

## 31.ARP的协议是干什么的?

ARP就是地址解析协议。其作用是在以太网环境中，数据的传输所依赖的是MAC(物理)地址而非IP地址，而将已知IP地址转换为MAC(物理)地址的工作是由ARP协议来完成的。用于实现从 IP 地址到 MAC 地址的映射，即询问目标IP对应的MAC地址。

## 32.什么是按需分页?

在操作系统中，进程是以页为单位加载到内存中的，按需分页是一种虚拟内存的管理方式。在使用请求分页的系统中，只有在尝试访问页面所在的磁盘并且该页面**尚未在内存中**时，也就发生了**缺页异常**，操作系统才会将磁盘页面复制到内存中。

## 33.map底层是用什么数据结构实现的?

vector底层是数组, list底层是双向链表

map和set底层都是红黑树

根据红黑树的原理，map与multimap可以实现 $O(\lg n)$ 的查找，插入和删除。

## 34.什么是多态?

多态可以理解为同一种事物, 具有多种形态, 或者说"**一个接口, 多种方法**" 分为**静态多态**和**动态多态**。

**静态多态**: 函数重载和运算符重载

**动态多态**: 虚函数

区别在于: 静态多态函数地址**早绑定**, 于编译期间确定函数地址 动态多态函数地址**晚绑定**, 于运行期间确定函数地址;

## 35.为什么会有设计模式?

"设计模式是对一个在我们周围不断重复发生的问题的描述，以及该问题的解决方案。这样，你就能一次又一次地使用该方案而不必做重复劳动。"

方便复用成熟解决方案，避免重复犯错与重复造轮。模式本身即业务，当使用了正确的设计模式，业务实现与迭代更轻松。预测、封装、隔离了变化，减少“改需求”产生的抵触情绪。方便测试与维护，程序测试结构稳定，方便单元测试与集成测试。

---

## 36.编译的几个阶段是什么？

---

预处理, 主要完成头文件展开的工作, 生成后缀为 .i 的文件。

编译, 主要检查语法错误, 无误则生成后缀为 .s 的汇编文件。

汇编, 将 .s 文件翻译成二进制 .o 目标文件。

链接, 链接程序中调用的一些库, 生成可执行文件

---

## 37.调度算法都有哪些？

---

先来先服务, 最短作业优先, 最短剩余时间优先,

轮询调度, 优先级调度, 最短进程优先, 公平分享调度

---

## 38.物理地址、逻辑地址、有效地址、线性地址、虚拟地址的区别是什么？

---

**物理地址**就是内存中真正的地址，它就相当于是你家的门牌号，你家就肯定有 这个门牌号，具有唯一性。不管哪种地址，最终都会映射为物理地址。

在**实模式**下，**段基址 + 段内偏移**经过地址加法器的处理，经过地址总线传输，最终也会转换为物理地址。

但是在**保护模式**下，**段基址 + 段内偏移**被称为线性地址，不过此时的段基址 不能称为真正的地址，而是会被称作为一个选择子的东西，选择子就是个索引，相当于数组的下标，通过这个索引能够在 GDT 中找到相应的段描述符，段描述符记录了段的起始、段的大小等信息，这样便得到了基地址。如果此时没有 开启内存分页功能，那么这个线性地址可以直接当做物理地址来使用，直接访问内存。如果开启了分页功能，那么这个线性地址又多了一个名字，这个名字 就是虚拟地址。

**不论在实模式还是保护模式下，段内偏移地址都叫做有效地址。有效地址也是 逻辑地址。**

线性地址可以看作是**虚拟地址**，**虚拟地址**不是真正的**物理地址**，但是虚拟地址 会最终被映射为物理地址。

---

## 39.什么是保护模式?什么是实模式?

---

**实模式**只能访问地址在**1M**以下的内存称为常规内存，我们把地址在1M 以上的内存称为扩展内存。

**寻址方式**: 由16位段寄存器的内容乘以16（左移4位）作为段基址，加上16位段偏移地址形成20位的物理地址，最大寻址空间1MB，最大分段64KB。

在**保护模式**下，全部32条地址线有效，可寻址高达**4G字节**的**物理地址空间**;

扩充的存储器**分段管理机制**和可选的存储器**分页管理机制**，不仅为存储器共享和保护提供了硬件支持，而且为实现虚拟存储器提供了硬件支持;

---

## 40.C语言和C++有什么区别?

---

C是一个**结构化语言**，它的重点在于**算法和数据结构**。对语言本身而言，C是C++的子集。

C程序的设计首要考虑的是**如何通过一个过程，对输入进行运算处理得到输出**。

对于C++，首要考虑的是**如何构造一个对象模型，让这个模型能够配合对应的问题，这样就可以通过获取对象的状态信息得到输出或实现过程控制**。因此C与C++的最大区别在于它们的**用于解决问题的思想方法不一样**。

C++实现了C中过程化控制及其它相关功能。而在C++中的C，相对于原来的C还有所加强，引入了重载、内联函数、异常处理等等，C++更是拓展了面向对象设计的内容，如类、继承、虚函数、模板和包容器类等等。

---