

ARM64 体系结构自测题

在阅读本书之前，请读者尝试做一下这份 ARM64 架构自测题目（模拟面试题目），从而掌握 ARM64 架构的了解程度。

下面一共有 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？
2. 下面几条 mov 指令，哪些能成功？哪些会失败？

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

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
```

4. 在下面示例代码中，X0 寄存器的值是多少？

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

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等于 `index`，二是输入值是否小于 0。如下两条指令来实现数组边界检查的功能，其中 X0 为数组的边界 `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 OutOfInex
```

6. 下面是 `kernel_ventry` 宏定义。

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

下面语句调用 `kernel_ventry` 宏，请解释该宏是如何展开的。

```
kernel_ventry 1, irq
```

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？

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？异常返回时，它是返回到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？

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 4KB，地址宽度为 48 位。

10. ARMv8 架构处理器主要提供两种类型的内存属性，分别是普通类型（normal memory）内存和设备类型（device memory）内存，它们之间有什么区别？

11. 在使能 MMU 时，为什么需要建立恒等映射？

12. 请简述直接映射、全映射以及组相联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？

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？

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 T0 时刻，CPU0 访问数据 A。在 T1 时刻，CPU1 访问数据 A。在 T2 时刻，CPU2 访问数据 A。在 T3 时刻，CPU3 想更新数据 A 的内容。请依次说明，T0~T3 时刻，4 个 CPU 中高速缓存行的变化情况。

15. DMA 和高速缓存容易产生缓存一致性问题。

从 DMA 缓冲区（内存）到设备 FIFO 搬运数据时，应该如何保证缓存一致性？

从设备 FIFO 到 DMA 缓冲区（内存）搬运数据时，应该如何保证缓存一致性？

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 表项然后再切换页表项？

17. 下面是一个无效指令高速缓存的代码片段，请解释为什么在无效指令高速缓存之后要发送一个 IPI 中断，而且这个 IPI 中断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);
}
```

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 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 语句有可能会失败吗？

CPU0	CPU1

void func0() { a = 1; b = 1; }	void func1() { while (b == 0) continue; assert (a == 1) }

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 地址进行独占地读操作，CPU1 能成功独占访问吗？

20. 假设函数调用关系为 main()->func1()->func2()，请画出 ARM64 架构的函数栈的布局。

以上题目的答案都分布在本书的各章中。