《openEuler操作系统》

测试题

**一、判断题**

1. openEuler是一个开源操作系统及其发行平台，根据用户不同需求进行免费/收费的服务。（F）

题解：openEuler是一个开源、免费的Linux发行版平台。

2. openEuler操作系统的前身是EulerOS。（T）

题解：openEuler的前身是运行在华为公司通用服务器上的操作系统EulerOS。

3. openEuler操作系统仅支持ARM处理器架构。（F）

题解：openEuler支持x86、ARM、RISC-V等多种处理器架构。

4. openEuler是一款通用的Linux操作系统。（T）

5. openEuler基于鲲鹏处理器做了增强。（T）

6. openEuler是与鲲鹏芯片配套的软件基础设施。（T）

题解：openEuler及其前身都是与鲲鹏芯片配套的软件基础设施。

7. openEuler提供一种NUMA aware的解决方案，提升多核调度性能。（T）

8. openEuler提供了iSulad轻量级容器全场景解决方案。（T）

9. 鲲鹏处理器是华为基于ARMv8架构开发的通用处理器。（T）

10. 鲲鹏处理器支持了CPU核虚拟化、内存虚拟化和中断虚拟化。（T）

11. 在NUMA中，openEuler采用Compact NUMA-Aware Lock（CNA）队列代替了Qspinlock中的MCS队列。（T）

12. openEuler支持POSIX和System V共享内存，仅支持POSIX消息传递机制。（F）

题解：openEuler支持POSIX和System V两种消息传递机制。

13. 鲲鹏920 SoC上已经集成了网络控制器和PCIe控制器。（T）

14. 2019年，轻量级容器引擎iSulad发布，并集成于openEuler中，其具有轻、快、易、灵的特点。（T）

15. openEuler的自动调优工具整体上是一个C/S架构。（T）

**二、填空题**

1. openEuler提供鲲鹏加速引擎插件使能鲲鹏硬件加速能力。

题解：鲲鹏加速引擎、Kunpeng Accelerator Engine(KAE)均可。

2. openEuler提供的操作系统配置参数智能优化引擎是A-tune。

3. openEuler中线程的实现采用的是内核级线程模型。

4. 为了解决单队列调度策略缺乏可扩展性和违背亲和性的问题，openEuler使用了多队列调度策略。

5. openEuler操作系统中的智能调优工具A-Tune包含了两个核心模块：智能决策模块和自动调优模块。

**三、单项选择题**

1. openEuler源于：（A）

A. 华为的高性能计算项目

B. 华为的openGauss数据库项目

C. 华为的云计算项目

D. 华为的鸿蒙操作系统项目

2. 关于openEuler操作系统，下列说法正确的是：（B）

A. openEuler的前身即是鸿蒙操作系统。

B. openEuler于2019年底开源。

C. 不允许基于openEuler社区版本发布自己二次开发的操作系统版本。

D. 商业组织不被允许免费使用openEuler社区版本。

3. 下列有关openEuler进程/线程间通信表述错误的是：（C）

A. openEuler提供了NUMA感知队列自旋锁实现互斥机制，减小了NUMA体系结构中使用自旋锁的开销。

B. openEuler提供了down原语与up原语实现线程的同步运行。

C. openEuler进程间通讯机制仅支持消息传递机制。

题解：(C) openEuler支持了共享内存与消息传递进程间通信机制。

4. 下列关于内存顺序模型叙述错误的是：（B）

A. 绝对顺序模型禁止所有优化导致的乱序执行，所有内存访问都将串行排队执行。

B. 强内存顺序模型以x86为代表，只允许load-load（即读-读）指令乱序执行。

C. 弱内存顺序模型以ARM为代表，允许所有情况下（load-load、store-store、load-store、store-load）的指令乱序执行。

题解：(B) 强内存顺序模型以x86为代表，只允许store-load指令（即先执行store指令，再执行load指令）乱序执行。

5. 关于不同架构间一直代码时需要注意的访存问题，下列表述错误的是：（A）

A. 虽然x86硬件会保证load-load、store-store、load-store类指令的执行顺序，但是为保险起见，在这些指令后添加内存屏障指令也是必须的。

B. 从x86向ARM移植代码时需要注意添加内存屏障以避免弱内存顺序模型导致指令乱序执行可能产生的问题。

C. 在不同CPU架构之间移植代码时，需要注意内存顺序模型的异同以保证移植后的程序正常运行，通常的解决方法是在所有需要插入内存屏障的位置调用内存屏障接口。

题解：(A) x86硬件会保证load-load、store-store、load-store类指令的执行顺序，因此不需要软件在这些指令后添加内存屏障指令。

6. 下列关于多核CPU场景下程序执行顺序，叙述错误的是：（C）

A. 多核CPU的发展使得程序的执行顺序可能与代码的编写顺序不符，操作系统使用内存屏障和特定的内存顺序模型来解决这个问题。

B. openEuler提供了由内存屏障指令封装的相关内存屏障函数，可以通过使用相关函数来保证程序语句执行顺序的正确性。

C. openEuler仅仅封装了鲲鹏处理器的数据内存屏障（DMB）指令和数据同步屏障（DSB）指令。

题解：(C) openEuler对鲲鹏处理器的3种屏障指令：数据内存屏障（DMB）指令、数据同步屏障（DSB）指令和指令同步屏障（ISB）均进行了封装。

**四、多项选择题**

1. 为了充分发挥鲲鹏处理器的优势，openEuler在以下几个方面做了增强：（ABCD）

A. 多核调度技术

B. 软硬件协同

C. 轻量级虚拟化

D. 智能优化引擎

2. 关于64核鲲鹏920处理器的物理结构，下列说法正确的有：（ABCD）

A. 是一个片上系统（Sysem on Chip），除了2个CPU DIE，还有1个I/O DIE。

B. 在一个CPU DIE中有8个Cluster，其中包含了4个Core。

C. CPU DIE、I/O DIE和DDRC等模块之间通过Advanced Microcontroller Bus Architecture总线进行互联。

D. I/O DIE支持PCIe总线及高速网卡。

3. 以下关于华为鲲鹏920处理器的描述哪些是正确的？（ABCD）

A. 采用了7nm的制造工艺。

B. 支持8通道的DDR4控制器。

C. 支持PCIe 4.0接口，并兼容PCIe 3.0/2.0/1.0。

D. 支持多种加速器。

4. 华为鲲鹏920处理器内置了那些加速器？（ABC）

A. SSL加速引擎

B. 加解密加速引擎

C. 压缩解压缩加速引擎

5. 关于鲲鹏920的存储子系统，下列那些说法是正确的？（ABCD）

A. 具有L1、L2、L3共3级Cache。

B. L1I（L1指令Cache）和L1D（L1数据Cache）容量均为64KB。

C. 每个CPU核独享L1 Cache和L2 Cache。

D. L3 Cache分Tag和Data两部分，前者用作内容的索引，由一个CPU Core Cluster内的4个核共享，后者由一个CPU DIE内的核共享。

6. 关于鲲鹏处理器架构，下列说法正确的有：（ABCD）

A. 基于ARMv8架构开发。

B. 采用Reduced Instruction Set Computers（RISC）指令集。

C. 支持64位指令集。

D. 大量使用寄存器。

7. 关于openEuler中进程被创建的先后顺序，下列书法正确的是：（ACDE）

A. openEuler启动后，内核用静态数据init\_task创建第一个进程（PID为0）以完成内核初始化工作。

B. 0号进程调用kernel\_thread()创建1号进程以完成剩下的系统初始化工作，之后1号进程创建2号进程。

C. 0号进程会演变为idle进程。

D. 1号进程会执行/sbin/init程序，初始化用户空间，成为init进程，运行在用户态下。

E. init进程就是其后操作系统中所有用户进程的共同祖先。

F. 2号进程对所有内核线程进行管理和调度，偶尔也会运行于用户空间中。

题解：(B) 0号进程调用kernel\_thread()创建1号进程和2号进程。(F) 2号进程又被称为kthreadd内核线程，一直运行于内核空间，对之后所有内核线程进行管理和调度。

8. 下列关于openEuler线程的生命周期说法错误的是：（F）

A. 用户通过调用pthread\_create()函数创建一个线程。

B. 线程被创建完成后即进入就绪状态。

C. 就绪状态的线程被操作系统调度执行，即进入运行状态。

D. 运行状态的线程因CPU被抢占或主动让出CPU会回到就绪状态。

E. 运行状态的线程会因为调用pthread\_join()函数、sleep()函数或I/O操作进入阻塞状态。

F. 当解除阻塞的条件满足时，被阻塞的线程回到执行状态。

题解：(F) 当解除阻塞的条件满足时，被阻塞的线程回到就绪状态。

9. 下列关于openEuler的CPU调度策略，说法不正确的有（B）

A. openEuler中存在多种类别的进程（限期进程、实时进程和普通进程），并实现了多种调度策略。

B. 限期进程对应是时限期调度策略，实时进程只对应轮转调度策略，普通进程对应的是标准轮流分时调度策略。

C. 标准轮流分时调度策略采用的是Completely Fair Scheduler（CFS）算法。

D. CFS调度算法使用了时间片和优先级概念，并引入了虚拟运行时间。

E. openEuler将CFS等调度算法与多队列调度策略进行了融合。

题解：(B) 实时进程对应两种调度策略：先进先出调度策略和轮转调度策略。

10. 下列关于openEuler内存管理表述正确的是：（ABC）

A. 在open Euler中，各级页表的表项大小为8B。

B. openEuler将标准大页封装为一个伪文件系统（hugetlbfs）提供给用户程序申请并访问。

C. openEuler采用Least Recently Used（LRU）策略实现页选择。

11. 对于下面所示程序的打印结果，表述不正确的有：（B）

初始状态：a = 0, b = 0

CPU0 CPU1

1. a = 1 b = 1

2. print b print a

A. 在绝对顺序模型下不会出现（0, 0）打印结果。

B. 以x86为代表的强内存顺序模型，不会打印出（0, 0）的结果。

C. 在弱内存顺序模型下，（0, 0）、（0, 1）、（1, 0）、（1, 1）四种打印结果都有可能出现。

题解：以x86为代表的强内存顺序模型页不会保证store-load顺序，所以四种情况都有可能出现。

12. 对于openEuler中的文件系统，下列叙述正确的有：（AB）

A. openEuler中的进程只与虚拟文件系统（VFS）交互。

B. openEuler中的Virtual File System（VFS）充当各类物理文件系统的管理者。

C. openEuler默认采用NTFS文件系统作为实现层的物理文件系统。

题解：(C) openEuler默认采用Fourth Extended File system（Ext4）文件系统作为实现层的物理文件系统。

13. 下列关于openEuler中虚拟化的叙述不正确的有：（C）

A. 对于CPU虚拟化，openEuler采用的方案是KVM结合QEMU一起模拟，因此，openEuler实现中提到的VMM就是KVM与QEMU的结合。

B. 鲲鹏处理器采用Address Space ID和Virtual Machine ID相结合的设计缓解了进程和虚拟机上下文切换导致的Translation Lookaside Buffer命中率降低问题。

C. 为提高效率，openEuler将I/O虚拟化集成到了KVM中。

题解：(C) openEuler中I/O设备的虚拟化是由QEMU实现的，以此辅助KVM完成整个虚拟化过程。

14. 关于轻量级容器引擎iSulad，下列说法正确的有：（AB）

A. iSulad在设计之初就考虑到了云计算和IoT场景，为解决通信和信息技术领域的不同需求提供了统一架构设计。

B. iSulad采用木兰开源许可，目前已经是openEuler发行版（openEuler 20.03 LTS）的默认容器引擎。

C. 在openEuler 20.03 LTS中，iSulad的关键组件采用Rust语言开发，相比于docker和containerd等其他容器引擎，iSulad的内存开销更小，并发启动性能更高。

D. 在使用习惯上，iSulad兼容docker大部分命令，可以像docker一样管理容器和容器镜像。

题解：(C) 在openEuler 20.03 LTS中，iSulad的关键组件采用C/C++语言开发。

15. 下列关于iSulad架构叙述正确的是：（ABC）

A. iSulad支持RESTful接口和gRPC接口，兼容CRI接口。

B. iSulad支持OCI标准镜像，提供content/metadata、rootfs及snapshot管理能力。

C. iSulad支持轻量级Runtime（lcr）和OCI标准的Runtime（runc，kata等）。

D. 其北向接口仅支持CRI。

题解：(D) iSulad的北向接口提供CLI（Command Line Interface）和CRI。

**五、问答题**

1. 在CPU调度中，openEuler为什么没有采用最短进程优先（Shortest Job First, SJF）算法？

参考答案：

基于以下两个原因，openEuler没有采用SJF算法：(1) SJF算法会导致运行时间长的进程响应时间越来越长，从而产生进程饥饿现象；(2) 因为很难预估每个进程的运行时间，SJF算法在实现上存在难点。

2. 简述openEuler在CPU调度时是如何把先进先出、轮转调度和优先级调度结合起来的？

参考答案：

(1) openEuler为每个优先级维护一个进程链表，调度时，从优先级最高的链表中选择一个进程占用CPU；如果优先级最高的链表中没有进程，调度程序从次高优先级的链表中选择一个进程占用CPU，以此类推。(2) 同时，openEuler的每个进程都有一个调度策略：a. 如果调度程序选择的进程使用FIFO调度策略，那么该进程会一直霸占CPU直至运行完成或者被更高优先级的进程抢占或者该进程自己发生阻塞；b. 如果被选择的进程使用Round-Robin（RR）调度策略，则其将运行指定的时间片，除非发生阻塞或提前运行结束。如果该进程用完时间片后还不能结束，调度程序会把它添加到进程优先级对应的链表尾部，然后把CPU让给优先级相同的其他进程。

3. 简述鲲鹏处理器提供的屏障指令。

参考答案：

鲲鹏处理器提供的屏障指令有以下3种：(1) Data Memory Barrier（DMB）数据内存屏障指令。该指令保证DMB之前的内存访问都完成后，指令DMB之后的内存访问指令才开始访问内存，其他非内存访问指令仍然可以乱序执行。(2) Data Synchronization Barrier（DSB）数据同步屏障指令。该指令保证DSB之前的内存访问、缓存维护指令和页表缓存维护指令完成之后，DSB之后的任何指令才能开始执行。该指令比DMB指令严格，相较于后者会牺牲更多性能。(3) Instruction Synchronization Barrier（ISB）指令同步屏障指令。该指令保证ISB之前的命令全都执行完成之后，ISB之后的指令才会重新从cache或内存取指。ISB指令通常用于发生异常、异常返回或者更改系统配置寄存器的场景下。

4. 简述openEuler推出面向计算机系统性能的自动调优工具A-Tune的原因。

参考答案：

操作系统提供软硬件资源管理，并为应用程序的执行提供受保护的环境，绝大多数的操作系统（如Linux和Windows）在设计时需要权衡使用场景，确保当前设计对大部分通用场景都是有益的.当某个功能机制无法保证对所有场景均有益时，设计者就会在系统中提供一个可配置参数，并确保该参数的默认配置对大部分通用场景有益，而使用者通过更改参数配置来满足特定的使用场景需求。这种设计带来的问题也就显而易见，对于不同的硬件和不同的应用，使用默认参数配置只能保证整个系统勉强可用，无法充分发挥软硬件的性能。

并且，系统调优一直是一个门槛很高的系统性工程，高度依赖工程师的技能和经验。如何在众多因素中找到性能瓶颈，需要工程师们熟悉大量参数的含义、配置方法以及业务场景，并不断积累经验，才能对系统进行快速精准调优。

openEuler基于Linux内核，Linux内核是一个面向通用场景设计的宏内核。在openEuler中，仅sysctl命令（用于运行时配置内核参数的命令）的参数（sysctl -a | wc -l）就超过1000个。在一个完整的计算机系统中，从CPU、加速器、网卡，到编译器、操作系统、中间件框架，再到上层应用，可调节对象超过7000个。此外，不同参数的调节空间也不同。有些参数只是功能的开关，例如，/proc/sys/kernel/numa\_balancing（启用自动NUMA平衡的配置参数）只有0和1两个可选配置，而有些参数则是一个很大连续区间，例如，/proc/sys/net/core/wmem\_max（最大的TCP数据接收窗口参数），这类参数的调节则需要大量验证。而且，参数对系统性能的影响也各不相同，有些参数对系统性能可能是没有影响的，有些参数对系统性能具有很大的影响，在可以影响系统性能的参数中，每个参数对系统性能的影响效果又不同，不同参数的调节甚至会相互影响。在这样的情况下，让系统工程师或运维人员去做系统调优是非常困难和耗时的。

面对上述这些问题，openEuler推出了面向计算机系统性能的自动调优工具A-Tune，旨在让操作系统能够满足不同应用场景的性能诉求，降低性能调优过程中反复调参的人工成本，提升性能调优效率。

**附：参考资料**

1. 《openEuler操作系统》·任炬、张尧学、彭许红编著·清华大学出版社

2. <https://openeuler.org/>

3. <https://gitee.com/openeuler>