

温故知新 | 思考题参考答案（二）

2023-01-23 LMOS 来自北京



《计算机基础实战课》

[课程介绍 >](#)



讲述：宇新

时长 00:23 大小 362.62K



你好，我是编辑小新。今天是大年初二，首先祝你春节愉快。

今天是我们第二次答疑课。上一次加餐，我们公布了课程前半部分的思考题参考答案。这次我们把后面课程的思考题答案也公布出来。建议你自行思考之后，再对答案，每节课都附上了超链接，方便你跳转回顾。

🔗 第二十三节课

Q：你觉得链接器使用的地址是物理内存地址，还是虚拟内存地址？

A：可以是物理内存地址也可以是虚拟内存地址，当 CPU 没有 MMU、PMU 等内存管理单元时，就是物理内存地址，否则就是虚拟地址。

🔗 第二十四节课

Q: 请问页表数据究竟放在什么地方呢?

A: 页表数据由操作系统构造, 放在物理内存中, 将其地址写入 MMU 的相关寄存器中。

🔗 第二十五节课

Q: 你觉得堆、栈空间是虚拟内存空间吗? 如果是, 请问是在什么时候分配的物理内存呢?

A: 堆和栈都是虚拟内存空间, 只是等到访问到具体虚拟地址产生了缺页异常中断, 系统才会分配物理内存。

🔗 第二十六节课

Q: 请简单说一下写时复制和请求调页的区别。

A: 写时复制是, 多个应用以只读方式共享物理内存页面, 任一应用写入时就分配新的物理内存页并进行复制。请示调页是应用访问自己虚拟内存时, 发生缺页异常(物理内存页面不存在)而分配一个新的物理内存页面, 供应用使用。

🔗 第二十七节课

Q: Golang 有了 mcentral 结构, 可以查找到 mspan 结构, 为什么还要建立 mcache 结构呢?

A: 因为 mcache 是每个 P 独有的, mcentral 是全局的, 直接访问 mcache 不用加锁, 会导致并行性下降。

🔗 第二十八节课

Q: 多个不同的进程可以包含相同的程序吗, 为什么?

A: 可以。因为一个程序在输入不同的数据就能得到不同的执行结果, 因而构成不同的进程; 但是在执行过程中, 程序代码区段是不能发生改变的, 可以参考 fork。

🔗 第二十九节课

Q: 请概述一下管道和信号这两种通信机制的不同。

A: 管道属于数据流通信，信号属于控制流通信。

🔗 第三十节课

Q: 进程间通信哪些是同步的，哪些是异步的？

A: 信号是异步的，其它三种通信方式既可以同步，也可以异步，创建的时候可以设置相关的标志。

🔗 第三十一节课

Q: 操作系统为什么要开发 IO Cache？

A: 其实 IO Cache 的存在，是为了缓解外设速度不同或者慢速的外设带来的系统性能问题。

🔗 第三十二节课

Q: 请问 Linux 网络通信的接口是什么？

A: Socket 套节字接口是 Linux 上 TCP/IP 网络接口 API，应用程序使用套节字 API 访问网络，而不是读写文件的方法。

🔗 第三十三节课

Q: 请说一说 iostat 与 iotop 的不同之处？

A: 关注点不同，iostat 关注系统全局，而 iotop 关注点是单个进程。

🔗 第三十四节课

Q: 一般 Linux 上的文件系统都有哪些内部结构？

A: 超级块、目录结构、inode 节点、数据块。

🔗 第三十五节课

Q: 请问 Ext3 文件系统的超级块放在硬盘分区的第几个扇区中。

A: 第二个到第三个扇区。

🔗 第三十六节课

Q: 请问 inode 号是对应于硬盘分区全局，还是相对于块组的？

A: inode 号对应于硬盘分区全局。

🔗 第三十七节课

Q: 浏览器的多进程模型下，进程之间是如何通信的呢？

A: 利用 legacy IPC 和 mojo，在不同平台下的实现是不同的。

🔗 第三十八节课

Q: 为什么 JS 代码会阻塞页面渲染？从浏览器设计的角度看，浏览器可以做哪些操作来进行优化？在开发前端应用过程中又可以做哪些优化呢？

A:

为什么 JS 代码会阻塞页面渲染

由于 JS 代码会修改 DOM 元素，渲染进程中的渲染线程和 JS 线程是互斥的。假如 JS 修改了一个元素的属性，但是渲染进程还没渲染到该元素，那 JS 修改属性就会失败。所以，把他们定为互斥是很有必要的。

浏览器可以做哪些操作来进行优化

Chrome 浏览器做了很多优化，其中一个主要的优化是预解析操作。当渲染引擎收到字节流之后，会开启一个预解析线程，用来分析 HTML 文件中包含的 JavaScript、CSS 等相关文件，

解析到相关文件之后，预解析线程会提前下载这些文件。

在开发前端应用过程中可以做哪些优化

可以使用 CDN 来加速 JavaScript 文件的加载，压缩 JavaScript 文件的体积。另外，如果 JavaScript 文件中没有操作 DOM 相关代码，就可以将该 JavaScript 脚本设置为异步加载，通过 `async` 或 `defer` 来标记代码。

`async` 和 `defer` 虽然都是异步的，不过还有一些差异，使用 `async` 标志的脚本文件一旦加载完成，会立即执行；而使用了 `defer` 标记的脚本文件，需要等到 `DOMContentLoaded` 事件之后执行。

在解析的过程中，JavaScript 代码可能会出现操作 CSS 的逻辑，所以在执行 JavaScript 之前，需要先解析 JavaScript 语句之上所有的 CSS 样式。所以如果代码里引用了外部的 CSS 文件，那么在执行 JavaScript 之前，还需要等待外部的 CSS 文件下载完成并解析生成 CSSOM 对象之后，才能执行 JavaScript 脚本。

因此我们知道，JavaScript 会阻塞 DOM 生成，而样式文件又会阻塞 JavaScript 的执行，所以在实际的工程中，需要重点关注 JavaScript 文件和样式表文件，以免不当影响到页面性能。

🔗 第三十九节课

Q: V8 在执行 JS 的过程中可以做哪些性能优化？

A: 核心就是利用 JIT 进行优化，一边将代码编译成机器码，一边执行这些机器码，适用于 JS 这种动态的解释型语言。

🔗 第四十节课

Q: 今天，我们学习了在计算机系统中常用的并行化、异步化和调度这三种通用的设计思想，那么请你思考一下，自己工作、生活中还有哪些场景用到了这些思想呢？

A: 开放问题，答案不唯一。存储比较大的文件，就会用到并行化思想，详情可以参考 🔗 加餐 02 的内容。

🔗 第四十一节课

Q: 这节课，我们学习了池化、分层、缓存这三招内功心法，请你思考一下在你的日常工作中有没有哪里用到了这几类设计思想呢？

A: 开放问题，答案不唯一。**CaaS** 通常被认为是 **IaaS** 的一种以容器为载体的子集，它介于 **IaaS** 和 **PaaS** 之间，它起到了屏蔽底层系统 **IaaS**，支撑并丰富上层应用平台 **PaaS** 的作用。这里就体现了分层思想。关于 **CaaS**，可以参考 🔗 加餐 04 了解。

🔗 第四十二节课

Q: 请你思考一下，**eBPF** 如果被误用，有没有可能带来新的安全问题呢？

A: 因为 **eBPF** 代码运行在内核态，并且拥有了从网络到文件系统等很多功能，所以其实也是可以被用来实现成 **rootkit** 后门的。

感兴趣的同学可以看一下这个 🔗 开源项目，它是一个典型的利用 **eBPF** 实现开启任意端口、反弹 **shell** 任意执行命令的后门程序。这意味着，攻与防的技术始终是在对抗中逐渐演进的，当 **eBPF** 程序被恶意利用的时候，其实也会引发更深层次的安全风险。

🔗 加餐 01

Q: **WebVirtCloud** 是如何把页面上创建虚拟机的操作传递给 **libvirt** 的，并找出对应的关键代码呢？

A: 你可以阅读 **vrtManager** 目录下的代码，可以看得出 **WebVirtCloud** 最终也是把各种 **WebUI** 中的操作。转换成 **libvirt** API 的调用，通过这种方式来实现虚拟机的管理。

以上就是今天的全部内容了，希望能对你有所帮助！如果你有更多问题，还是请你在留言区中提出。

分享给需要的人，Ta购买本课程，你将得 20 元

📄 生成海报并分享

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 温故知新 | 思考题参考答案（一）

下一篇 加餐01 | 云计算基础：自己动手搭建一款IAAS虚拟化平台

精选留言

💬 写留言

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。