FIG. 1

(54) Title: BOOTING AN APPLICATION FROM MULTIPLE MEMORIES

(57) Abstract: Disclosed herein are system, apparatus, article of manufacture, method and/or computer program product embodiments for booting an application from multiple memories. An embodiment operates by executing in place from a first memory a first portion of the application, loading a second portion of the application from a second memory, and executing the second portion of the application.
BOOTING AN APPLICATION FROM MULTIPLE MEMORIES

CROSS-REFERENCE TO RELATED APPLICATIONS

[0000] This application is an International Application of U.S. Patent Application No. 14/319,079, filed June 30, 2014, all of which is incorporated by reference herein in its entirety.

BACKGROUND

[0001] Generally, current approaches to booting an application use one of two general techniques. The first technique executes an application in place from a non-volatile, low latency memory, e.g. NOR memory. This technique has several disadvantages. Particularly, non-volatile, low latency memory is very expensive; even for simple applications, the amount of memory needed to store the entire application is costly. Attempts to reduce an application's footprint in memory, e.g. compression, adds computational complexity that affects performance by increasing execution time.

[0002] The second technique loads a memory from a non-volatile memory with a higher latency. This technique has its own disadvantages. Because of the higher latency, it is either not possible or undesirable to execute an application in place. Thus, the application needs to be loaded into another memory, typically a volatile memory, e.g. RAM, from which it is executed. The loading time affects performance by also increasing execution time.

SUMMARY

[0003] Provided herein are system, apparatus, article of manufacture, method and/or computer program product embodiments, and/or combinations and sub-combinations thereof, for booting an application from multiple memories.
[0004] An embodiment includes a method for booting an application from multiple memories. The method operates by executing in place from a first memory a first portion of the application, loading a second portion of the application from a second memory, and executing the second portion of the application.

[0005] Another embodiment includes a system for booting an application from multiple memories. The system includes a memory and at least one processor coupled to the memory. The processor is configured to execute in place from a first memory a first portion of an application, load a second portion of the application from a second memory, and execute the second portion of the application.

[0006] A further embodiment includes a tangible computer-readable device having instructions stored thereon that, when executed by at least one computing device, cause the computing device to perform operations. The operations include executing in place from a first memory a first portion of an application, loading a second portion of the application from a second memory, and executing the second portion of the application.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0007] The accompanying drawings are incorporated herein and form a part of the specification.

[0008] FIG. 1 is a block diagram of a system for booting an application from multiple memories, according to an example embodiment.

[0009] FIG. 2 is a block diagram of a memory address map, according to an example embodiment.

[0010] FIG. 3 is a flowchart illustrating a process for accessing a cache, according to an example embodiment.

[0011] FIG. 4 is a block diagram of an execution flow, according to an example embodiment.

[0012] FIG. 5 is an example computer system useful for implementing various embodiments.

[0013] In the drawings, like reference numbers generally indicate identical or similar elements. Additionally, generally, the left-most digit(s) of a reference number identifies the drawing in which the reference number first appears.
DETAILED DESCRIPTION

[0014] FIG. 1 is a block diagram of an example system 100 that includes a microcontroller (MCU) 102, a first memory 104, a second memory 106, and a third memory 108. Although system 100 is depicted as having one MCU and three memories, embodiments of the invention support any number of MCUs and any number of memories.

[0015] In an embodiment, MCU 102 stores data to or loads data from first memory 104, second memory 106, or third memory 108. For example, MCU 102 can boot an application from first memory 104 and second memory 106. MCU 102 can comprise an integrated circuit, application-specific integrated circuit (ASIC), field-programmable gate array (FPGA), processing core, digital signal processor (DSP), microcontroller (MCU), microprocessor (MPU), or any combination thereof.

[0016] An application can be divided into two or more portions. In an embodiment, the application is divided into at least a first portion 112 and a second portion 114, so that first portion 112 is executed in place from first memory 104 by the time second portion 114 has loaded from second memory 106. The application can be divided in numerous ways. For example, a programmer, software engineer or architect, or other developer can segment the code into the first portion 112 and second portion 114 to achieve the property that the first portion 112 is executed in place from first memory 104 by the time second portion 114 has loaded from second memory 106. Automated systems and/or processors for performing the following can also be used. As another example, the application can be developed using an application programming interface (API). The API can provide functionality so that a developer can specify which portions of the application belong in the first portion 112 and second portion 114. Alternatively or additionally, the API can provide functionality so that a developer can specify an order of priority of code, from which a compiler, linker, other tool, or any combination thereof can build the first portion 112 and second portion 114 to achieve the property that the first portion 112 is executed in place from first memory 104 by the time second portion 114 has loaded from second memory 106. In another embodiment, a compiler, linker, other tool, or any combination thereof segment the application into the first portion 112 and second portion 114 without input or instruction from a user to achieve the property that the first portion 112 is
executed in place from first memory 104 by the time second portion 114 has loaded from second memory 106.

[0017] In an embodiment, first portion 112 of the application comprises application initialization code (AIC) and application execution code (AEC). AIC can include instructions that initialize the application. AEC can include instructions for executing the application once the application has been initialized.

[0018] In an embodiment, first memory 104 comprises a non-volatile, low latency memory that supports execution in place (XIP). XIP can refer to a method of executing a program directly from storage rather than copying it to one or more intermediate memories, e.g. random access memory (RAM). For example, first memory 104 can be any non-volatile memory, such as a NOR memory, a read only memory (ROM), an erasable programmable read only memory (EPROM), a mask ROM (MROM), a resistive random access memory (RRAM), a phase-change memory (PCM), or any combination thereof.

[0019] In an embodiment, first memory 104 comprises hardware initialization code (HIC) 110, a first portion 112 of an application, other code, or any combination thereof. MCU 102 can be configured to execute in place HIC 110, a first portion 112 of an application, other code, or any combination thereof.

[0020] In an embodiment, second memory 106 comprises a non-volatile memory having a higher latency than first memory 104. For example, second memory 106 can be a NAND memory, magnetic media (e.g. a hard drive), optical media (e.g. a CD-ROM or DVD), RRAM, a slow, multi-level NOR, or any combination thereof. In an embodiment, second memory 106 does not support XIP or cannot practically be used for XIP due to its latency.

[0021] In an embodiment, second memory 106 comprises a second portion 114 of the application. First portion 112 and second portion 114 may form the complete application or only a part of the application.

[0022] In an embodiment, third memory 108 comprises a volatile, low latency memory. For example, third memory 108 can be a RAM, dynamic RAM (DRAM), synchronous dynamic RAM (SDRAM), static RAM (SRAM), pseudo-static RAM (pSRAM), or any combination thereof. In an embodiment, an application may be copied into third memory
108 from first memory 104, second memory 106, another memory, or any combination thereof.

[0023] In an embodiment, MCU 102 can be configured to load second portion 114 of the application into third memory 108 and to execute second portion 114 of the application from third memory 108.

[0024] FIG. 2 is a block diagram of a memory address map 200, according to an example embodiment. Memory address map 200 reflects the address space layout code in memory. In an embodiment, MCU 102 uses memory address map 200 to locate or address HIC 110, first portion 112 of the application, second portion 114 of the application, other code, or any combination thereof.

[0025] Memory address map 200 comprises a first section 202 and a second section 204. First section 202 includes address mappings of HIC 110 and first portion 112 of the application. In an embodiment, first section 202 corresponds to first memory 104 and maps addresses thereto. For example, first section 202 can correspond to a NOR memory storing HIC 110 and first portion 112 of the application.

[0026] Second section 204 includes address mappings to second portion 114 of the application. In an embodiment, second section 204 corresponds to second memory 106, third memory 108, or any combination thereof, and maps addresses thereto. For example, second section 204 can correspond to a NAND memory or RAM storing second portion 114 of the application.

[0027] Although memory address map 200 has two sections with the particular layout depicted, embodiments of the invention support any number of sections or layouts. For example, sections 202 and 204 can be non-contiguous, interwoven, or any combination thereof. As another example, second memory 106 and third memory 108 can have their own regions in memory address map 200.

[0028] FIG. 3 is a flowchart illustrating a process 300 for booting an application from multiple memories, according to an example embodiment. Process 300 can be performed by processing logic that can comprise hardware (e.g., circuitry, dedicated logic, programmable logic, microcode, etc.), software (e.g., instructions run on a processing device), or a combination thereof. For example, process 300 may be performed by MCU 102.
FIG. 4 is a block diagram of execution flow 400 resulting from process 300, according to an example embodiment, and will be discussed in conjunction with FIG. 3, although it should be understood that process 300 is not limited to the example of FIG. 4. In execution flow 400, the horizontal axis represents time during execution.

In block 302, first portion 112 of the application is executed in place from first memory 104. For example, first portion 112 can be executed directly from first memory 104 rather than copying it to one or more intermediate memories, such as second memory 106 or third memory 108. In an embodiment, MCU 102 executes in place first portion 112 of the application.

In an embodiment, HIC 110 is executed in place from first memory 104. MCU 102 can execute HIC 110 in place from first memory 104 before executing the first portion 112 of the application. FIG. 4 depicts execution of HIC 110 at block 402 and the execution of first portion 112 at block 404. In an embodiment, execution of the first portion 112 of the application includes execution of AIC at block 406 and execution of AEC at block 408.

Referring back to FIG. 3, in block 304, second portion 114 of the application is loaded from second memory 106. In an embodiment, MCU 102 loads second portion 114 of the application from second memory 106. FIG. 4 depicts loading second portion 114 at block 410.

In an embodiment, second portion 114 of the application is loaded from second memory 106 into third memory 108. For example, MCU 102 can load second portion 114 of the application from a NAND memory into a RAM memory.

In an embodiment, at least some of the executing in place of the first portion 112 of the application from the first memory and at least some of the loading the second portion 114 occur in parallel. For example, FIG. 4 depicts execution of the first portion 112 at block 404 and the loading of the second portion 114 at block 410 occurring concurrently.

In an embodiment, second portion 114 is loaded by the time the first portion 112 of the application has been executed in place. For example, FIG. 4 depicts the second portion 114 having been loaded by time 412, the time at which the first portion 112 of the application has been executed in place.
In an embodiment, the second portion 114 is loaded into third memory 108 by the
time first portion 112 has been executed in place. For example, second portion 114 can
be loaded from a NAND memory into RAM by the time first portion 112 has been
executed in place in a NOR memory.

Referring back to FIG. 3, in block 306, second portion 114 of the application is
executed. In an embodiment, MCU 102 executes second portion 114 of the application.
For example, FIG. 4 depicts the second portion 114 being executed at block 414 after
time 412, the time at which the first portion 112 of the application has been executed in
place.

In an embodiment, second portion 114 of the application is executed from third
memory 108. For example, second portion 114 can be executed from a volatile memory,
e.g. RAM.

In an embodiment, second portion 114 of the application is executed using at least
one of demand paging or shadow mode. For example, in shadow mode, second portion
114 may be loaded entirely from second memory 106 into third memory 108, and then
second portion 114 is executed entirely from third memory 108. In another example
using demand paging, the currently required part of the code, which may be represented
as second portion 114, can be loaded and executed out of third memory 108. As one or
more other portions of the application are required, they can be loaded and executed out
of third memory 108 as needed.

Example Computer System

Various embodiments can be implemented, for example, using one or more well-
known computer systems, such as computer system 500 shown in FIG. 5. Computer
system 500 can be any well-known computer capable of performing the functions
described herein, such as computers available from International Business Machines,
Apple, Sun, HP, Dell, Sony, Toshiba, etc. Computer system 500 can also be any
embedded system capable of performing the functions described herein, such as
embedded systems available from Renesas, Freescale, TI, Spansion, etc.

Computer system 500 includes one or more processors (also called central
processing units, or CPUs), such as a processor 504. Processor 504 is connected to a
communication infrastructure or bus 506.
One or more processors 504 may each be a graphics processing unit (GPU). In an
embodiment, a GPU is a processor that is a specialized electronic circuit designed to
rapidly process mathematically intensive applications on electronic devices. The GPU
may have a highly parallel structure that is efficient for parallel processing of large blocks
of data, such as mathematically intensive data common to computer graphics
applications, images and videos.

Computer system 500 also includes user input/output device(s) 503, such as
monitors, keyboards, pointing devices, etc., which communicate with communication
infrastructure 506 through user input/output interface(s) 502.

Computer system 500 also includes a main or primary memory 508, such as
random access memory (RAM). Main memory 508 may include one or more levels of
cache. Main memory 508 has stored therein control logic (i.e., computer software) and/or
data.

Computer system 500 may also include one or more secondary storage devices or
memory 510. Secondary memory 510 may include, for example, a hard disk drive 512
and/or a removable storage device or drive 514. Removable storage drive 514 may be a
floppy disk drive, a magnetic tape drive, a compact disk drive, an optical storage device,
tape backup device, and/or any other storage device/drive.

Removable storage drive 514 may interact with a removable storage unit 518.
Removable storage unit 518 includes a computer usable or readable storage device having
stored thereon computer software (control logic) and/or data. Removable storage unit
518 may be a floppy disk, magnetic tape, compact disk, DVD, optical storage disk, and/
any other computer data storage device. Removable storage drive 514 reads from and/or
writes to removable storage unit 518 in a well-known manner.

According to an exemplary embodiment, secondary memory 510 may include
other means, instrumentalities or other approaches for allowing computer programs
and/or other instructions and/or data to be accessed by computer system 500. Such means,
instrumentalities or other approaches may include, for example, a removable storage unit
522 and an interface 520. Examples of the removable storage unit 522 and the interface
520 may include a program cartridge and cartridge interface (such as that found in video
game devices), a removable memory chip (such as an EPROM or PROM) and associated
socket, a memory stick and USB port, a memory card and associated memory card slot, and/or any other removable storage unit and associated interface.

Computer system 500 may further include a communication or network interface 524. Communication interface 524 enables computer system 500 to communicate and interact with any combination of remote devices, remote networks, remote entities, etc. (individually and collectively referenced by reference number 528). For example, communication interface 524 may allow computer system 500 to communicate with remote devices 528 over communications path 526, which may be wired and/or wireless, and which may include any combination of LANs, WANs, the Internet, etc. Control logic and/or data may be transmitted to and from computer system 500 via communication path 526.

In an embodiment, a tangible apparatus or article of manufacture comprising a tangible computer useable or readable medium having control logic (software) stored thereon is also referred to herein as a computer program product or program storage device. This includes, but is not limited to, computer system 500, main memory 508, secondary memory 510, and removable storage units 518 and 522, as well as tangible articles of manufacture embodying any combination of the foregoing. Such control logic, when executed by one or more data processing devices (such as computer system 500), causes such data processing devices to operate as described herein.

Based on the teachings contained in this disclosure, it will be apparent to persons skilled in the relevant art(s) how to make and use the invention using data processing devices, computer systems and/or computer architectures other than that shown in FIG. 5. In particular, embodiments may operate with software, hardware, and/or operating system implementations other than those described herein.

Conclusion

It is to be appreciated that the Detailed Description section, and not the Summary and Abstract sections, is intended to be used to interpret the claims. The Summary and Abstract sections may set forth one or more but not all exemplary embodiments of the invention as contemplated by the inventors, and thus, are not intended to limit the invention or the appended claims in any way.
While the invention has been described herein with reference to exemplary embodiments for exemplary fields and applications, it should be understood that the invention is not limited thereto. Other embodiments and modifications thereto are possible, and are within the scope and spirit of the invention. For example, and without limiting the generality of this paragraph, embodiments are not limited to the software, hardware, firmware, and/or entities illustrated in the figures and/or described herein. Further, embodiments (whether or not explicitly described herein) have significant utility to fields and applications beyond the examples described herein.

Embodiments have been described herein with the aid of functional building blocks illustrating the implementation of specified functions and relationships thereof. The boundaries of these functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternate boundaries can be defined as long as the specified functions and relationships (or equivalents thereof) are appropriately performed. Also, alternative embodiments may perform functional blocks, steps, operations, methods, etc. using orderings different than those described herein.

References herein to "one embodiment," "an embodiment," "an example embodiment," or similar phrases, indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it would be within the knowledge of persons skilled in the relevant art(s) to incorporate such feature, structure, or characteristic into other embodiments whether or not explicitly mentioned or described herein.

The breadth and scope of the invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.
WHAT IS CLAIMED IS:

1. A method for booting an application from multiple memories, comprising:
executing in place from a first memory a first portion of the application;
loading a second portion of the application from a second memory; and
executing the second portion of the application.

2. The method of claim 1, further comprising:
executing in-place hardware initialization code from the first memory before executing
the first portion of the application.

3. The method of claim 1, the loading further comprising:
loading the second portion of the application into a third memory.

4. The method of claim 3, wherein the first memory comprises a NOR memory, the second
memory comprises a NAND memory, and the third memory comprises RAM.

5. The method of claim 1, wherein the first memory comprises a low-latency, non-volatile
memory and the second memory comprises a longer-latency, non-volatile memory.

6. The method of claim 1, wherein at least some of the executing in place from the first
memory and at least some of the loading occur in parallel.

7. The method of claim 1, wherein the second portion of the application is loaded into the
third memory by the time the first portion of the application has been executed in place.

8. The method of claim 1, the executing the second portion of the application further
comprising:
executing the second portion of the application using at least one of demand paging or
shadow mode.

9. A system, comprising:
a memory; and
at least one processor coupled to the memory and configured to:
execute in place from a first memory a first portion of an application;
load a second portion of the application from a second memory; and
execute the second portion of the application.

10. The system of claim 9, the at least one processor further configured to:
execute in-place hardware initialization code from the first memory before executing the first portion of the application.

11. The system of claim 9, wherein to load the second portion of the application the at least one processor is configured to:
load the second portion of the application into a third memory.

12. The system of claim 11, wherein the first memory comprises a NOR memory, the second memory comprises a NAND memory, and the third memory comprises RAM.

13. The system of claim 9, wherein the first memory comprises a low-latency, non-volatile memory and the second memory comprises a longer-latency, non-volatile memory.

14. The system of claim 9, wherein at least some of the executing in place from the first memory and at least some of the loading occur in parallel.

15. The system of claim 9, wherein the second portion of the application is loaded into the third memory by the time the first portion of the application has been executed in place.

16. The system of claim 9, wherein to execute the second portion of the application the at least one processor is configured to:
execute the second portion of the application using at least one of demand paging or shadow mode.
17. A tangible computer-readable device having instructions stored thereon that, when executed by at least one computing device, causes the at least one computing device to perform operations comprising:

executing in place from a first memory a first portion of an application;
loading a second portion of the application from a second memory; and
executing the second portion of the application.

18. The computer-readable device of claim 17, the loading comprising:

loading the second portion of the application into a third memory.

19. The computer-readable device of claim 17, wherein at least some of the executing in place from the first memory and at least some of the loading occur in parallel.

20. The computer-readable device of claim 17, wherein the second portion of the application is loaded into the third memory by the time the first portion of the application has been executed in place.
Execute in place from a first portion of an application

Load a second portion of the application from a second memory

Execute the second portion of the application

302  304  306
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 9/06, 9/445, 12/02 (2015.01)
CPC - G06F 9/06, 9/44573, 12/0238

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - G06F 9/06, 9/445, 12/02 (2015.01)
CPC - G06F 9/06, 9/44573, 9/5072, 12/0238, 12/0607

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PatSeer (US, EP, WO, JP, DE, GB, CN, FR, KR, ES, AU, IN, CA, INPADOC); ProQuest; IEEE/IEEE Xplore; Google/Google Scholar;
Keywords: execute, place, run, boot, distributed, memory, NOR, NAND, flash, program

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US 2012/0331281 A1 (BORRAS, J et al.) 27 December 2012; Paragraphs [0050], [0062], [0068], [0148], [0152]</td>
<td>1, 3-9, 11-20</td>
</tr>
<tr>
<td>Y</td>
<td>WO 2014/063330 A1 (INTEL CORPORATION) 1 May 2014; Abstract; Page 4, Lines 25-26, Page 5, Line 1; Page 5, Lines 10-18; Page 6, Lines 12-15</td>
<td>2-10</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search: 24 August 2015 (24.08.2015)
Date of mailing of the international search report: 11 Sep 2015

Name and mailing address of the ISA/Authorized officer
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-8300

Form PCT/ISA/210 (second sheet) (January 2015)