

冯·诺依曼结构和哈佛结构有什么异同？

共同点：使用两个独立的存储器模块，分别存储指令和数据，每个存储模块都不允许指令和数据并存。

区别：

1、性质不同：冯·诺依曼体系结构冯·诺伊曼理论的要点是计算机的数制采用二进制，计算机应该按照程序顺序执行。哈佛结构是一种将程序指令存储和数据存储分开的存储器结构。

2、特点不同：哈佛结构将程序和数据存储在不同的存储空间中，即程序存储器和数据存储器是两个独立的存储器，每个存储器独立编址、独立访问。诺依曼结构的计算机运行过程中，把要执行的程序和处理的数据首先存入主存储器，计算机执行程序时，将自动地并按顺序从主存储器中取出指令一条一条地执行。

3、组成不同：哈佛结构的计算机由 CPU、程序存储器和数据存储器组成。诺依曼结构指令由操作码和地址码组成。

进程 0 和进程 1

所有进程的祖先叫做进程 0 (idle 进程或者 swapper 进程)，它是在 Linux 的初始化阶段从无到有的创建的内核进程（该进程描述符使用的是静态分配的数据结构，其他进程都是动态分配的）。

进程 0 调用 start_kernel()函数初始化内核需要的所有数据结构，激活中断，这样就可以创建一个进程 1 (init 进程) 的内核进程，进程 0 和进程 1 共享每进程的所有内核数据结构。

多处理器中，每个 CPU 都有自己的进程 0。

进程 0 创建的内核进程执行 init()函数初始化内核。然后调用 execve()系统调用装入可执行程序 init。至此，init 为一个普通的进程，且拥有每进程内核数据结构。在系统关闭之前，init 一直存在，它创建和监控在操作系统外层的执行的所有进程的活动。

这样*进程调度算法*总能成功的选择要执行的进程。总是至少有一个进程。即 swapper 进程。PID 等于 0。只有当没有其他进程处于 TASK_RUNNING 状态下，调度程序才选择进程 0。

ARM64 跟 x64 有什么区别

##

x64 一般指 x86_64, 64 位 x86 架构处理器, x64 是 x86 系列中的 64 位体系; ARM64 指的是 64 位 ARM 架构处理器, ARM64 是 ARM 中 64 位体系结构。

指令集不同, 属于两种不同的体系。: ARM 属于精简指令集体系, 汇编指令比较简单。x86 属于复杂指令集体系, 汇编指令较多。

##

先说结论:

ARM64 是全新指令集 (arm 设计);

x64 是 x86_64 (Intel 设计) 和 AMD64 (AMD 设计) 的简称。

再上公式:

ARM64 = AArch64

x64 = amd64 = x86_64 != IA-64

IA 是 Intel Architecture 英特尔架构的缩写

时间线:

x86(Intel, 1978.6.8) => x64/AMD64(AMD 设计 1999) => IA-64(Intel 设计 2001)
=> x86_64(AMD 设计 2003 年 4 月)

进程和作业的区别与联系

作业与进程的区别:

一个进程是一个程序对某个数据集的执行过程, 是分配资源的基本单位。作业是用户需要计算机完成的某项任务, 是要求计算机所做工作的集合。一个作业的完成要经过作业提交、作业收容、作业执行和作业完成 4 个阶段。而进程是对已提交完毕的程序所执行过程的描述, 是资源分配的基本单位。其主要区别如下。

(1) 作业是用户向计算机提交任务的任务实体。在用户向计算机提交作业后, 系统将它放入外存中的作业等待队列中等待执行。而进程则是完成用户任务的执行实体, 是向系统申请分配资源的基本单位。任一进程, 只要它被创建, 总有相应的部分存在于内存中。

(2) 一个作业可由多个进程组成, 且必须至少由一个进程组成, 反过来则不成立。

(3) 作业的概念主要用在批处理系统中, 像 UNIX 这样的分时系统中就没有作业的概念。而进程的概念则用在几乎所有的多道程序系统中。

作业、进程和程序之间的联系

一个作业通常包括程序、数据和操作说明书 3 部分。每一个进程由 PCB、程序和数据集合组成。这说明程序是进程的一部分, 是进程的实体。因此, 一个作

业可划分为若干个进程来完成，而每一个进程有其实体——程序和数据集合。

协程 Coroutine

协程不是进程或线程，其执行过程更类似于子例程，或者说不带返回值的函数调用。

页和页框的区别是什么？

页（page）

一页指一系列的线性地址和包含于其中的数据

页框(page frame)

分页单元认为所有的 RAM 被分成了固定长度的页框

每个页框可以包含一页，也就是说一个页框的长度和一个页的长度是一样的
页框是内存的一部分，是一个实际的存储区域。

页只是一组数据块，可以存放在任何页框中

也就是说：

页对应的是线性地址

而页框对应的是物理地址，是实际的存储区域

大小核（big.LITTLE）架构

big.LITTLE 处理解决了当今行业面临的一个难题：如何创建既有高性能又有极佳节能效果的片上系统 (SoC) 以延长电池使用寿命。

big.LITTLE 将 ARM Cortex-A15 MPCore 处理器的性能与 Cortex-A7 处理器的节能效果结合在一起，使同一应用程序软件在二者之间无缝切换。通过为每个任务选择最佳处理器，big.LITTLE 可以使电池的使用寿命延长高达 70%。

big.LITTLE 处理的设计旨在为适当的作业分配恰当的处理器。Cortex-A15 处理器是目前已开发的性能最高的低功耗 ARM 处理器，而 Cortex-A7 处理器是目前已开发的最节能的 ARM 应用程序处理器。可以利用 Cortex-A15 处理器的性能来承担繁重的工作负载，而 Cortex-A7 可以最有效地处理智能手机的大部分工作负载。这些操作包括操作系统活动、用户界面和其他持续运行、始终连接的任务。

big.Little 系统中两个处理器之间高效无缝地切换工作负载是通过开发高级 ARM 系统 IP 实现的，这样可确保 Cortex-A15 和 Cortex-A7 处理器之间完全的高速缓存、I/O 的一致性。

那么，到底什么是 big.LITTLE，它又是如何进行工作的呢？

为此，ARM 专门录制了一个视频，深入浅出地做出解释，并用两个人来分别代表 big.LITTLE 架构中的“角色”，身材魁梧高大的男士是 big，身材娇小的是 LITTLE，感兴趣的读者可以观看这个视频，搜索关键字：“ARM+big.LITTLE”即可，下图为该视频的截图。

当手机不需要工作时，big 核心和 LITTLE 核心都可以停下来休息。

“big.LITTLE 是一种节能省耗技术，最高性能的 ARM CPU 核心与最高效的 ARM CPU 核心相结合，可以以更低的功耗提供最好的工作性能，最快的处理任务速度。”——ARM 解释道。

基于 big.LITTLE 技术的八核处理器，并没有将传统内核放在单一的处理器的上，而是一分为二，其中一个使用了 4 个“小核心”，另一个则使用了 4 个“大核心”，这两个“核心”都有着自己独立的速度和性能。通过两大核心自主运行，搭载 big.LITTLE 技术的处理器比之前的手机 CPU 更加高效，毕竟后者只有一个或者两个内核。

当需要用智能手机打开一个网页时，手机就可以用一个大的内核来处理该任务，而小的内核则同时处理其他小任务，比如查看电子邮件、拨打电话等。

big.LITTLE 软件与平台能够节省 75% 的 CPU 功耗，同时在线程负载方面提升 40% 的性能。根据任务所需的性能，big.LITTLE 软件和 big.LITTLE MP 自动和无缝的将工作负载转至适合的 CPU 核心进行任务处理。ARM big.LITTLE 技术同时兼顾了最佳性能和最低功耗，减少手机的电量消耗。

智能手机与平板电脑对性能的要求越来越高，与此同时，电池的技术发展缓慢。人们希望手机或平板电脑有足够高的性能，处理任务的速度足够快，同时又希望设备电池续航更长一些。传统的处理器技术已经无法满足用户的这种需求，因此，ARM 不得不设计出创新的 big.LITTLE 大小架构。

big.LITTLE 是 ARM 的诸多电力系统管理技术之一，它与动态电压和频率调节(DVFS)、时钟门控以及温度管理协调工作，全面控制着片上处理器的功耗。2015 年最佳智能手机阵容处理器均基于 ARM 的 big.LITTLE 架构，采用该架构处理器的手机工作速度更快更高效。三星 Galaxy S6、HTC M9、LG G4 等手机均采用基于 big.LITTLE 的处理器。实际上，在 ARM 提出大小核之前，nVIDIA 的 Tegra 3 已经有这个意思了。

Tegra 3 包括四个高性能的 A9 核心（相当于大核心），和一个性能较低的，采用低功耗设计的 A9 伴核（小核）。当然那时候的设计还没有 big.LITTLE 完善，也没有特殊设计的缓存一致性互联，切换的时间也较长，达到了 ms 级别。

12 代酷睿将使用 big.LITTLE 大小核架构

根据英特尔的路线图，2022 年将会推出代号 Alder Lake 的第十二代酷睿处理器，制程将升级到 10nm。更重要的是这一代酷睿处理器将升级到 16 核，同时支持 PCIe 4.0 通道；至于 DDR5 内存，目前还没有确切消息证实。

Product	TDP (W)	Socket
Alder Lake S (8+8+1) ²	125 ¹	LGA 1700
Alder Lake S (8+8+1)	80	LGA 1700
Alder Lake S (6+0+1) ³	80	LGA 1700

Notes:

¹ Investigating performance scaling upto 150W

² 8+8+1: 8 big core, 8 small core, GT1

³ 6+0+1: 6 big core, 0 small core, GT1

VC 网站得到的最新资料显示，英特尔的 16 核并不是传统意义上的 16 颗大核心，而是采用 ARM 处理器上常见的 8 颗大核+8 颗小核的结构。这种结构的设计很容易让人想到 ARM 公司在 Cortex-A 系列处理器使用的 big.LITTLE 大小核架构，将高性能核心与低功耗核心混合使用，得到最佳的处理器能效。

事实上，英特尔已经在 Lakefield 上进行尝试，利用 3D 封装技术将 4 小核和 1 大核封装在一起，大核采用 Sunny Cove 架构，小核则是 Atom 产品线的 Tremont 核心。如果英特尔在桌面级的 Alder Lake 用上使用 8+8 架构，正面英特尔已经有信心在高性能产品上使用 3D 封装技术，而不是现在的低功耗产品上。

RUMORED INTEL Mainstream CPU Series Specs				
VideoCardz.com	Coffee Lake-S	Comet Lake-S	Rocket Lake-S	Alder Lake-S
CPU Fabrication Node	14nm	14nm	14nm	10nm
Max Core Count	up to 8 cores	up to 10 cores	up to 8 cores	up to 16 (8 BIG + 8 SMALL)
Socket	LGA1151	LGA1200	LGA1200 (?)	LGA1700 (?)
Memory Support	DDR4	DDR4	DDR4	?
PCI-E Support	PCIe 3.0	PCIe 3.0	PCIe 4.0 (?)	PCIe 4.0 (?)
Intel Core Series	8th/9th Gen Core-S	10th Gen Core-S	11th Gen Core-S (?)	12th Gen Core-S (?)
Motherboard Chipsets	Intel 300 (eg. Z390)	Intel 400 (eg. Z490)	Intel 500 (?)	Intel 600 (eg. Z690) (?)
Launch	2018	2020	?	?

但奇怪的是 Alder Lake 处理器使用 GT1 核显，目前的产品至少是 GT2 级别产品，英特尔反而在更新一代产品中弱化核显规模。Alder Lake 处理器的热设计功耗会增加，普通版提高至 80W、高端 125W。

另外，爆料显示英特尔还在研究扩展 TDP 的方法，尝试做到 150W，越高的 TDP 意味着 CPU 频率、性能会越高。

英特尔还将在第十二代酷睿处理器上更换插槽，LGA1700 插槽意味着处理器的核心规模将有大幅提升。LGA1200 插槽寿命只会持续 2 代，主要用于 14nm Comet Lake 及 14nm Rocket Lake。