

# ARM64 架构面试题 参考答案

在芯片被卡脖子，国产芯片大发展的背景下，不了解和熟悉 ARM64 体系结构会被时代浪潮抛弃！

Arm China 教育计划推荐教材，全球首本 ARMv8v9 体系结构编程与实践的新书《ARM64 体系结构：编程与实践》。



在阅读本书之前，请读者尝试完成以下自测题，从而了解自己对 ARM64 架构（ARMv8 和 ARMv9）的掌握程度。一共有 20 道题，每道题 5 分，总分 100 分。

1. A64 指令集支持 64 位宽的数据和地址寻址，为什么指令的编码宽度只有 32 位？

[参考答案] 在第 3.2 章。

2. 下面几条 MOV 指令中，哪些能执行成功？哪些会执行失败？

```
mov x0, 0x1234
mov x0, 0x1abcd
mov x0, 0x12bc0000
mov x0, 0xffff0000ffff
```

[参考答案] 在第 3.6 章，参见例 3-13,3-14 以及 3-15。

3. 在下面的示例代码中，X0 和 X1 寄存器的值分别是多少？

```
string1:
    .string "Booting at EL"
ldr x0, string1
ldr x1, =string1
```

[参考答案] 在第 3.7 章，参见例 3-18。

4. 在下面的示例代码中，X0 寄存器的值是多少？

```
mov x1, #3
mov x2, #1
sbc x0, x1, x2
```

[参考答案] 在第 4.2.6 章，参见例 4-17。

5. 检查数组 `array[0, index-1]` 是否越界需要判断两个条件，一是输入值是否大于或等于 `index`，二是输入值是否小于 0。如下两条指令可实现数组越界检查的功能，其中 X0 寄存器存储了数组的边界 `index`，X1 为输入值 `input`。请解释这两条指令为什么能实现数组越界检查。

```
subs xzr, x1, x0
b.hs OutOfIndex
```

[参考答案] 在第 4.4 章，参见例 4-22。

6. 下面是 `kernel_ventry` 宏的定义。

```
.macro kernel_ventry, el, label
b    el\()\el\()\_label
.endm
```

下面的语句调用 `kernel_ventry` 宏，请解释该宏是如何展开的。

```
kernel_ventry 1, irq
```

[参考答案] 在第 8.4.5 章，参见例 8-13。

7. 关于链接器，请解释：链接地址、虚拟地址以及加载地址。当一个程序的代码段的链接地址与加载地址不一致时，我们应该怎么做才能让程序正确运行？

[参考答案] 在第 9.3.1 章。

8. 在 ARM64 处理器中，异常发生后 CPU 自动做了哪些事情？软件需要做哪些事情？在发生异常后，CPU 是返回发生异常的指令还是下一条指令？什么是中断现场？对于 ARM64 处理器来说，中断现场应该保存哪些内容？中断现场保存到什么地方？

[参考答案]

1. 异常发生之后，CPU 和软件做了哪些事情，参考第 11.2 章。
2. 异常发生后，CPU 返回哪条指令，参考第 11.2.3 章
3. 中断现场，参考第 11.4 章
4. 中断现场保存，参考第 12.4 章

9. 为什么页表要设计成多级页表？直接使用一级页表是否可行？多级页表又引入了什么问题？请简述 ARM64 处理器的 4 级页表的映射过程，假设页面粒度为 4 KB，地址宽度为 48 位。

[参考答案]

1. 一级页表和多级页表问题，参考第 14.1.4 章
2. ARM64 的 4 级页表，参考第 14.2.2 章，图 14.12.

10. ARMv8 体系结构处理器主要提供两种类型的内存属性，分别是普通类型内存（normal

memory) 和设备类型内存 (device memory), 它们之间有什么区别?

[参考答案] 参考第 14.5 章。

11. 在使能 MMU 时, 为什么需要建立恒等映射?

[参考答案] 参考第 14.6 章。

12. 请简述直接映射、全相连映射以及组相连映射的高速缓存的区别。什么是高速缓存的重名问题? 什么是高速缓存的同名问题? VIPT 类型的高速缓存会产生重名问题吗?

[参考答案]

1. 直接映射、全相连映射以及组相连映射, 参考 15.4 章
2. 重名问题和同名问题, 参考第 15.6 章
3. VIPT 重名问题, 参考第 15.6.2 章

13. 在 ARM64 处理器中, 什么是内部共享和外部共享的高速缓存? 什么是 PoU 和 PoC?

[参考答案]

1. 内部共享与外部共享, 参考第 15.8.1 章
2. PoU 和 PoC, 参考第 15.8.2 章。

14. 假设系统中有 4 个 CPU, 每个 CPU 都有各自的一级高速缓存, 处理器内部实现的是 MESI 协议, 它们都想访问相同地址的数据  $a$ , 大小为 64 字节, 这 4 个 CPU 的高速缓存在初始状态下都没有缓存数据  $a$ 。在  $T_0$  时刻, CPU0 访问数据  $a$ 。在  $T_1$  时刻, CPU1 访问数据  $a$ 。在  $T_2$  时刻, CPU2 访问数据  $a$ 。在  $T_3$  时刻, CPU3 想更新数据  $a$  的内容。请依次说明,  $T_0 \sim T_3$  时刻, 4 个 CPU 中高速缓存行的变化情况。

[参考答案] 参考第 16.4.8 章, 图 16.21 – 图 16.24。

15. DMA 缓冲区和高速缓存容易产生缓存一致性问题。从 DMA 缓冲区向设备的 FIFO 缓冲区搬运数据时, 应该如何保证缓存一致性? 从设备的 FIFO 缓冲区向 DMA 缓冲区搬运数据时, 应该如何保证缓存一致性?

[参考答案] 参考第 16.8 章。

16. 为什么操作系统在切换 (或修改) 页表项时需要先刷新对应的 TLB 表项后切换页表项?

[参考答案] 参考第 17.5.4 章。

17. 下面是关于无效指令高速缓存的代码片段, 请解释为什么在使指令高速缓存失效之后要发送一个 IPI, 而且这个 IPI 的回调函数还是空的。

```
void flush_icache_range(unsigned long start, unsigned long end)
{
    flush_icache_range(start, end);
    smp_call_function(do_nothing, NULL, 1);
}
```

[参考答案] 参考第 18.3.8 章。

18. 假设在下面的执行序列中，CPU0 先执行了  $a=1$  和  $b=1$ ，接着 CPU1 一直循环判断  $b$  是否等于 1，如果等于 1 则跳出 while 循环，最后执行 “assert ( $a == 1$ )” 语句来判断  $a$  是否等于 1，那么 assert 语句有可能会出错吗？

CPU0	CPU1
-----	
<pre>void func0() {     a = 1;     b = 1; }</pre>	<pre>void func1() {     while (b == 0) continue;     assert (a == 1) }</pre>

[参考答案] 参考第 19.1 和 19.2 章。

19. 假设 CPU0 使用 LDRXB/STXRB 指令对 0x341B0 地址进行独占访问操作，CPU1 也使用 LDRXB/STXRB 指令对 0x341B4 地址进行独占读操作，CPU1 能成功独占访问吗？

[参考答案] 参考第 20.3.3 章。

20. 假设函数调用关系为  $\text{main}() \rightarrow \text{func1}() \rightarrow \text{func2}()$ ，请画出 ARM64 体系结构的函数栈的布局。

[参考答案] 参考第 21.3 章。

订阅配套视频课程:

淘宝: <https://shop115683645.taobao.com/>

微店: <https://weidian.com/?userid=1692165428>

## 第3季旗舰篇



# ARM64体系结构与编程

armv8v9芯片手册导读+深入理解+动手实验+项目实验总结

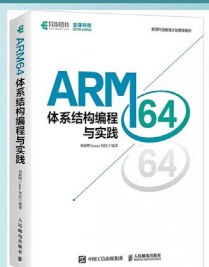
### 您将学到

- √熟悉armv8和armv9指令集与汇编
- √深入解读armv8和armv9体系结构
- √armv8与armv9重点与难点
- √几十个深入的动手实验
- √项目经验陷阱与坑
- √动手实践BenOS
- √树莓派4B+arm仿真器调试技巧

主讲人 笨叔

笨叔叔带队◇以练带学

几十个实验◇等您来撸



**奔跑吧Linux社区**  
**第三季**  
**ARM64体系结构与编程**  
**课程目录**

课程名称	进度	时长（分钟）
第1课：课程介绍	完成	30
第2课：ARM处理器介绍和ARMv8概述	完成	75
第3课：使用QEMU+树莓派4+JLINK搭建调试环境	完成	40
第4课：ARM64汇编指令之存储指令	完成	96
第5课：ARM64汇编指令之算术和移位指令	完成	56
第6课：ARM64汇编指令之比较和跳转指令	完成	49
第7课：ARM64汇编指令之其他重要指令	完成	63
第8课：总结 - 那些年我们踩过的汇编坑	完成	43
第9课：GNU AS汇编器介绍	完成	46
第10课：LD链接器介绍	完成	55
第11课：内嵌汇编	完成	68
第12课：ARM64异常处理	完成	115
第13课：ARM64异常处理之中断处理	完成	64
第14课：GIC中断控制器	完成	74
第15课：ARMv8内存管理	完成	97
第16课：MMU实验讲解	完成	122
第17课：MMU芯片手册解读	完成	97
第18课：cache和TLB基础知识	完成	80
第19课：cache一致性-part1	完成	103
第20课：解读armv8芯片手册之cache	完成	98
第21课：cache实验详解	完成	34
第22课：TLB基础知识	完成	39
第23课：内存屏障基础知识	完成	66
第24课：解读armv8手册中的内存屏障	完成	64
第25课：缓存一致性与内存屏障	完成	58
第26课：原子操作	完成	63
第27课：浮点数指令	完成	36
第28课：NEON矢量优化	完成	88
第29课：可扩展矢量计算SVE/SVE2	完成	100
第30课：GICv3中断控制器与ITS服务	完成	105
第31课：深入理解SMMUv3与IOMMU驱动	完成	76
第32课：SMMU之SVA	完成	45
第33课：深入理解AXI5以及AXI5-Lite总线	完成	78
第34课：深入理解ACE以及ACE-Lite总线	完成	90
第35课：深入理解CHI总线	完成	82
		总计 <b>42</b> 小时

更多精彩内容马上献上....  
微信公众号：runninglinuxkernel

订购热线：  
<https://shop115683645.taobao.com/>  
<https://weidian.com/?userid=1692165428>