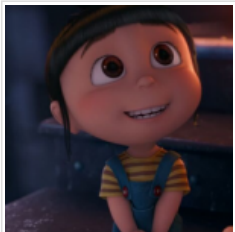


个人资料



翡青



访问： 545224次

积分： 11051

等级： BLOG > ?

排名： 第556名

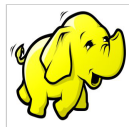
原创： 309篇 转载： 30篇

译文： 2篇 评论： 695条

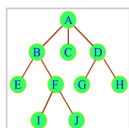
博客专栏



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_8_内存管理 --虚拟内存管理

分类： 计算机系统与Linux内核

2014-07-19 21:55

1448人阅读

评论(0) 收藏 举报

操作系统

内存管理

虚拟

求分页管理

页面置换算法

目录(?)

内存管理

--虚拟内存管理

一、虚拟内存基本概念

1.局部性原理

(1)时间局部性:程序中的某条指令一旦运行,不久以后该指令可能再次运行。产生时间局部性的典型原因是由于程序中存在大量的循环操作。

(2)空间局部性:一旦程序访问了某个存储单元,不久以后其附近的存储单元也将被访问,其典型情况是程序顺序运行。

2.虚拟内存

基于局部性原理,应用程序在运行之前并不必全部装入内存,仅需将当前运行到的那部分程序和数据装入内存便可启动程序的运行,其余部分仍驻留在外存上。当要运行的指令或访问的数据不在内存时,再由操作系统通过请求调入功能将它们调入内存,以使程序能继续运行。如果此时内存已满,则还需通过置换功能,将内存中暂时不用的程序或数据调至盘上,腾出足够的内存空间后,再将要访问的程序或数据调入内存,使程序继续运行。

3.实现虚拟内存的基础

硬件基础	一定容量的内存; 大容量的外存; 地址变换机构(含快表); 缺页中断机构。
软件基础	虚实转换的数据结构(页表、段表等); 中断服务处理程序; 操作系统支持。

实现虚拟的方案:

- 页式虚存(请求分页:最常用)
- 段式虚存
- 段页式虚存

4.虚拟内存的主要特征

- (1)多次性。
- (2)对换性。
- (3)虚拟性。

二、请求分页管理方式

1.请求分页的基本原理

文章分类

C++ (115)

Linux (74)

算法与数据结构 (33)

Hadoop与云计算 (8)

Linux环境高级编程 (51)

计算机网络与TCP/IP (4)

计算机系统与Linux内核 (14)

MySQL数据库编程 (15)

程序人生 (32)

文章存档

2015年05月 (1)

2015年04月 (4)

2015年03月 (13)

2015年02月 (40)

2015年01月 (24)

展开

阅读排行

2.5年, 从0->阿里 (17574)

Windows下远程登录到Li (5950)

C++ Primer 学习笔记_1_ (4751)

UNIX网络编程 --环境搭建 (4039)

我的2013 --岁月划过生命 (3884)

C++ Primer 学习笔记_27 (3804)

C++ Primer 学习笔记_45 (3652)

C++ Primer 学习笔记_2_ (3106)

为了那永不坠落的梦想... (2990)

Linux 学习笔记_2_Linux (2770)

最新评论

2.5年, 从0->阿里
pirDOL: 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...

2.5年, 从0->阿里
adminabcd: 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

2.5年, 从0->阿里
wodeai1625: 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

我的2013 --岁月划过生命线(大二)
wodeai1625: 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

2.5年, 从0->阿里
MonroeD: 请问一下学长，有没有问你mySQL等数据库的问题吗？

C++ Primer 学习笔记_108(大结局)
cyfcsd: 楼主高人

TCP/IP入门(3) --传输层
荒漠之弦: 编辑的非常好！我也转走了，向你学习！

为了那永不坠落的梦想...
kkwsj: 向LZ学习

2.5年, 从0->阿里
翡青: @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

2.5年, 从0->阿里

请求分页存储管理是在简单分页管理基础上发展起来的。请求页式管理在作业或进程开始执行之前,不要求把作业或进程的程序段和数据段一次性地全部装入内存,而只把当前需要的一部分页面装入内存,其它部分在作业执行过程中需要时,再从外存上调入内存。

2.页表的扩充

读写控制	页框号
------	-----

(a) 简单分页系统的页表

外存地址	其它扩展	读写控制	修改位	引用位	存在位	页框号
------	------	------	-----	-----	-----	-----

(b) 虚拟请求分页系统的页表

页表的表项

- (1)存在位(present/absent):表示该页是否在内存。
- (2)修改位(modified):该位为“0”时,在示访页面中的数据未被修改过。
- (3)引用位(referenced):表示该页面在最近期间是否被访问引用过。
- (4)外存地址(swaparea address):指出该页面在外存上的存放地址。
- (5)其它:如页面保护位(protection),禁止缓存位(cachedisabled)等。

3.地址变换

请求分页的地址变换初始过程十分类似于简单分页系统的地址变换。

4.缺页中断处理

当存在位为0时,表示该页不在内存,则必须确定它在外存中的存放地址。并把它从外存中调入内存。若内存中没有空闲块时,首先按照某种策略选择某页进行淘汰。以腾出空闲块供本次调入的页占用。这个过程也被称之为页面置换。若被选中淘汰的页面中的信息修改过(修改位=1)还必须将其写回外存。如内存中有空闲块,则根据该页在外存的地址,调入所需页面,并更新页表表项,最后恢复被中断的指令重新执行。

5.调页策略

这是一个何时把页面装入内存的问题。如果出现缺页中断,表明企图对一个不存在于内存的页面要求访问。显然,应该立即装入该页面。这种仅当需要时才调取页面的策略,称为请求式调页,采用请求式调页策略的分页系统称为请求式分页;而把事先调取页面的策略称为预调页。

三、页面置换算法

1.随机淘汰算法

在无法确定那些页被访问的概率较低时,随机地选择某个用户的页面并将其换出。

2.先进先出算法(FIFO)

FIFO(firstin first out)算法:总是选择驻留内存时间最长的页面进行淘汰。其理由是:最早调入内存的页面,其不再被使用的可能性最大。

FIFO算法忽略了一种现象的存在,就是在内存中停留时间最长的页往往也是经常被访问的页。将这些页淘汰,很可能刚置换出去,又请求调用该页,致使缺页中断较频繁,严重降低内存的利用率。

FIFO的另一缺点是它有一种异常现象。称为Belady异常。

3.最佳置换算法(OPT)

最佳置换算法的基本思想是:从内存中移出永远不再需要的页面。

4.最近最久未使用页面置换算法(LRU)

这种算法的基本思想是,利用局部性原理,根据一个作业在执行过程中过去的页面访问历史来推测未来的行为。它认为过去一段时间里不曾被访问过的页面,在最近的将来可能也不会再被访问。

5.最近没有使用页面置换算法(NRU)

6.时钟算法(CLOCK)

时钟算法是将作业已调入内存的页面链成循环队列,使用页表中的“引用位”,用一个指针指向循环队列中的下一个将被替换的页面。

四、页面分配策略

- 1.固定分配局部置换策略
- 2.可变分配全局置换策略
- 3.可变分配局部置换策略

五、工作集

工作集也称为驻留集,是某一个进程调入物理内存的页面的集合,这些页面是频繁地被使用到的,因此长期驻留内存是有利于提高处理机的效率。工作集模型是基于局部性原理假设的。

六、抖动

如果分配给进程的存储块数量小于进程所需要的最小值,进程的运行将很频繁地产生缺页中断。这种频率非常高的页面置换现象称之为抖动(也称为颠簸)。往往是刚被淘汰的页面马上被选中调页而进入内存。抖动将引起严重的系统性能下降。

防止抖动现象有多种办法,例如,采取局部替换策略,引入工作集算法,挂起或撤销若干进程等。

版权声明：本文为博主原创文章，未经博主允许不得转载。

上一篇 操作系统学习笔记_7_内存管理 --内存管理基础

下一篇 操作系统学习笔记_9_文件管理 --文件系统基础

顶

20

踩

0

猜你在找

C语言及程序设计提高	windows下32位汇编语言学习笔记
Qt基础与Qt on Android入门	windows下32位汇编语言学习笔记
Cocos2d-Lua手游开发基础篇	Debian学习笔记
实战进阶学习Unity3d游戏开发	Redis内存使用优化与存储学习笔记三
JavaScript for Qt Quick(QML)	《Linux命令行与shell脚本编程大全》 第一二章

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](#) [apttech](#) [Perl](#) [Tornado](#) [Ruby](#) [Hibernate](#) [ThinkPHP](#)
[HBase](#) [Pure](#) [Solr](#) [Angular](#) [Cloud Foundry](#) [Redis](#) [Scala](#) [Django](#) [Bootstrap](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) [webmaster@csdn.net](#) 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 