

全国计算机技术与软件专业技术
资格（水平）考试用书

系统架构设计师考试 历年试题分析与解答

希赛教育软考学院 主编

考点脉络

总结和归纳
考试必备的知识点

+

考点精讲

“画龙点睛” 考点脉络
部分中列出的重点

+

考前冲刺

提供了整个学科
体系的强化练习，
使读者做到举一反三

打通软考任督二脉



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

TP3-44/118

2013

全国计算机技术与软件专业技术
资格（水平）考试用书

系统架构设计师考试 历年试题分析与解答

希赛教育软考学院 主编

北方工业大学图书馆



C00340141

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书由希赛教育软考学院组织编写,作为计算机技术与软件专业技术资格(水平)考试中的系统架构设计师级别考试辅导培训教材。本书根据最新的系统架构设计师考试大纲及培训指南,按照所有知识点对历年考试试题进行了分析和总结,对新版的考试大纲规定的内容有重点地进行细化和深化。

考生可通过阅读本书掌握考试大纲规定的知识,掌握考试重点和难点,熟悉考试方法、试题形式、试题的深度和广度,以及内容的分布、解答问题的方法和技巧。

本书可作为系统架构设计师考试的辅导书籍,也可作为信息系统项目管理师、系统分析师和网络规划设计师考试的参考书籍。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

系统架构设计师考试历年试题分析与解答 / 希赛教育软考学院主编. —北京: 电子工业出版社, 2013.9
全国计算机技术与软件专业技术资格(水平)考试用书
ISBN 978-7-121-21193-5

I. ①系… II. ①希… III. ①计算机系统工程技术人员—资格考试—题解 IV. ①TP30-44

中国版本图书馆 CIP 数据核字 (2013) 第 182147 号

责任编辑: 孙学瑛

特约编辑: 顾慧芳

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 27 字数: 518 千字

印 次: 2013 年 9 月第 1 次印刷

定 价: 69.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

全国计算机技术与软件专业技术资格（水平）考试（以下简称“软考”）是一个难度很高的考试，考生的平均通过率很低，主要原因是考试范围十分广泛，涉及计算机专业的每门课程，还要加上数学、英语、系统工程、信息化和知识产权等知识，且注重考查新技术和新方法的应用。考试不但注重广度，而且还有一定深度。特别是高级资格考试，不但要求考生具有扎实的理论基础知识，还要具备丰富的实战经验。

本书是为软考中的系统架构设计师级别而编写的考试用书，全书分析了历年系统架构设计师考试的所有考题，对试题进行详细的分析与解答，对有关重点和难点进行了深入的分析。

作者权威，阵容强大

希赛教育（www.educity.cn）专业从事人才培养、教育产品开发、教育图书出版，在职业教育方面具有极高的权威性。特别是在在线教育方面，稳居国内首位。希赛教育的远程教育模式得到了国家教育部门的认可和推广。

希赛教育软考学院（www.csairk.com）是全国计算机技术与软件专业技术资格（水平）考试的顶级培训机构，拥有近 20 名资深软考辅导专家，负责了高级资格的考试大纲制订工作，以及软考辅导教材的编写工作，共组织编写和出版了 80 多本软考教材，内容涵盖了初级、中级和高级的各个专业，包括教程系列、辅导系列、考点分析系列、冲刺系列、串讲系列、试题精解系列、疑难解答系列、全程指导系列、案例分析系列、指定参考用书系列、一本通 11 个系列的书籍。希赛教育软考学院的专家录制了软考培训视频教程、串讲视频教程、试题讲解视频教程、专题讲解视频教程 4 个系列的软考视频。希赛教育软考学院的软考教材、软考视频、软考辅导为考生助考、提高通过率做出了不可磨灭的贡献，在软考领域有口皆碑。特别是在高级资格领域，无论是考试教材，还是在线辅导和面授，希赛教育软考学院都独占鳌头。

本书由希赛教育软考学院主编，参加编写的人员有张友生、桂阳、胡钊源、谢顺、刘洋波、王勇、邓旭光、彭雪阳、左水林、胡光超和何玉云。

在线测试，心中有数

上学吧（www.shangxueba.com）在线测试平台为考生准备了在线测试，其中有数十套全真模拟试题和考前密卷，考生可选择任何一套进行测试。测试完毕，系统自动判卷，立即给出分数。

对于考生做错的地方，系统会自动记忆，待考生第二次参加测试时，可选择“试题复习”。这样，系统就会自动把考生原来做错的试题显示出来，供考生重新测试，以加强记忆。

如此，读者可利用上学吧在线测试平台的在线测试系统检查自己的实际水平，加强考前训练，做到心中有数，考试不慌。

诸多帮助，诚挚致谢

在本书出版之际，要特别感谢全国软考办的命题专家们，编者在本书中引用了部分考试原题，使本书能够尽量方便读者的阅读。在本书的编写过程中，参考了许多相关的文献和书籍，编者在此对这些参考文献和书籍的作者表示感谢。

感谢电子工业出版社孙学瑛老师和黄爱萍老师，她们在本书的策划、选题的申报、写作大纲的确定，以及编辑、出版等方面，付出了辛勤的劳动和智慧，给予了我们很多的支持和帮助。

感谢参加希赛教育软考学院辅导和培训的学员，正是他们的想法汇成了本书的源动力，他们的意见使本书更加贴近读者。

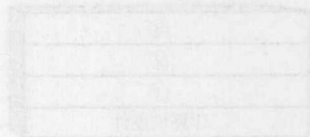
由于编者水平有限，且本书涉及的内容很广，书中难免存在错漏和不妥之处，编者诚恳地期望各位专家和读者不吝指正和帮助，对此，我们将十分感激。

希赛教育软考学院

2013年8月

目 录

第 1 章 操作系统	1	第 8 章 软件架构设计	186
1.1 考点分析	1	8.1 考点分析	186
1.2 试题精解	1	8.2 试题精解	187
第 2 章 数据库系统	18	第 9 章 应用数学	253
2.1 考点分析	18	9.1 考点分析	253
2.2 试题精解	19	9.2 试题精解	254
第 3 章 计算机硬件基础及嵌入式 系统设计	37	第 10 章 知识产权与标准化	262
3.1 考点分析	37	10.1 考点分析	262
3.2 试题精解	38	10.2 试题精解	263
第 4 章 数据通信与计算机网络	65	第 11 章 系统配置与性能评价	275
4.1 考点分析	65	11.1 考点分析	275
4.2 试题精解	65	11.2 试题精解	276
第 5 章 系统安全性与保密性设计	88	第 12 章 专业英语	289
5.1 考点分析	88	12.1 考点分析	289
5.2 试题精解	89	12.2 试题精解	289
第 6 章 信息化基础	103	第 13 章 案例分析	297
6.1 考点分析	103	13.1 考点分析	297
6.2 试题精解	104	13.2 试题精解	299
第 7 章 系统开发基础	126	第 14 章 架构设计论文	392
7.1 考点分析	126	14.1 考点分析	392
7.2 试题精解	128	14.2 试题精解	393



第 1 章

操作系统

1.1 考点分析

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

- (1) 操作系统的类型和结构。
- (2) 操作系统基本原理。
- (3) 网络操作系统及网络管理。

在这些知识点中，考查重点是操作系统的概念及特点、进程管理、存储管理、文件管理。具体考查知识点分布情况如表 1-1 所示。

表 1-1 历年考查知识点分布情况表

试 题	考查知识点
2009 年 11 月试题 1~4	操作系统基本概念、前趋图与 PV 操作
2010 年 11 月试题 1~4	微内核操作系统、文件系统、页面置换算法
2011 年 11 月试题 1~4	操作系统接口、前趋图与 PV 操作
2012 年 11 月试题 1~4	PV 操作与信息量、索引文件

1.2 试题精解

试题 1（2009 年 11 月试题 1）

计算机系统中硬件层之上的软件通常按照三层来划分，如图 1-1 所示，图中①②③分别表示（1）。

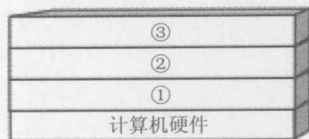


图 1-1 计算机结构图

- (1) A. 操作系统、应用软件和其他系统软件
 B. 操作系统、其他系统软件和应用软件
 C. 其他系统软件、操作系统和应用软件
 D. 应用软件、其他系统软件和操作系统

试题分析

操作系统 (Operating System) 的目的是为了填补人与机器之间的鸿沟，即建立用户与计算机之间的接口，而为裸机配置的一种系统软件，如图 1-2 所示。

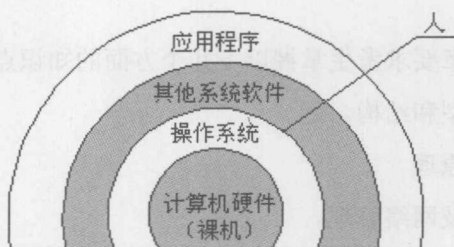


图 1-2 计算机结构图 (完整)

从图 1-2 可以看出，操作系统是裸机上的第一层软件，是对硬件系统功能的首次扩充。它在计算机系统中占据重要而特殊的地位，其他系统软件属于第二层，如编辑程序、汇编程序、编译程序和数据库管理系统等系统软件（这些软件工作于操作系统之上，可服务于应用软件，所以有别于应用软件）；大量的应用软件属于第三层，例如希赛教育网上辅导平台，常见的一系列 MIS 系统等。其他系统软件和应用软件都是建立在操作系统基础之上的，并得到它的支持和取得它的服务。从用户角度看，当计算机配置了操作系统后，用户不再直接使用计算机系统硬件，而是利用操作系统所提供的命令和服务去操纵计算机，操作系统已成为现代计算机系统中必不可少的最重要的系统软件，因此把操作系统看作是用户与计算机之间的接口。

试题答案

- (1) B

试题 2 (2009 年 11 月试题 2~4)

某计算机系统有一个 CPU、一台扫描仪和一台打印机。现有三个图像任务，每个任务有三个程序段：扫描 S_i ，图像处理 C_i 和打印 P_i ($i=1,2,3$)。图 1-3 为三个任务各程序段并发执行的前趋图，其中，(2) 可并行执行，(3) 的直接制约，(4) 的间接制约。

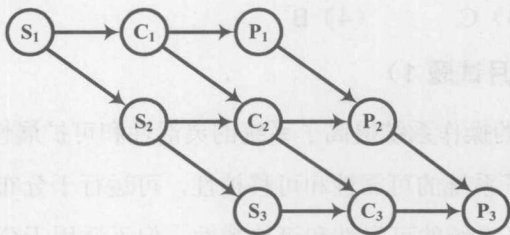


图 1-3 前趋图

- (2) A. “ C_1S_2 ”, “ $P_1C_2S_3$ ”, “ P_2C_3 ”
B. “ C_1S_1 ”, “ $S_2C_2P_2$ ”, “ C_3P_3 ”
C. “ $S_1C_1P_1$ ”, “ $S_2C_2P_2$ ”, “ $S_3C_3P_3$ ”
D. “ $S_1S_2S_3$ ”, “ $C_1C_2C_3$ ”, “ $P_1P_2P_3$ ”
- (3) A. S_1 受到 S_2 和 S_3 、 C_1 受到 C_2 和 C_3 、 P_1 受到 P_2 和 P_3
B. S_2 和 S_3 受到 S_1 、 C_2 和 C_3 受到 C_1 、 P_2 和 P_3 受到 P_1
C. C_1 和 P_1 受到 S_1 、 C_2 和 P_2 受到 S_2 、 C_3 和 P_3 受到 S_3
D. C_1 和 S_1 受到 P_1 、 C_2 和 S_2 受到 P_2 、 C_3 和 S_3 受到 P_3
- (4) A. S_1 受到 S_2 和 S_3 、 C_1 受到 C_2 和 C_3 、 P_1 受到 P_2 和 P_3
B. S_2 和 S_3 受到 S_1 、 C_2 和 C_3 受到 C_1 、 P_2 和 P_3 受到 P_1
C. C_1 和 P_1 受到 S_1 、 C_2 和 P_2 受到 S_2 、 C_3 和 P_3 受到 S_3
D. C_1 和 S_1 ，受到 P_1 、 C_2 和 S_2 受到 P_2 、 C_3 和 S_3 受到 P_3

试题分析

如图 1-3 所示，当 S_1 执行完毕后，计算 C_1 与扫描 S_2 可并行执行； C_1 与 S_2 执行完毕后，打印 P_1 、计算 C_2 与扫描 S_3 可并行执行； P_1 、 C_2 与 S_3 执行完毕后，打印 P_2 与计算 C_3 可并行执行。

根据题意，系统中有三个任务，每个任务有三个程序段，从前趋图中可以看出，系统要先进行扫描 S_i ，然后再进行图像处理 C_i ，最后进行打印 P_i ，所以 C_1 和 P_1 受到

S_1 直接制约、 C_2 和 P_2 受到 S_2 的直接制约、 C_3 和 P_3 受到 S_3 的直接制约。

系统中有一台扫描仪，因此 S_2 和 S_3 不能运行是受到了 S_1 的间接制约。如果系统中有三台扫描仪，那么 S_2 和 S_3 能运行；同理， C_2 和 C_3 受到 C_1 的直接制约、 P_2 和 P_3 受到 P_1 的间接制约。

试题答案

(2) A (3) C (4) B

试题 3 (2010 年 11 月试题 1)

采用微内核结构的操作系统提高了系统的灵活性和可扩展性，(1)。

- (1) A. 并增强了系统的可靠性和可移植性，可运行于分布式系统中
- B. 并增强了系统的可靠性和可移植性，但不适用于分布式系统
- C. 但降低了系统的可靠性和可移植性，可运行于分布式系统中
- D. 但降低了系统的可靠性和可移植性，不适用于分布式系统

试题分析

现代操作系统大多拥有两种工作状态，分别是核心态和用户态。一般应用程序工作在用户态，而内核模块和最基本的操作系统核心工作在核心态。

微内核操作系统结构是 20 世纪 80 年代后期发展起来的。操作系统的一个发展趋势是将传统的操作系统代码放置到更高层，从操作系统中去掉尽可能多的东西，而只留下一个最小的核心，称之为微内核。通常的方法是将大多数操作系统功能由在用户态运行的服务器进程来实现。为了获取某项服务，用户进程（客户进程）将请求发送给一个服务器进程，服务器进程完成此操作后，把结果返回给用户进程。这样，服务器以用户进程的形式运行，而不是运行在核心态。因此，它们不能直接访问硬件，某个服务器的崩溃不会导致整个系统的崩溃。客户/服务器结构的另一个优点是它更适用于分布式系统。

微内核技术的主要优点如下。

- ① 统一的接口，在用户态和核心态之间无需进程识别。
- ② 可伸缩性好，能适应硬件更新和应用变化。
- ③ 可移植性好，所有与具体机器特征相关的代码，全部隔离在微内核中，如果操作系统要移植到不同的硬件平台上，只需修改微内核中极少代码即可。
- ④ 实时性好，微内核可以方便地支持实时处理。

⑤ 安全可靠性强，微内核将安全性作为系统内部特性来进行设计，对外仅使用少量应用编程接口。

⑥ 支持分布式系统，支持多处理器的体系结构和高度并行的应用程序。

虽然微内核操作系统具有诸多优点，但它并非完美无缺。例如，在运行效率方面，它就不如以前传统的操作系统。

试题答案

(1) A

试题 4 (2010 年 11 月试题 2)

若操作系统文件管理程序正在将修改后的 (2) 文件写回磁盘时系统发生崩溃，对系统的影响相对较大。

- (2) A. 用户数据 B. 用户程序
C. 系统目录 D. 空闲块管理

试题分析

操作系统为了实现“按名存取”，必须为每个文件设置用于描述和控制文件的数据结构，专门用于文件的检索，因此至少要包括文件名和存放文件的物理地址，该数据结构称为文件控制块 (File Control Block, FCB)，文件控制块的有序集合称为文件目录，或称为系统目录文件。若操作系统正在将修改后的系统目录文件写回磁盘时系统发生崩溃，则对系统的影响相对较大。

试题答案

(2) C

试题 5 (2010 年 11 月试题 3~4)

某虚拟存储系统采用最近最少使用的 (LRU) 页面淘汰算法，假定系统为每个作业分配 4 个页面的主存空间，其中一个页面用来存放程序。现有某作业的程序如下：

```
Var A: Array[1.. 100, 1.. 100] OF integer;  
i, j: integer;  
FOR i:=1 to 100 DO  
  FOR j:=1 to 100 DO  
    A[i, j]:=0;
```

设每个页面可存放 200 个整数变量，变量 i、j 存放在程序页中。初始时，程序

及 i, j 均已在内存, 其余 3 页为空。若矩阵 A 按行序存放, 那么当程序执行完后共产生 (3) 次缺页中断; 若矩阵 A 按列序存放, 那么当程序执行完后共产生 (4) 次缺页中断。

- | | |
|-----------|----------|
| (3) A. 50 | B. 100 |
| C. 5000 | D. 10000 |
| (4) A. 50 | B. 100 |
| C. 5000 | D. 10000 |

试题分析

虚拟存储管理的提出就是为了解决这一问题, 应用程序在运行之前未必全部装入内存, 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。当要执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, 以使程序能继续执行。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外存上, 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, 使程序继续执行。这样, 便可使一个大的用户程序能在较小的内存空间中运行, 也可在内存中同时装入更多的进程使它们并发执行。从用户的角度看, 该系统具有的内存容量比实际的内存容量大得多。将这种具有请求调入功能和置换功能, 能从逻辑上对内存容量加以扩充的存储器系统称为虚拟存储系统。

局部性原理

虚拟存储管理能够在作业信息不全部装入内存的情况下保证作业正确运行, 是利用了程序执行时的局部性原理。局部性原理是指程序在执行时呈现出局部性规律, 即在一较短的时间内, 程序的执行仅局限于某个部分。相应地, 它所访问的存储空间也仅局限于某个区域。程序局部性包括时间局部性和空间局部性, 时间局部性是指程序中的某条指令一旦执行, 不久以后该指令可能再次执行。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; 空间局部性是指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, 即程序在一段时间内所访问的地址可能集中在一定的范围内, 其典型情况是程序顺序执行。

工作集

在虚拟存储管理中, 可能会出现这种情况, 即对于刚被替换出去的页, 立即又要被访问, 需要将它调入, 因无空闲内存又要替换另一页, 而后者是即将被访问的

页，于是造成了系统需花费大量的时间忙于进行这种频繁的页面交换，致使系统的实际效率很低，严重时导致系统瘫痪，这种现象称为抖动现象。防止抖动现象有多种办法，例如，采取局部替换策略、引入工作集算法和挂起若干进程等。工作集是指在某段时间间隔内，进程实际要访问的页面的集合。引入虚拟内存后，程序只需有少量的内存就可运行，但为了使程序有效地运行，较少产生缺页，必须使程序的工作集全部在内存中。

页面置换算法

当内存中没有空闲页面，而又有程序和数据需要从外存中装入内存运行时，就需要从内存中选出一个或多个页面淘汰出去，以便新的程序和数据装入运行，良好的页面置换算法应该淘汰那些被访问概率最低的页，将它们移出内存。

① 随机淘汰算法。无法确定哪些页被访问的概率较低时，随机地选择某个页面，并将其换出。

② 轮转算法。按照内存页面的编号，循环地换出内存中一个可以被换出的页，无论该页是刚换进来还是已驻留内存很长时间。

③ 先进先出算法（First In First Out, FIFO）。FIFO 算法总是选择在内存驻留时间最长的一页将其淘汰。实现 FIFO 算法需要把各个已分配页面按页面分配时间顺序链接起来，组成 FIFO 队列，并设置一置换指针，指向 FIFO 队列的队首页面。FIFO 算法忽略了一种现象的存在，那就是在内存中停留时间最长的页往往也是经常要访问的页。将这些页淘汰，很可能刚置换出去，又请求调用该页，致使缺页中断太频繁，严重降低内存的利用率。

FIFO 的另一个缺点是它可能会产生一种异常现象。一般来说，对于任一作业或进程，如果给它分配的内存页面数越接近于它所要求的页面数，则发生缺页的次数会越少。但使用 FIFO 算法时，有时会出现分配的页面数增多，缺页次数反而增加的现象，称为 belady 现象。

④ 最近最久未使用算法（Least Recently Used, LRU）。当需要淘汰某一页时，选择离当前时间最近的一段时间内最久没有使用过的页先淘汰。例如，考虑一个仅 460 个字节的程序的内存访问序列（10，11，104，170，73，309，185，245，246，434，458，364），页面的大小为 100 个字节，则 460 个字节应占 5 页，编号为 0~4，第 0 页字节为 0~99，第 1 页为 100~199，依此类推。得到页面的访问序列是（0，0，1，1，0，3，1，2，2，4，4，3），可简化为（0，1，0，3，1，2，4，3）。如果内

存中有 200 个字节可供程序使用，则内存提供 2 个页帧供程序使用。按照 FIFO 算法，共产生 6 次缺页中断，如表 1-2 所示。

表 1-2 FIFO 算法缺页中断

0	1	0	3	1	2	4	3
0	0	0	3	3	3	4	4
	1	1	1	1	2	2	3
×	×		×		×	×	×

按照 LRU 算法，共产生 7 次缺页中断，如表 1-3 所示。

表 1-3 LRU 算法缺页中断

0	1	0	3	1	2	4	3
0	0	0	0	1	1	4	4
	1	1	3	3	2	2	3
×	×		×	×	×	×	×

⑤ 最近没有使用页面置换算法（No Used Recently，NUR）。在需要置换某一页时，从那些最近的一个时期内未被访问的页任选一页置换。只要在页表中增设一个访问位即可实现。当某页被访问时，访问位置为 1，否则访问位置为 0。系统周期性地对所有引用位清零。当需淘汰一页时，从那些访问位为零的页中选一页进行淘汰。

⑥ 最优置换算法。选择那些永久不使用的，或者在最长时间内不再被访问的页面置换出去。因为要确定哪个页面是未来最长时间内不再被访问的，目前来说很难估计，所以，该算法通常用来评价其他算法。

⑦ 时钟页面替换算法（Clock）。使用页表中的引用位，将作业已调入内存的页面链成循环队列，用一个指针指向循环队列中的下一个将被替换的页面。其实现方法如下：一个页面首次装入内存时，其引用位置 1；在内存中的任何一个页面被访问时，其引用位置 1；淘汰页面时，存储管理从指针当前指向的页面开始扫描循环队列，把所遇到的引用位是 1 的页面的引用位清 0，并跳过这个页面；把所遇到的引用位是 0 的页面淘汰掉，指针推进一步；扫描循环队列时，如果遇到的所有页面的引用位均为 1，则指针就会绕整个循环队列一圈，将碰到的所有页面的引用位清 0；指针停在起始位置，并淘汰掉这一页，然后指针推进一步。

在本题中，从题干可知，作业共有 4 个页面的主存空间，其中一个已被程序本身占用，所以在读取变量时可用的页面数只有 3 个。每个页面可存放 200 个整数变

量, 程序中 A 数组共有 $100 \times 100 = 10000$ 个变量。按行存放时, 每个页面调入的 200 变量刚好是程序处理的 200 个变量, 所以缺页次数为 $10000/200 = 50$ 。而按列存放时, 虽然每个页面调取数据时, 同样也读入了 200 个变量, 但这 200 个变量中, 只有 2 个是近期需要访问的 (如: 第 1 个页面调入的是 $A[*,1]$ 与 $A[*,2]$, 但程序近期需要访问的变量只有 $A[1,1]$ 和 $A[1,2]$), 所以缺页次数为: $10000/2 = 5000$ 。

试题答案

(3) A (4) C

试题 6 (2011 年 11 月试题 1)

操作系统为用户提供了两类接口: 操作一级和程序控制一级的接口, 以下不属于操作一级的接口是 (1)。

(1) A. 操作控制命令

B. 系统调用

C. 菜单

D. 窗口

试题分析

操作系统是用户和计算机之间的接口, 用户通过操作系统的帮助可以快速、有效和安全可靠地使用计算机各类资源。通常操作系统提供两类接口, 分别是程序一级的接口 (程序接口) 和操作一级的接口 (联机用户接口和脱机用户接口)。

用户与操作系统的接口通常是由“命令”和“系统调用”的形式表现出来的。命令是提供给用户在键盘终端上使用 (命令接口), 系统调用是用户在编程时使用 (程序接口)。

在不同的系统中, 系统调用的实现方式可能不同, 但大体上都可以把系统调用的执行过程分成以下几步。

① 设置系统调用号和参数

在一个系统中, 往往都设置了许多条系统调用命令, 并赋予每条系统调用命令一个唯一的系统调用号。设置系统调用方式有 2 种方式。

- 直接将参数送入相应的寄存器中, 这是最简单的一种方式。这种方式的主要问题是由于寄存器数量有限, 从而限制了设置参数的数目。
- 参数表方式。将系统调用所需要的参数, 放入一张参数表中, 再将该参数表的指针放在某个规定的寄存器中。

② 系统调用命令的一般性处理

为了使不同系统调用能方便地转向相应的命令处理程序，在系统中配置了一张系统调用入口表。表中每个表目都对应一条系统调用命令，核心可利用系统调用号去查找该表，就可以找到相应命令处理程序的入口地址而去执行它。

③ 系统调用命令处理程序的处理过程

为了提供系统调用的功能，操作系统内必须有事先编制好的实现这些功能的子程序或过程。这些程序是操作系统程序模块的一部分，且不能直接被用户程序调用。

程序员给定了系统调用名和参数之后是怎样得到系统服务的呢？这需要有一个类似于硬件终端处理的中断处理机构。当用户使用系统调用时，产生一条相应的指令，处理机在执行到该指令时发生相应的中断，并发出有关信号给该处理机构。该处理机构在收到了处理机发来的信号后，启动相关的处理程序去完成该系统调用所要求的功能。

在系统中为控制系统调用服务的机构称为陷阱处理机构。与此相对应，把由于系统调用引起处理中断的指令为陷阱指令。在操作系统中，每个系统调用都对应一个功能号。在陷阱指令中必须包括对应系统调用的功能号。而且，在有些陷阱指令中，还带有传递给陷阱处理机构和内部处理程序的有关参数。

为了实现系统调用，系统设计人员还必须为实现各种系统调用功能的子程序编造入口地址表，每个入口地址都与相应的系统子程序名相对应。然后，由陷阱处理程序把陷阱指令中所包含的功能号与该入口地址表的有关项对应起来，从而由系统调用功能号驱动有关系统子程序执行。

由于在系统调用处理结束之后，用户程序还需利用系统调用的返回结果继续执行，因此，在进入系统调用处理之前，陷阱处理机构还需保存处理机现场。再者，在系统调用处理结束之后，陷阱处理机构还要回复处理机现场。在操作系统中，处理机的现场一般被保护在特定的内存区或寄存器中。

试题答案

(1) B

试题 7 (2011 年 11 月试题 2~4)

进程 P1、P2、P3、P4 和 P5 的前趋图如图 1-4 所示。

若用 PV 操作控制进程 P1~P5 并发执行的过程，则需要设置 5 个信号量 S1、S2、S3、S4 和 S5，进程间同步所使用的信号量标注在图 1-4 中的边上，且信号量 S1~S5

的初值都等于零，初始状态下进程 P1 开始执行。在如图 1-5 所示的 PV 操作示意图中 a、b 和 c 处应分别填写 (2)；d 和 e 处应分别填写 (3)，f 和 g 处应分别填写 (4)。

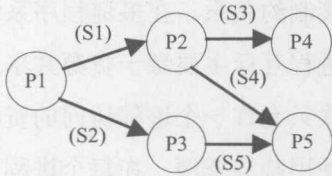


图 1-4 前趋图

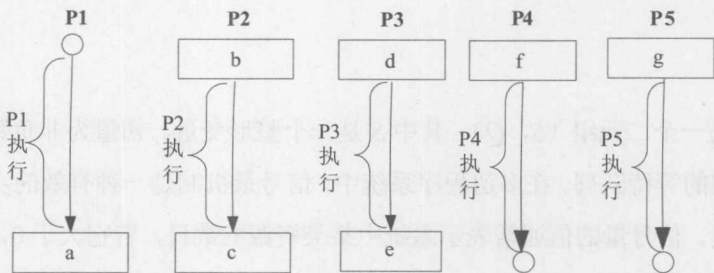


图 1-5 PV 操作示意图

- (2) A. V(S1) V(S2)、P(S1)和 V(S3) V(S4)

B. P(S1) V(S2)、P(S1)和 P(S2) V(S1)

C. V(S1) V(S2)、P(S1)和 P(S3) P(S4)

D. P(S1) P(S2)、V(S1)和 P(S3) V(S2)
- (3) A. P(S1) 和 V(S5)

B. V(S1) 和 P(S5)

C. P(S2) 和 V(S5)

D. V(S2) 和 P(S5)
- (4) A. P(S3)和 V(S4) V(S5)

B. P(S3)和 P(S4) P(S5)

C. V(S3)和 V(S4) V(S5)

D. V(S3)和 P(S4) P(S5)

试题分析

在多道程序系统中，由于资源共享与进程合作，使各进程之间可能产生两种形式的制约关系，一种是间接相互制约，例如，在仅有一台打印机的系统中，有两个进程 A 和 B，如果进程 A 需要打印时，系统已将打印机分配给进程 B，则进程 A 必须阻塞；一旦进程 B 将打印机释放，系统便将进程 A 唤醒，使之由阻塞状态变为就绪状态；另一种是直接相互制约，例如，输入进程 A 通过单缓冲区向进程 B 提供数据。当该缓冲区为空时，进程 B 不能获得所需的数据而阻塞，一旦进程 A 将数据送