



9978 Granite Point Ct.
Granite Bay, CA 95746
www.codesourcery.com

ARM GNU/Linux Application Binary Interface Supplement

The Application Binary Interface (ABI) for the ARM Architecture specifies various aspects of compilation and linkage required for interoperation between toolchains used for the ARM Architecture. However, as the ABI is designed without reference to a particular operating system, there are certain aspects which remain unspecified. This document supplements the ABI for the ARM Architecture by specifying the decisions made for these aspects when using the GNU/Linux operating system.

Copyright © 2005 CodeSourcery, LLC.

1 About This Document

1.1 Change History

Version	Date	Change
1.0	2005-06-28	First public release.

1.2 Scope

The Application Binary Interface (ABI) for the ARM Architecture specifies various aspects of compilation and linkage required for interoperation between toolchains used for the ARM Architecture. However, as the ABI is designed without reference to a particular operating system, there are certain aspects which remain unspecified. This document supplements the ABI for the ARM Architecture by specifying the decisions made for these aspects when using the GNU/Linux operating system.

1.3 Organization

Each section of this document corresponds to one of the ABI documents in version 2.0 of the ABI for the ARM Architecture.

1.4 Omissions

There are unspecified aspects of the ABI for the ARM Architecture that remain unspecified in this document. Future versions of this specification will specify additional aspects.

1.5 Comments

Comments about this document, including corrections or completions, should be sent to arm.eabi@codesourcery.com.

2 Procedure Call Standard for the ARM Architecture (AAPCS)

2.1 Base document information

ARM Document Number GENC-003534

ARM Issue Number 1.0

URL <http://www.arm.com/miscPDFs/8031.pdf>

2.2 Procedure call standard (AAPCS §5)

GNU/Linux uses the Base Procedure Call Standard, as described in §5 of the AAPCS. In particular, GNU/Linux does not use any of the standard variants described in §6 of the AAPCS.

2.3 Use of register `r9` (AAPCS §5.1)

Register `r9` is designated as `v6`. (As a result, subroutines must preserve the contents of `r9`).

2.4 Type of `wchar_t` (AAPCS §7.1.1)

The type of `wchar_t` is `unsigned int`.

2.5 Underlying types for enumerations (AAPCS §7.1.3)

Enumeration types have type `int` or `unsigned int`.

3 ELF For the ARM Architecture (AAELF)

3.1 Base document information

ARM Document Number GENC-003538

ARM Issue Number 1.0

URL <http://www.arm.com/miscPDFs/8030.pdf>

3.2 Meaning of `R_ARM_TARGET1` (AAELF §4.6.1.1)

`R_ARM_TARGET1` is treated identically to `R_ARM_ABS32`. In particular, the calculation performed is $(S + A) | T$.

3.3 Meaning of `R_ARM_TARGET2` (AAELF §4.6.1.1)

`R_ARM_TARGET2` is treated identically to `R_ARM_GOT_PREL`. In particular, the calculation performed is $GOT(S) + A - P$.