ARM结构，一直都在升级，到目前，已经升级到了ARMv8。

ARMv8，提出了很多新的概念。

一、两种执行状态

在ARMv8中，引入了两种执行状态：

AArch32

ARMv7的升级版

A32(ARM)和T32(thumb)，两种指令集

ARMv8架构中，增加了一些指令

传统ARM的特权模式

通用寄存器位宽是32bit

使用单一CPSR保存PE状态

使用32bit的虚拟地址

支持协处理器

AArch64

通用寄存器位宽是64bit

提供64bit PC，SP 和 ELR(exception-link-register)

新的指令集-A64，固定32bit的指令集

新的特权模式

使用一组PSTATE保存PE状态

不支持协处理器

使用64bit虚拟地址

AArch32(简称A32)，兼容以前的arm指令，包括ARM和thumb指令，而AArch64，是全新的指令集，不兼容以前的arm指令。对于A64，使用64bit的虚拟地址，因此支持操作更大的memory空间。而pc也是64bit，因此可以在更大的memory空间上取指令执行。

可以认为AArch64(简称A64)，是全新的ARM指令，和以前ARM指令完全不一样，因此区别也是比较大的。推出全新的指令集，是为了设计出更高性能的CPU。

二、特权模式与安全状态

1、特权模式

A64中，提出了全新的特权模式，不再使用A32中所使用的7种复杂的特权模式。

A64提出了简单的4种特权模式EL(exception level)，分别是EL0，EL1，EL2，EL3。

EL0 ： 运行应用程序

EL1 ： 运行操作系统

EL2 ： 运行虚拟机

EL3 ： 运行安全管理

其中EL3的运行权限最高，EL0的运行权限最低。运行权限，会影响资源的访问。

ARMv8虽然定义了EL2和EL3，但是这两个EL不是一定要实现的。可以根据自己的需求，进行裁剪，或者都实现，或者都不实现，或者只实现一个。但EL0和EL1肯定是要实现的。

对于A32，还是兼容之前的特权模式。而每个特权模式，其实和A64的EL是有对应关系的。

对于A32：

user特权模式，对应于A64的EL0。

supervisor, abort, undef, system, irq, fiq六种特权模式，对应于A64的EL1。

hyp对应于A64的EL2，运行虚拟机。

mon对应于A64的EL3，运行安全管理。

同理，hyp和mon不是一定要实现的。但7种特权模式，肯定是要实现的。

2、安全状态

ARMv8中，引入了两种安全状态：

secure state

non-secure state

这里的安全状态，主要是影响资源的访问，比如memory，系统寄存器等。

对于memory，有时候需要做数据隔离，用于保护数据的安全性。因此就可以将memory分成两个区域，secure区域和non-secure区域。对于secure的memory区域，只允许secure状态去访问，而对于non-secure的memory区域，允许secure状态和non-secure状态都可访问。这样可以保护数据的安全。

对于系统寄存器，有些系统寄存器限制了最低EL的访问，这样可以有效的保护系统。不让低EL运行的程序，误操作这些系统寄存器而使整个系统崩溃掉。

对于EL1和EL0，可以是non-secure状态，也可以是secure状态。

对于EL2，只能是non-secure状态。据说最新的ARMv8.4，EL2可以是secure状态。

对于EL3，只能是secure状态。

secure和non-secure状态的切换，只能通过EL3进行切换，也就是如果想从non-secure的EL0切换到secure的EL0，首先要先从non-secure的EL0，通过异常切换到EL3，然后再通过异常，返回到secure的EL0。

三、A64与A32的切换

对于A32来说，ARM和thumb的切换，通过bx指令即可切换。但是对于A64和A32两种执行状态，只能通过异常，进行切换。

ARMv8对EL切换，进行了以下的限定：

切换到低EL，执行状态要么保持，要么切换到A32

切换到高EL，执行状态要么保持，要么切换到A64

也就是如果当前执行状态是A32，切换到低EL，那么执行状态只能是A32。如果当前执行状态是A64，切换到高EL，那么执行状态只能是A64。

如下图所示，在A64的操作系统中，可以运行A32的应用程序，但是在A32的操作系统中，不能运行A64的应用程序。

如果EL3是A32，那么所有的EL都是A32了。