

CLAIMS

1. A computer program product for initializing a control utility in a computing environment, said computer program product comprising:

a computer readable storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method comprising:

performing, by a processor, a boot sequence to boot the control utility in a selected architectural mode which is based on a selected architecture, the selected architecture having a first instruction set architecture, and wherein the control utility is written for the selected architecture and configured to be initialized in at least one of a first architectural configuration that initializes in one architecture having a second instruction set architecture different from the first instruction set architecture and switches to the selected architecture for processing, or in a second architectural configuration that initializes and processes in the selected architecture, and wherein the performing the boot sequence comprises:

executing one or more instructions irrespective of whether the processor is to perform initialization based on the first architectural configuration or the second architectural configuration, the one or more instructions common to the one architecture and the selected architecture in that the one or more instructions are defined in the one architecture and the selected architecture, and the executing comprising executing a switch operation to request a switch of architectural modes for the processor regardless of whether the processor is to perform initialization in the one architecture or the selected architecture, wherein the switch operation performs differently based on which architecture, the one architecture or the selected architecture, the processor is to perform initialization, and based on executing the switch operation, the control utility is booted in the selected architectural mode.

2. The computer program product of claim 1, wherein the control utility comprises an operating system.

3. The computer program product of claim 1, wherein the performing the boot sequence comprises: creating a program status word in a format of the one architecture or the selected architecture and setting fields of the program status word, based on whether the processor is to perform initialization in the one architecture or the selected architecture; and preparing to mode switch to execute the switch operation.

4. The computer program product of claim 3, wherein the program status word controls operations of the computing environment and includes fields configured to specify: an access key for storage references, a number of bits to be used for addressing, and a location of a next instruction to be executed.
5. The computer program product of claim 1, wherein the executing the switch operation comprises switching from the one architecture to the selected architecture, wherein processing of the control utility is in the selected architecture.
6. The computer program product of claim 1, wherein the executing the switch operation comprises requesting to switch from the selected architecture to the selected architecture, and wherein the executing the switch operation results in storing status indicating the processor is currently in the selected architectural mode, the status being treated as acceptable to an issuer of the switch operation.
7. The computer program product of claim 6, wherein the switch operation comprises a signal processor instruction.
8. The computer program product of claim 1, wherein the one architecture uses 31-bit addressing and 32-bit general purpose registers, and the selected architecture uses 64-bit addressing and 64-bit general purpose registers.
9. The computer program product of claim 1, wherein the performing the boot sequence comprises determining whether the control utility is written for the selected architecture, and continuing with performing the boot sequence based on the control utility being written for the selected architecture.
10. The computer program product of claim 1, wherein the method further comprises performing a reset to re-boot the control utility.
11. The computer program product of claim 10, wherein the performing the reset comprises setting a program status word to reflect the reset.

12. A computer system for initializing a control utility in a computing environment, said computer system comprising:

a memory; and a processor in communications with the memory, wherein the computer system is configured to perform a method, said method comprising:

performing, by the processor, a boot sequence to boot the control utility in a selected architectural mode which is based on a selected architecture, the selected architecture having a first instruction set architecture, and wherein the control utility is written for the selected architecture and configured to be initialized in at least one of a first architectural configuration that initializes in one architecture having a second instruction set architecture different from the first instruction set architecture and switches to the selected architecture for processing, or in a second architectural configuration that initializes and processes in the selected architecture, and wherein the performing the boot sequence comprises:

executing one or more instructions irrespective of whether the processor is to perform initialization based on the first architectural configuration or the second architectural configuration, the one or more instructions common to the one architecture and the selected architecture in that the one or more instructions are defined in the one architecture and the selected architecture, and the executing comprising executing a switch operation to request a switch of architectural modes for the processor regardless of whether the processor is to perform initialization in the one architecture or the selected architecture, wherein the switch operation performs differently based on which architecture, the one architecture or the selected architecture, the processor is to perform initialization, and based on executing the switch operation, the control utility is booted in the selected architectural mode.

13. The computer system of claim 12, wherein the performing the boot sequence comprises: creating a program status word in a format of the one architecture or the selected architecture and setting fields of the program status word, based on whether the processor is to perform initialization in the one architecture or the selected architecture; and preparing to mode switch to execute the switch operation.

14. The computer system of claim 13, wherein the program status word controls operations of the computing environment and includes fields configured to specify: an access key for storage references, a number of bits to be used for addressing, and a location of a next instruction to be executed.

15. The computer system of claim 12, wherein the executing the switch operation comprises switching from the one architecture to the selected architecture, wherein processing of the control utility is in the selected architecture.
16. The computer system of claim 12, wherein the executing the switch operation comprises requesting to switch from the selected architecture to the selected architecture, and wherein the executing the switch operation results in storing status indicating the processor is currently in the selected architectural mode, the status being treated as acceptable to an issuer of the switch operation.
17. The computer system of claim 12, wherein the one architecture uses 31-bit addressing and 32-bit general purpose registers, and the selected architecture uses 64-bit addressing and 64-bit general purpose registers.
18. The computer system of claim 12, wherein the performing the boot sequence comprises determining whether the control utility is written for the selected architecture, and continuing with performing the boot sequence based on the control utility being written for the selected architecture.
19. The computer system of claim 12, wherein the method further comprises performing a reset to re-boot the control utility.
20. The computer system of claim 12, wherein the switch operation comprises a signal processor instruction.
21. A method of initializing a control utility in a computing environment, said method comprising:
 - performing, by a processor, a boot sequence to boot the control utility in a selected architectural mode which is based on a selected architecture, the selected architecture having a first instruction set architecture, and wherein the control utility is written for the selected architecture and configured to be initialized in at least one of a first architectural configuration that initializes in one architecture having a second instruction set architecture different from the first instruction set architecture and switches to the selected architecture for processing, or

in a second architectural configuration that initializes and processes in the selected architecture, and wherein the performing the boot sequence comprises:

executing one or more instructions irrespective of whether the processor is to perform initialization based on the first architectural configuration or the second architectural configuration, the one or more instructions common to the one architecture and the selected architecture in that the one or more instructions are defined in the one architecture and the selected architecture, and the executing comprising executing a switch operation to request a switch of architectural modes for the processor regardless of whether the processor is to perform initialization in the one architecture or the selected architecture, wherein the switch operation performs differently based on which architecture, the one architecture or the selected architecture, the processor is to perform initialization, and based on executing the switch operation, the control utility is booted in the selected architectural mode.

22. The method of claim 21, wherein the control utility comprises an operating system.

23. The method of claim 21, wherein the performing the boot sequence comprises:

creating a program status word in a format of the one architecture or the selected architecture and setting fields of the program status word, based on whether the processor is to perform initialization in the one architecture or the selected architecture; and
preparing to mode switch to execute the switch operation.

24. The method of claim 23, wherein the program status word controls operations of the computing environment and includes fields configured to specify: an access key for storage references, a number of bits to be used for addressing, and a location of a next instruction to be executed.

25. The method of claim 21, wherein the executing the switch operation comprises switching from the one architecture to the selected architecture, wherein processing of the control utility is in the selected architecture.

26. The method of claim 21, wherein the executing the switch operation comprises requesting to switch from the selected architecture to the selected architecture, and wherein the executing the switch operation results in storing status indicating the processor is currently in the selected architectural mode, the status being treated as acceptable to an issuer of the

switch operation.

27. The method of claim 26, wherein the switch operation comprises a signal processor instruction.

28. The method of claim 21, wherein the one architecture uses 31-bit addressing and 32-bit general purpose registers, and the selected architecture uses 64-bit addressing and 64-bit general purpose registers.

29. The method of claim 21, wherein the performing the boot sequence comprises determining whether the control utility is written for the selected architecture, and continuing with performing the boot sequence based on the control utility being written for the selected architecture.

30. The method of claim 21, further comprising performing a reset to re-boot the control utility.

31. The method of claim 30, wherein the performing the reset comprises setting a program status word to reflect the reset.