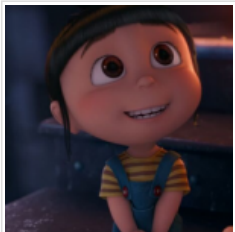


## 个人资料



翡青



访问: 545223次

积分: 11051

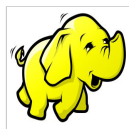
等级: BLOG &gt; ?

排名: 第556名

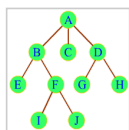
原创: 309篇 转载: 30篇

译文: 2篇 评论: 695条

## 博客专栏

Programming In  
The Linux  
Environment文章: 57篇  
阅读: 48041

## Hadoop与云计算

文章: 8篇  
阅读: 8168

## 数据结构与STL

文章: 31篇  
阅读: 26711

## Linux实践与提高

文章: 53篇  
阅读: 50032

## C++ 入门与提高

文章: 110篇  
阅读: 139604

## 操作系统学习笔记\_7\_内存管理 --内存管理基础

分类: 计算机系统与Linux内核

2014-07-19 18:18

1362人阅读

评论(0)

收藏 举报

操作系统

内存管理

存储

区分配

非连续分区分配

目录(?)

## 内存管理

## --内存管理基础

## 一、内存管理概念

## 1.存储管理的功能

- (1)内存空间的分配与回收,包括内存的分配和共享。
- (2)地址转换:内存管理配合硬件进行地址转换,把逻辑地址转换成物理地址。
- (3)内存空间的扩充:借助于虚拟存储器或交换覆盖技术来达到扩充内存容量的目的。
- (4)存储保护:为了避免相互干扰和破坏,必须提供保护功能。

## 2.地址重定位[程序的装入]

- (1)逻辑地址空间
- (2)物理地址空间
- (3)地址重定位
- (4)重定位类型

地址重定位分为静态重定位和动态重定位两类。把作业在装入过程中随即进行的地址变换方式,称为静态重定位。在作业执行过程中,当访问内存单元时才进行的地址变换方式,称为动态重定位。动态重定位是在程序执行过程中由硬件地址变换机构实现的。动态重定位的主要优点如下:用户作业在执行过程中,可以动态申请存储空间和在内存中移动;有利于程序段的共享。

## 3.链接

## (1)静态链接

静态链接(static linking)是在生成可执行文件时进行的,即在目标模块中记录被调用模块的名字符号地址(symbolic address),在可执行文件中将该名字改写为指令直接使用的数字地址。

## (2)动态链接

动态链接(dynamic-linking)是在运行可执行文件时进行的,亦即,执行过程中,当发现一个被调用模块未装入内存时,立即取找到该模块,并链接到调用者模块上。

通常被链接的共享代码称为动态链接库(Dynamic Link Library, DLL)或共享库(shared library)。动态链接又分为:装入时动态链接与运行时动态链接。

## 动态链接的优点:

- 1)共享:多个进程可以共用一个DLL,比较节省内存,从而可以减少文件的交换。
- 2)部分装入:一个进程可以将多种操作分散在不同的DLL中实现,而只将当前操作的DLL装入内存。
- 3)便于局部代码修改:即便于代码升级和代码重用;只要函数的接口参数(输入和输出)不变,则修改函数及其DLL时,无需对可执行文件重新编译或链接。
- 4)便于适应运行环境:调用不同的DLL,就可以适应多种使用环境并提供不同的功能。

文章分类
<a href="#">C++</a> (115)
<a href="#">Linux</a> (74)
<a href="#">算法与数据结构</a> (33)
<a href="#">Hadoop与云计算</a> (8)
<a href="#">Linux环境高级编程</a> (51)
<a href="#">计算机网络与TCP/IP</a> (4)
<a href="#">计算机系统与Linux内核</a> (14)
<a href="#">MySQL数据库编程</a> (15)
<a href="#">程序人生</a> (32)

文章存档
<a href="#">2015年05月</a> (1)
<a href="#">2015年04月</a> (4)
<a href="#">2015年03月</a> (13)
<a href="#">2015年02月</a> (40)
<a href="#">2015年01月</a> (24)
展开

阅读排行
<a href="#">2.5年, 从0-&gt;阿里</a> (17574)
<a href="#">Windows下远程登录到Li</a> (5950)
<a href="#">C++ Primer 学习笔记_1_</a> (4751)
<a href="#">UNIX网络编程 --环境搭建</a> (4039)
<a href="#">我的2013 --岁月划过生命</a> (3884)
<a href="#">C++ Primer 学习笔记_27</a> (3804)
<a href="#">C++ Primer 学习笔记_45</a> (3652)
<a href="#">C++ Primer 学习笔记_2_</a> (3106)
<a href="#">为了那永不坠落的梦想...</a> (2990)
<a href="#">Linux 学习笔记_2_Linux</a> (2770)

最新评论
<a href="#">2.5年, 从0-&gt;阿里</a> <b>pirDOL</b> : 楼主你好，拜读了《2.5年，从0->阿里》，收获良多。我也在今年拿到了yunos的实习offer，但...
<a href="#">2.5年, 从0-&gt;阿里</a> <b>adminabcd</b> : 哎，自从上了csdn才发现牛人那么多，而自己却是个井底之蛙，大学完全荒废了
<a href="#">2.5年, 从0-&gt;阿里</a> <b>wodeai1625</b> : 哥们看了你的博客后感慨很深很深，但是有些内容我还不是很了解。想和你qq交流下。不知道学长方便吗
<a href="#">我的2013 --岁月划过生命线(大二)</a> <b>wodeai1625</b> : 哥们，看你的博客感慨很深很深。可不可以加你的qq想和你好好聊聊
<a href="#">2.5年, 从0-&gt;阿里</a> <b>MonroeD</b> : 请问一下学长，有没有问你mySQL等数据库的问题吗？
<a href="#">C++ Primer 学习笔记_108(大结局)</a> <b>cyfcsd</b> : 楼主高人
<a href="#">TCP/IP入门(3) --传输层</a> <b>荒漠之弦</b> : 编辑的非常好！我也转走了，向你学习！
<a href="#">为了那永不坠落的梦想...</a> <b>kkwsj</b> : 向LZ学习
<a href="#">2.5年, 从0-&gt;阿里</a> <b>翊青</b> : @Suprman:恩恩，对，还是感觉自己会的东西太浅太少了... 正加倍努力中....奋斗
<a href="#">2.5年, 从0-&gt;阿里</a>

## 二、交换与覆盖[现在用的已经不多]

覆盖是指一个作业的某些程序段,或几个作业的某些部分轮流使用某一段存储空间。

交换实质上是使用外存做缓冲,让用户程序在较小的存储空间中通过不断地换出换入而可以运行较大的作业。交换可以是进程的整体交换,称为“进程交换”。如果交换以进程的部分页面或段为单位进行,则分别称之为“页面交换”或“分段交换”,又统称为“部分交换”。这种交换方法是实现请求分页及请求分段式存储管理的基础,**虚拟存储系统**就是采用了这种部分交换而得以实现。

## 三、存储分配方式

### 1.静态分配

在装配程序把目标模块进行链接装入时确定它们在内存中的位置。

### 2.动态分配

其执行过程中可根据需要申请附加的存储空间。

## 四、连续分配管理方式

### 1.固定式和可变式分区存储管理

- (1)固定式分区存储管理{分区大小相等，分区大小不等}
- (2)可变分区存储管理:根据作业的实际需要动态地划分存储空间。

### 分配算法

- 1)首次适应算法(FirstFit)
- 2)邻近适应算法(NextFit)
- 3)最佳适应算法(BestFit)
- 4)最坏适应算法(WorstFit)

动态分区分配可能会出现“外部碎片”问题，而采用“内存紧缩”技术,可以把碎片集中起来形成一个大的空闲区。

### 2.分区的存储保护

(1)界地址保护:界地址保护又称为界限寄存器保护。

界限寄存器方式:下界寄存器存放起始地址,上界寄存器存放结束地址。

基址寄存器和限长寄存器:基址寄存器存放起始地址,限长寄存器存放最大长度。

(2)存储键保护:同一作业的各页面所对应的内存块都要指定一个相同的,但又不与其他作业相重的键码。这个键码存于快速寄存器和该作业的程序状态字PSW中,当程序要访问某一块时,将程序状态字中的键码与被访问块的键码进行比较,若相符,则表明允许本次访问,否则发出越界中断。

## 五、非连续分配管理方式

### 1.简单分页存储管理

分页存储管理技术中的基本作法是:

- (1)等分内存:把内存划分成大小相等的单位,称为存储块,或称为页框(PageFrame)。
- (2)用户逻辑地址空间的分页:把用户的逻辑地址空间划分成若干个与存储块大小相等的单位,称之为页面或页(Page)。并给各页从起始开始依次编以连续的页号0,1,2,.....。
- (3)逻辑地址的表示:用户的逻辑地址从基地址“0”开始连续编址,即相对地址。
- (4)内存分配原则:系统以存储块为单位把内存分给作业或进程。
- (5)页表和页表地址寄存器:作业的一页可以分配到内存空间中任何一个可用的存储块。

简单分页管理方法在作业或进程开始执行之前,把该作业或进程的程序段和数据全部装入内存的各个空闲块中,并通过页表和硬件地址变换机构实现虚拟地址到内存物理地址的地址映射。

- (6)快表。
- (7)页的共享与保护:保护的项目一般有读、写、运行等。

### 2.分段存储管理

段式存储管理按用户作业中的自然段来划分逻辑空间,例如代码段,数据段,堆栈段等等。每段占用连续的地址空间,因此作业的逻辑地址是二维的,由段号和段内地址组成。

- (1)把程序按内容或过程(函数)关系分成段,每段有自己的名字。
- (2)地址转换。
- (3)段的共享与保护。
- (4)分段存储管理特点:优点是便于程序模块化处理和便于处理变换的数据结构;便于动态链接;便于共享分段;可以实现虚拟存储器,使作业的地址空间不受内存容量的限制。

上一篇 操作系统学习笔记\_6\_进程管理 --死锁

下一篇 操作系统学习笔记\_8\_内存管理 --虚拟内存管理

顶 24

踩 0

猜你在找

- iOS8开发技术（Swift版）：iOS基础知识
- Cocos2d-Lua手游开发基础篇
- 实战进阶学习Unity3d游戏开发
- 韦东山嵌入式Linux第一期视频
- VC++游戏开发基础系列从入门到精通
- 《深入理解Linux内存管理》学习笔记--00201修正
- C++内存管理学习笔记4
- 从C到C++再到Obj-C内存管理学习笔记三
- IOS学习笔记3Objective C简单的内存管理
- Object-C学习笔记十-----内存管理

准备好了么？跳 吧 ！

更多职位尽在 CSDN JOB

网络管理专员	我要跳槽	开发管理工程师（互联网）	我要跳槽
广州新东方培训学校	3-5K/月	重庆东银控股集团有限公司	6-10K/月
信息安全管理工程师/IT审计	我要跳槽	社区管理专员	我要跳槽
华宝（上海）管理有限公司	12-24K/月	亚艺网媒科技发展（北京）有限公司	2-4K/月

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker

OpenStack VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC

WAP jQuery BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML

LBS Unity Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra

CloudStack FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App

SpringSide Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP

HBase Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap