical resource of the first load operating system according to the type identifier of the physical resource that needs to be added.

8. The method according to claim 5 or 7, wherein if the physical resource that needs to be added is a processor core, the adding, by the management operating system, the physical resource of the first load operating system comprises:

marking, by the management operating system, the processor core, which needs to be added for the first load operating system, as a physical resource that belongs to the first load operating system, and sending a resource adding message to the first load operating system, wherein the resource adding message comprises an identifier of the processor core that needs to be added; and

sending, by the management operating system, a jump instruction to the added processor core, wherein the jump instruction is used to instruct the added processor core to execute a kernel program of the first load operating system, and the kernel program comprises all mapping relationships that are from virtual memory addresses to physical memory addresses and that are required for executing the first load operating system.

9. The method according to claim 5 or 6, wherein if the physical resource that needs to be reduced comprises a processor core or a physical memory resource, the reducing, by the management operating system, the physical resource of the first load operating system according to the type identifier of the physical resource that needs to be reduced comprises:

sending, by the management operating system, a resource reducing message to the first load operating system, wherein the resource reducing message comprises an identifier of the physical resource that needs to be released; and

acquiring, by the management operating system, the identifier of the physical resource released by the first load operating system, and adding the identifier of the released physical resource to a list of available physical resources in the resource pool.

10. The method according to claim 5 or 7, wherein if the physical resource that needs to be added is a physical memory resource, the adding, by the management operating system,

the physical resource of the first load operating system comprises:

marking, by the management operating system, a physical memory address, which needs to be added for the first load operating system, as a physical resource that belongs to the first load operating system, and sending a resource adding message to the first load operating system, wherein the resource adding message comprises an identifier of the physical memory address that needs to be added.

11. A resource processing method, wherein the method is applied to a multi-kernel operating system, the multi-kernel operating system comprises a management operating system and multiple load operating systems that run on a host machine including a plurality of processors, wherein the management operating system runs on a first processor in the plurality of processors and a first load operating system runs on a second processor in the plurality of processors, wherein the host machine comprises a physical resource pool, the physical resource pool comprises processor cores and a physical memory that are of the host machine, each load operating system exclusively uses a physical resource set allocated by the management operating system, each physical resource set comprises some of the processor cores of the host machine and a part of the physical memory of the host machine, and the management operating system pre-constructs a boot image for each load operating system, and sets, for each load operating system, a mapping relationship that is from a virtual memory address to a physical memory address and that is required for executing the boot image, and the method comprises:

reading, by a startup processor core of a first load operating system, a mapping relationship that is from a virtual memory address to a physical memory address and that is required for executing a boot image of the first load operating system, wherein the mapping relationship is used to enable the startup processor core of the first load operating system to access the physical memory address, and the first load operating system is any load operating system in the multiple load operating systems;

jumping, by the startup processor core of the first load operating system according to an instruction that is set by the management operating system and that is for executing a boot image jump, to a location that is of the boot image corresponding to the first load operating system and that is in the physical memory address, and executing, according to the mapping

relationship, the boot image corresponding to the first load operating system to start up the first load operating system;

acquiring, by the startup processor core of the first load operating system and in a process of executing the boot image of the first load operating system, a startup parameter of the first load operating system from the boot image of the first load operating system, wherein the startup parameter comprises identifiers of processor cores allocated to the first load operating system and an identifier of the physical memory address allocated to the first load operating system; and

determining, by the first load operating system according to the identifiers of the processor cores and the identifier of the physical memory address of the first load operating system, a physical resource set that is allocated by the management operating system to the first load operating system and that is exclusively used by the first load operating system;

accessing, by the first load operating system without the involvement of the management operating system, a physical resource in the physical resource set allocated to the first load operating system after the first load operating system is started up, wherein the accessing comprises the processor cores of the first load operating system convert, according to mapping relationships from virtual memory addresses to physical memory addresses, the virtual memory addresses into the physical memory addresses directly, and access the physical memory addresses, wherein the mapping relationships are constructed by the first load operating system.

12. The resource processing method according to claim 11, wherein the reading, by a startup processor core of a first load operating system, a mapping relationship that is from a virtual memory address to a physical memory address and that is required for executing a boot image of the first load operating system comprises:

receiving, by the startup processor core of the first load operating system, an inter-core interrupt message sent by the management operating system, wherein the inter-core interrupt message is used to instruct the startup processor core to execute an initialization program that is configured to initialize the startup processor core and that is corresponding to the first load operating system; and

executing, by the startup processor core of the first load operating system, the

initialization program, and reading, from the initialization program, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the boot image and is pre-constructed by the management operating system for the first load operating system; and

the jumping, by the startup processor core of the first load operating system according to an instruction that is set by the management operating system and that is for executing a boot image jump, to a location that is of the boot image corresponding to the first load operating system and that is in the physical memory address comprises:

executing, by the startup processor core of the first load operating system, the initialization program, reading, from the initialization program, a jump instruction that is preset in the initialization program by the management operating system, and jumping to the location that is of the boot image corresponding to the first load operating system and that is in the physical memory address.

13. The method according to claim 11, wherein after the determining, by the first load operating system, a physical resource set that is allocated by the management operating system to the first load operating system and that is exclusively used by the first load operating system, the method further comprises:

constructing, by the first load operating system, all mapping relationships that are from virtual memory addresses to physical memory addresses and that are required for executing the first load operating system;

accessing, by the processor cores of the first load operating system, the physical memory address according to all the mapping relationships that are from the virtual memory addresses to the physical memory addresses and that are required for executing the first load operating system.

14. The method according to claim 11, wherein after the determining, by the first load operating system, a physical resource set that is allocated by the management operating system to the first load operating system and that is exclusively used by the first load operating system, the method further comprises:

executing, by the first load operating system by using the processor cores allocated to the first load operating system, a privileged instruction delivered by the management operating

system.

15. The method according to claim 11, wherein after the determining, by the first load operating system, a physical resource set that is allocated by the management operating system to the first load operating system and that is exclusively used by the first load operating system, the method further comprises:

monitoring, by the first load operating system, a physical resource of the first load operating system; and

if the first load operating system determines that utilization of the physical resource allocated by the management operating system to the first load operating system is less than a first preset threshold, sending, by the first load operating system, a resource reducing request message to the management operating system, wherein the resource reducing request message comprises a type identifier of a physical resource that needs to be reduced; or

if the first load operating system determines that utilization of the physical resource allocated by the management operating system to the first load operating system is greater than a second preset threshold, sending, by the first load operating system, a resource adding request message to the management operating system, wherein the resource adding request message comprises a type identifier of a physical resource that needs to be added, wherein

the type identifier of the physical resource is used to distinguish whether the physical resource is a processor core resource or a physical memory resource.

16. The method according to claim 15, wherein if the physical resource that needs to be added is a processor core, after the sending, by the first load operating system, a resource adding request message to the management operating system, the method further comprises:

receiving, by the first load operating system, a resource adding message sent by the management operating system, wherein the resource adding message comprises an identifier of the processor core that needs to be added;

determining, by the first load operating system according to the identifier of the processor core that needs to be added, the processor core added by the management operating system for the first load operating system; and

receiving, by the added processor core of the first load operating system, a jump instruction sent by the management operating system, executing a kernel program of the first

load operating system according to the jump instruction, and reading, from the kernel program, all mapping relationships that are from virtual memory addresses to physical memory addresses and that are required for executing the first load operating system, wherein the first load operating system presets, in the kernel program, all the mapping relationships that are from the virtual memory addresses to the physical memory addresses and that are required for executing the first load operating system.

17. The method according to claim 15, wherein if the physical resource that needs to be reduced is a processor core or a physical memory resource, after the sending, by the first load operating system, a resource reducing request message to the management operating system, the method further comprises:

receiving, by the first load operating system, a resource reducing message sent by the management operating system, wherein the resource reducing message comprises an identifier of the physical resource that needs to be released; and

releasing, by the first load operating system, the physical resource according to the identifier of the physical resource that needs to be released.

18. The method according to claim 15, wherein if the physical resource that needs to be added is a physical memory resource, after the sending, by the first load operating system, a resource adding request message to the management operating system, the method further comprises:

receiving, by the first load operating system, a resource adding message sent by the management operating system, wherein the resource adding message comprises an identifier of a physical memory address that needs to be added; and

determining, by the first load operating system according to the identifier of the physical memory address that needs to be added, the physical memory address added by the management operating system for the first load operating system.

19. A host machine, wherein the host machine is configured to run a management operating system and multiple load operating systems, the multiple load operating systems comprise a first load operating system, the host machine comprises a plurality of processors, wherein the management operating system runs on a first processor in the plurality of processors and a first load operating system runs on a second processor in the plurality of

processors, wherein the host machine comprises a physical resource pool, the physical resource pool comprises processor cores and a physical memory that are of the host machine, and when the host machine runs, the management operating system is configured to:

determine, in processor cores allocated to the first load operating system, a startup processor core that starts up the first load operating system, wherein the first load operating system is any load operating system in the multiple load operating systems, the management operating system allocates, to the first load operating system, a physical resource set exclusively used by the first load operating system, and the physical resource set comprises some of the processor cores and a part of the physical memory that are of the host machine;

instruct the startup processor core to read a mapping relationship that is from a virtual memory address to a physical memory address and that is required for executing a boot image of the first load operating system and is pre-constructed by the management operating system, wherein the physical memory address comprises a location of the boot image corresponding to the first load operating system; and

instruct the startup processor core to execute the boot image pre-constructed for the first load operating system, so as to start up the first load operating system, wherein the boot image comprises a startup parameter of the first load operating system, and the startup parameter comprises identifiers of the processor cores allocated to the first load operating system and an identifier of the physical memory address allocated to the first load operating system; and

the first load operating system is configured to:

read the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the boot image of the first load operating system, wherein the mapping relationship is used to enable the startup processor core of the first load operating system to access the physical memory address;

jump, according to an instruction that is set by the management operating system and that is for executing a boot image jump, to the location that is of the boot image corresponding to the first load operating system and that is in the physical memory address, and execute, according to the mapping relationship, the boot image corresponding to the first load operating system;

acquire, in a process of executing the boot image of the first load operating system, the

startup parameter of the first load operating system from the boot image of the first load operating system; and

determine, according to the identifiers of the processor cores and the identifier of the physical memory address of the first load operating system, the physical resource set that is allocated by the management operating system to the first load operating system and that is exclusively used by the first load operating system; and

accessing, by the first load operating system without the involvement of the management operating system, a physical resource in the physical resource set allocated to the first load operating system after the first load operating system is started up, wherein the accessing comprises the processor cores of the first load operating system convert, according to mapping relationships from virtual memory addresses to physical memory addresses, the virtual memory addresses into the physical memory addresses directly, and access the physical memory addresses, wherein the mapping relationships are constructed by the first load operating system.

20. The host machine according to claim 19, wherein the management operating system is further configured to:

construct the boot image for the first load operating system; and

set, for the first load operating system, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the boot image.

21. The host machine according to claim 19 or 20, wherein the management operating system is specifically configured to:

send an inter-core interrupt message to the startup processor core, wherein the inter-core interrupt message instructs the startup processor core to execute an initialization program that is configured to initialize the startup processor core and that is corresponding to the first load operating system;

instruct, by instructing the startup processor core to execute the initialization program corresponding to the first load operating system, the startup processor core to read, from the initialization program, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the boot image of the first load

operating system; and

instruct the startup processor core to jump, after running of the initialization program ends, to the location that is of the boot image and is in the physical memory, so as to execute the boot image of the first load operating system; and

the first load operating system is specifically configured to:

receive the inter-core interrupt message sent by the management operating system, wherein the inter-core interrupt message is used to instruct the startup processor core to execute the initialization program that is configured to initialize the startup processor core and that is corresponding to the first load operating system; and

execute the initialization program, and read, from the initialization program, the mapping relationship that is from the virtual memory address to the physical memory address and that is required for executing the boot image and is pre-constructed by the management operating system for the first load operating system; and

execute the initialization program, read, from the initialization program, a jump instruction that is preset in the initialization program by the management operating system, and jump to the location that is of the boot image corresponding to the first load operating system and that is in the physical memory address.

22. The host machine according to claim 19 or 20, wherein the management operating system is further configured to:

acquire available processor cores of a second load operating system before determining, in the processor cores allocated to the first load operating system, the startup processor core that starts up the first load operating system, wherein the second load operating system is a logged-out load operating system, other than the first load operating system, in the multiple load operating systems, the available processor cores comprise a processor core that has been started up; and

when the physical resource set of the first load operating system comprises the available processor cores, determine, in the available processor cores, the startup processor core that starts up the first load operating system.

23. The host machine according to claim 19 or 20, wherein the first load operating system is further configured to:

after determining the physical resource set that is allocated by the management operating system to the first load operating system and that is exclusively used by the first load operating system, construct all mapping relationships that are from virtual memory addresses to physical memory addresses and that are required for executing the first load operating system; and

access the physical memory address according to all the mapping relationships that are from the virtual memory addresses to the physical memory addresses and that are required for executing the first load operating system.

24. The host machine according to claim 19 or 20, wherein the first load operating system is further configured to:

execute, by using the processor cores allocated to the first load operating system, a privileged instruction delivered by the management operating system.

25. The host machine according to claim 19, wherein the management operating system is further configured to:

monitor running of the first load operating system after instructing the startup processor core to execute the boot image pre-constructed for the first load operating system; and

when utilization of a physical resource allocated to the first load operating system is less than a first preset threshold, determine a type identifier of a physical resource that needs to be reduced for the first load operating system, and reduce the physical resource of the first load operating system according to the type identifier of the physical resource that needs to be reduced; or

when utilization of a physical resource allocated to the first load operating system is greater than a second preset threshold, determine a type identifier of a physical resource that needs to be added for the first load operating system, and add the physical resource of the first load operating system according to the type identifier of the physical resource that needs to be added, wherein

the type identifier of the physical resource is used to distinguish whether the physical resource is a processor core resource or a physical memory resource.

26. The host machine according to claim 19, wherein the first load operating system is further configured to:

monitor a physical resource of the first load operating system; and

if the first load operating system determines that utilization of the physical resource allocated by the management operating system to the first load operating system is less than a first preset threshold, send a resource reducing request message to the management operating system, wherein the resource reducing request message comprises a type identifier of a physical resource that needs to be reduced, and the type identifier of the physical resource is used to distinguish whether the physical resource is a processor core resource or a physical memory resource; and

the management operating system is further configured to:

receive the resource reducing request message sent by the first load operating system; and

reduce the physical resource of the first load operating system according to the type identifier of the physical resource that needs to be reduced.

27. The host machine according to claim 19, wherein:

the first load operating system is further configured to:

monitor a physical resource of the first load operating system; and

when the first load operating system determines that utilization of the physical resource allocated by the management operating system to the first load operating system is greater than a second preset threshold, send a resource adding request message to the management operating system, wherein the resource adding request message comprises a type identifier of a physical resource that needs to be added, and the type identifier of the physical resource is used to distinguish whether the physical resource is a processor core resource or a physical memory resource; and

the management operating system is further configured to:

receive the resource adding request message sent by the first load operating system; and

add the physical resource of the first load operating system according to the type identifier of the physical resource that needs to be added.

28. The host machine according to claim 25 or 27, wherein if the physical resource that needs to be added is a processor core, the management operating system is specifically configured to:

mark the processor core, which needs to be added for the first load operating system, as a physical resource that belongs to the first load operating system, and send a resource adding message to the first load operating system, wherein the resource adding message comprises an identifier of the processor core that needs to be added; and

send a jump instruction to the added processor core, wherein the jump instruction is used to instruct the added processor core to execute a kernel program of the first load operating system, and the kernel program comprises all mapping relationships that are from virtual memory addresses to physical memory addresses and that are required for executing the first load operating system; and

the first load operating system is specifically configured to:

receive the resource adding message sent by the management operating system;

determine, according to the identifier of the processor core that needs to be added, the processor core added by the management operating system for the first load operating system; and

receive, by using the added processor core of the first load operating system, the jump instruction sent by the management operating system, execute the kernel program of the first load operating system according to the jump instruction, and read, from the kernel program, all the mapping relationships that are from the virtual memory addresses to the physical memory addresses and that are required for executing the first load operating system.

29. The host machine according to claim 25 or 27, wherein if the physical resource that needs to be added is a physical memory resource, the management operating system is specifically configured to:

mark a physical memory address, which needs to be added for the first load operating system, as a physical resource that belongs to the first load operating system, and send a resource adding message to the first load operating system, wherein the resource adding message comprises an identifier of the physical memory address that needs to be added; and

the first load operating system is specifically configured to:

receive the resource adding message sent by the management operating system; and

determine, according to the identifier of the physical memory address that needs to be added, the physical memory address added by the management operating system for the first

load operating system.

30. The host machine according to claim 25 or 26, wherein if the physical resource that needs to be reduced is a processor core or a physical memory resource,

the management operating system is specifically configured to: send a resource reducing message to the first load operating system, wherein the resource reducing message comprises an identifier of the physical resource that needs to be released;

the first load operating system is specifically configured to: receive the resource reducing message sent by the management operating system, and release the physical resource according to the identifier of the physical resource that needs to be released; and

the management operating system acquires the identifier of the physical resource released by the first load operating system, and adds the identifier of the released physical resource to a list of available physical resources in the resource pool.

31. A computer readable medium comprising one or more computer-executable instructions, wherein said one or more computer-executable instructions, when executed on one or more processors of a computer system, cause the computer system to perform the method according to any one of claims 1-10.

32. A computer readable medium comprising one or more computer-executable instructions, wherein said one or more computer-executable instructions, when executed on one or more processors of a computer system, cause the computer system to perform the method according to any one of claims 11-18.