

## 1.单选题

1.11

避免大材小用，能提高内存利用率的动态分区分配算法是()。

- a 首次适应算法
- b 循环首次适应算法
- c 最佳适应算法
- d 以上都是

首次适应算法先分配低地址部分的内存空间，容易使低地址部分留下小分区，而高地址部分大空闲区较多。当进程请求大空间时，要找到合适的空闲分区，时间开销大。而低地址部分的空闲分区反复被划分，可能留下许多难以利用的很小的空闲分区，这种小空闲区被称为外部碎片或外碎片。故不选 A 和 D。循环首次适应算法的优点是：空闲区分布均匀，查找开销较小。缺点是容易使系统缺乏大空闲区。故不选 B。最佳适应算法的优点是避免大材小用，能提高内存利用率。故选 C。

1.21

如果程序需要的数据存储存储在高速缓存中，程序执行期间需要()个周期访问它们。

- a 0~9
- b 0~10
- c 1~9
- d 1~10

如果程序需要的数据存储的高速缓存中,程序执行期间需要 1~10 个周期访问它们。故选 D。

### 1.31

在为进程分配内存空间时,不再每次从链首开始查找合适的空闲分区,而是从上次找到的空闲分区的下一个空闲分区开始查找的动态分区分配算法是()。

- a 首次适应算法
- b 循环首次适应算法
- c 最佳适应算法
- d 以上都是

首次适应算法在进行内存分配时,从链首开始顺序查找,直至找到一个能满足进程大小要求的空闲分区为止。故不选 A 和 D。循环首次适应算法为进程分配内存空间时,不再每次从链首开始查找,而是从上次找到的空闲分区的下一个空闲分区开始查找,直到找到第一个能满足要求的空闲分区,并从中划出一块与请求的大小相等的内存空间分配给进程。故选 B。最佳适应算法每次为作业分配内存,总是把大小与进程所请求的内存空间大小最接近的空闲分区分配给进程。故不选 C。

### 1.41

编译程序事先已知程序在内存中的驻留位置的装入方式是()。

- a 绝对装入方式
- b 可重定位装入方式
- c 静态重定位装入方式

- d 动态运行时装入方式

绝对装入方式，编译程序事先已知程序在内存中的驻留位置，编译时产生物理地址的目标代码，绝对装入程序按照装入模块的物理地址将程序和数据装入内存。故选 A。可重定位装入方式，也叫静态重定位装入方式，如果编译时不知道目标程序将驻留在内存的什么位置，则编译时就必须生成可重定位的代码。故不选 B、C。动态运行时装入，也叫动态重定位，进程在装入内存后，还可能从内存的一个区域移动到另一个区域。故不选 D。

1.51

以下关于页表的描述，不正确的是()。

- a 页表是系统为进程建立的数据结构
- b 实现从页框号到页号的映射
- c 每个进程有一个页表
- d 页表在内存中连续存放

页表是系统为进程建立的数据结构。故不选 A。页表的作用是实现从页号到页框号的映射。故选 B。在基本的分页机制中，每个进程有一个页表，进程的每一个页在页表中有一个对应的页表项。故不选 C。页表在内存中连续存放。故不选 D。

1.61

为了能将用户地址空间中的逻辑地址变换为内存空间中的物理地址，在系统中必须设置()。

- a 进程 PCB

- b 页表寄存器
- c 物理地址寄存器
- d 地址变换机构

为了能将用户地址空间中的逻辑地址变换为内存空间中的物理地址,在系统中必须设置地址变换机构,该机构的基本任务是实现逻辑地址到物理地址的变换。故选 D。进程 PCB、页表寄存器、物理地址寄存器都是地址变换机构的组成部分。故不选 ABC。

1.71

实现文件存储的几种方法中,不适合大容量的磁盘的方法是()。

- a 连续分配
- b 使用磁盘链接表的分配
- c 使用内存的链接表分配
- d i-结点

连续分配,把每个文件作为一连串连续数据块存储在磁盘上,实现简单,但随着时间的推移,磁盘会变得零碎。删除文件所释放的簇形成“空洞”,故不选 A。使用磁盘链接表分配,为每个文件构造簇的链接表,每个簇开始的几个字节用于存放下一个簇的簇号,簇的其他部分存放数据,每个文件可以存放在不连续的簇中,这一方法可以充分利用每个簇,不会因为磁盘碎片而浪费存储空间,故不选 B。使用内存的链接表分配,是将文件所在的磁盘的簇号存放在内存的表中。此方法不适合大容量的磁盘。故选 C。i-结点为每个文件赋予一个被称为 i 结点的

数据结构，其中列出了文件属性和文件块的磁盘地址。给定一个文件的 i 结点，就有可能找到文件的所有块。故不选 D。

1.81

现在的计算机系统基本上都把页表存放在内存中，CPU 要访问内存读写数据或读取指令，必须访问()内存。

- a 一次
- b 两次
- c 三次
- d 四次

现在的计算机系统基本上都把页表存放在内存中，CPU 要访问内存读写数据或读取指令，必须访问两次内存。第一次访问内存，从内存页表中获取访存单元所在的页框号，以形成访存单元的物理地址。第二次访存是根据计算出的物理地址实现对内存单元的访问，读写数据或读取指令。故选 B。

1.91

从用户角度看，引入文件系统的主要目的是()。

- a 实现对文件的按名存取
- b 实现虚拟存储
- c 保存用户文档
- d 保存系统文档

用户关心的是文件怎样命名，可进行哪些操作等。从用户角度看，引入文件系统的主要目的是实现对文件的按名存取。故选 A。

1.10<sub>1</sub>

关于页与页框，描述不正确的是()。

- a 页也称为页框
- b 页是将一个进程的逻辑地址空间分成若干个大小相等的片
- c 页框是将物理内存空间分成与页大小相同的若干个存储块
- d 页框也称为页帧

页（Page），将一个进程的逻辑地址空间分成若干个大小相等的片，称为页。

页框，将物理内存空间分成与页大小相同的若干个存储块，称为页框或页帧。故选 A。

1.11<sub>1</sub>

仅适用于单用户、单任务的分配方式是()。

- a 动态分区分配方式
- b 固定分区分配方式
- c 单一连续区分配方式
- d 以上均可

动态分区分配方式，系统动态地对内存进行划分，根据进程需要的空间大小分配内存。故不选 A 和 D。固定分区分配方式，将内存用户区划分成若干个固定大小的区域，每个区域中驻留一道程序。故不选 B。单一连续区分配方式，内存中只有一个用户区，任意时刻内存中只能装入一道程序，这种分配方式只适用于单用户、单任务系统。故选 C。

1.121

关于段页式存储管理描述不正确的是()。

- a 将用户进程的逻辑空间先划分成若干个段，每个段再划分成若干个页。
- b 进程以页为单位在物理内存中连续存放
- c 操作系统为每个进程建立一个段表，为进程的每个段建立一个页表。
- d 进程段表的每一个段表项存放某个段的页表起始地址和页表长度。

在段页式存储管理系统中，将用户进程的逻辑空间先划分成若干个段，每个段再划分成若干个页。故不选 A。进程以页为单位在物理内存中离散存放。故选 B。

为了实现段页式存储管理的地址映射，操作系统为每个进程建立一个段表，为进程的每个段建立一个页表。故不选 C。进程段表的每一个段表项存放某个段的页表起始地址和页表长度。故不选 D。

1.131

下面哪个选项不是虚拟存储技术带来的好处()。

- a 提高内存利用率

- b 减少 CPU 在有效访存上的时间开销，提高访存速度。
- c 提高多道程序度
- d 把逻辑地址空间和物理地址空间分开

虚拟存储技术带来的好处：（1）提高内存利用率；故不选 A。（2）提高多道程序度；故不选 C。（3）把逻辑地址空间和物理地址空间分开，使程序员不再关心物理内存的容量对编程的限制。故不选 D。减少 CPU 在有效访存上的时间开销，提高访存速度，是引入快表的好处。故选 B。

1.141

用目录树组织文件系统时，需要用某种方法指明文件名。常用的方法有()种。

- a 一
- b 两
- c 三
- d 四

用目录树组织文件系统时，需要用某种方法指明文件名。常用的方法有两种：绝对路径名和相对路径名。故选 B。

1.151

下面有关程序的链接，描述不正确的是()。

- a 链接程序属于操作系统的构成部分



- b 将编译后的目标模块装配成一个可执行的程序
- c 可分为静态链接和动态链接
- d 它为操作系统提供可装入的程序模块

链接程序不属于操作系统的构成部分，故选 A。链接程序要解决的问题是将编译后的目标模块装配成一个可执行的程序，故不选 B。根据链接进行的时间和实现方式的不同，可以把链接分为静态链接和动态链接，故不选 C。链接程序为操作系统提供可装入的程序模块，故不选 D。

1.161

树形目录的最高层为根目录，最底层为()。

- a 子目录
- b 当前目录
- c 用户目录
- d 文件

树形目录也称多级目录，最高层为根目录，最底层为文件。用户可以创建任意数量的子目录。故选 D。

1.171

根据离散内存管理分配内存空间的基本单位不同，将其分为()种管理方式。

- a 1

- b 2
- c 3
- d 4

根据离散内存管理分配内存空间的基本单位不同,分为 3 种管理方式:分页存储管理、分段存储管理和段页式存储管理。故选 C。

1.181

容易使系统缺乏大空闲区的动态分区分配算法是()。

- a 首次适应算法
- b 循环首次适应算法
- c 最佳适应算法
- d 以上都是

首次适应算法先分配低地址部分的内存空间,容易使低地址部分留下小分区,而高地址部分大空闲区较多。当进程请求大空间时,要找到合适的空闲分区,时间开销大。而低地址部分的空闲分区反复被划分,可能留下许多难以利用的很小的空闲分区,这种小空闲区被称为外部碎片或外碎片。故不选 A 和 D。循环首次适应算法的优点是:空闲区分布均匀,查找开销较小。缺点是容易使系统缺乏大空闲区。故选 B。最佳适应算法的优点是避免大材小用,能提高内存利用率。但是,采用最佳适应算法容易留下难以利用的小空闲区。故不选 C。

1.191

为了减少 CPU 在有效访存上的时间开销，提高访存速度，在硬件上引入了()。

- a 快表机制
- b 页表机制
- c 两级页表
- d 多级页表

为了减少 CPU 在有效访存上的时间开销，提高访存速度，在硬件上引入了快表机制。故选 A。页表是系统为进程建立的数据结构，作用是实现从页号到页框号的映射，即记录了相应页在内存中对应的页框号。在基本的分页机制中，每个进程有一个页表，进程的每一个页在页表中有一个对应的页表项。故不选 B。从系统性能考虑，不希望用大的连续地址空间存放页表，解决的办法是把页表再分页，形成两级或多级页表。故不选 CD。

1.201

下面有关虚拟存储器，描述不正确的是()。

- a 具有请求调入功能
- b 具有置换功能
- c 能从物理上对内存容量进行扩充
- d 进程无需全部装入，只要装入一部分就可运行

虚拟存储器是指具有请求调入功能和置换功能。故不选 AB。能从逻辑上对内存容量进行扩充的一种存储器系统。故选 C。在虚拟存储器系统中，进程无需全部装入，只要装入一部分即可运行。故不选 D。

1.21<sub>1</sub>

磁盘空间管理使用位图()。

- a 记录文件
- b 保护文件
- c 访问文件
- d 记录空闲块

磁盘空间管理采用空闲簇链接表和位图两种方法记录空闲块。故选 D。

1.22<sub>1</sub>

存储器系统是一个层次结构，设计算法时尽可能把最近将要访问的指令或数据存储在()的地方。

- a 层次较低
- b 层次较高
- c 容量较大
- d 容量较小

在编写程序、设计算法时尽可能把最近将要访问的指令或数据存储在层次较高的地方，以便让 CPU 更快地访问到它们。故选 B。

1.23<sub>1</sub>

实现文件存储的几种方法中，必须把整个表都存放在内存中的方法是()。

- a 连续分配
- b 使用磁盘链接表的分配
- c 使用内存的链接表分配
- d i-结点

连续分配，把每个文件作为一连串连续数据块存储在磁盘上，实现简单，但随着时间的推移，磁盘会变得零碎。删除文件所释放的簇形成“空洞”，故不选 A。使用磁盘链接表分配，为每个文件构造簇的链接表，每个簇开始的几个字节用于存放下一个簇的簇号，簇的其他部分存放数据，每个文件可以存放在不连续的簇中，这一方法可以充分利用每个簇，不会因为磁盘碎片而浪费存储空间，故不选 B。使用内存的链接表分配，是将文件所在的磁盘的簇号存放在内存的表中。该方法的缺点是必须把整个表都存放在内存中。故选 C。i-结点为每个文件赋予一个被称为 i 结点的数据结构，其中列出了文件属性和文件块的磁盘地址。给定一个文件的 i 结点，就有可能找到文件的所有块。故不选 D。

1.24<sub>1</sub>

只要路径名的第一个字符是()，则这个路径就是绝对路径。

- a \

- b /
- c 盘符
- d 分隔符

绝对路径名由从根目录到文件的路径组成,并且是唯一的,绝对路径名的第一个字符是分隔符。故选 D。在 Windows 系统中用 “\” 分隔,故不选 A。在 UNIX 系统中,用 “/” 分隔,故不选 B。

1.251

关于段的描述不正确的是()。

- a 每个段由一个从 0 到最大线性地址的逻辑地址空间构成
- b 各个段的长度可以是 0 到最大值之间的任何一个值
- c 不同段的长度可以不相同
- d 段的长度在进程运行期间不能改变

把分别存放逻辑上相关的信息、相互独立的逻辑地址空间称为一个段,每个段由一个从 0 到最大线性地址的逻辑地址空间构成。故不选 A。各个段的长度可以是 0 到最大值之间的任何一个值,故不选 B。不同段的长度可以不相同,故不选 C。段的长度在进程运行期间可以改变,故选 D。

1.261

分页地址变换机构中,物理地址的计算公式为()。

- a 页框大小+页内偏移量

- b 页框号+页内偏移量
- c 页框大小\*页框号+页内偏移量
- d 页表起始地址+页表项长度\*页号

页框号和页内偏移地址送物理地址寄存器，计算物理地址。物理地址=页框大小\*页框号+页内偏移量。故选 C。页号对应的页表项起始地址=页表起始地址+页表项长度\*页号。故不选 D。

1.27<sub>1</sub>

如果程序需要的数据是存放在 CPU 的寄存器中，程序执行期间在()周期内就可以访问到它们。

- a 0 个
- b 1 个
- c 2 个
- d 3 个

如果程序需要的数据是存放在 CPU 的寄存器中，程序执行期间在零个周期内就可以访问到它们。故选 A。

1.28<sub>1</sub>

将一个用户源程序变为一个可在内存中执行的程序，须经过()个阶段。

- a 1

- b 2
- c 3
- d 4

将一个用户源程序变为一个可在内存中执行的程序，须经过编译、链接和装入 3 个阶段，故选 C。

1.29<sub>1</sub>

要求空闲分区链以地址递增的顺序链接的动态分区分配算法是()。

- a 首次适应算法
- b 循环首次适应算法
- c 最佳适应算法
- d 以上都是

首次适应算法要求空闲分区链以地址递增的顺序链接，故选 A。循环首次适应算法由首次适应算法演变而来，为进程分配内存空间时，不再每次从链首开始查找，而是从上次找到的空闲分区的下一个空闲分区开始查找，故不选 B。最佳适应算法，每次为作业分配内存时，总是把大小与进程所请求的内存空间大小最接近的空闲分区分配给进程。为了加速寻找，该算法要求将所有的空闲区按分区大小递增顺序形成一个空闲区链，故不选 C。

1.30<sub>1</sub>

源程序中的地址通常是()。



- a 符号地址
- b 重定位地址
- c 逻辑地址
- d 物理地址

源程序中的地址通常是符号地址。故选 A。编译器将符号地址变成可重定位地址，通常是相对于本模块开始位置的地址。故不选 B。逻辑地址是相对于整个可执行程序起始地址的地址。故不选 C。逻辑地址映射为物理地址，物理地址是程序在内存中的实际地址。故不选 D。

1.31<sub>1</sub>

快表用来存放()。

- a 整个页表
- b 最近被访问过的页表项
- c 将要被访问的页表项
- d 两级页表

快表也称转换后备缓冲 ( TLB )，是为了提高 CPU 访存速度而采用的专用缓存，用来存放最近被访问过的页表项。故选 B。

1.32<sub>1</sub>

R1、R2 为空闲分区，按照下图，被释放的内存区域（回收区）与相邻的空闲区合并后，新空闲区的起始地址为 R1 对应的结点的分区起始地址的是()。

- a 图 a)和图 b)
- b 图 b)和图 c)
- c 图 a)和图 c)
- d 图 a)、图 b)和图 c)

图 a)仅回收区的前面有相邻空闲分区 R1，把回收区与 R1 合并成一个空闲分区，将 R1 的起始地址作为新分区的起始地址，分区大小修改为 R1 与回收区大小之和。图 b)仅回收区的后面有相邻空闲分区 R2，把回收区与 R2 合并成一个空闲分区，将回收区的起始地址作为新分区的起始地址，分区大小修改为 R2 与回收区大小之和。故不选 ABD。图 c)回收区的前、后有相邻空闲分区 R1、R2，把回收区与 R1、R2 合并成一个空闲分区，将 R1 的起始地址作为新分区的起始地址，分区大小修改为 R1、R2 与回收区三者大小之和，删除与 R2 分区对应的空闲分区结点。故选 C。

1.331

R1、R2 为空闲分区，如下图，被释放的内存区域（回收区）与相邻的空闲区合并后，空闲分区的个数减少的是()。

- a 图 a)和图 b)
- b 图 a)
- c 图 b)

- d 图 c)

如果被释放区域与其他空闲区间相邻,合并时需考虑以下 3 种情况:如上图,图 a)和图 b),回收后空闲分区链中结点的个数不变。故不选 ABC。图 c),回收后空闲分区链中结点的个数减少。故选 D。

1.34<sub>1</sub>

需要用到重定位寄存器的装入方式是()。

- a 绝对装入方式
- b 可重定位装入方式
- c 静态重定位装入方式
- d 动态运行时装入方式

在采用动态运行时装入方式的系统中,在进程运行访存的过程中才进行地址转换,需要重定位寄存器的支持。当进程获得 CPU 时,系统把该进程在内存的起始地址存入重定位寄存器,进程在运行过程中访存时,通过重定位寄存器与被访问单元的逻辑地址计算出物理地址。故选 D。

1.35<sub>1</sub>

一般页的大小为 2 的整数次幂,目前的计算机系统中,大多选择()大小的页。

- a 1KB
- b 2KB

- c 3KB
- d 4KB

在分页系统中,页的大小是由机器的体系结构和操作系统共同决定的。若页较小,页表较长,页表需要占的内存空间较大,而且页换入、换出频繁;若页较大,页内碎片会增大。一般页的大小为 2 的整数次幂,目前的计算机系统中,大多选择 4KB 大小的页。故选 D。

1.36<sub>1</sub>

文件的两级目录包括主目录和()。

- a 根目录
- b 当前目录
- c 用户目录
- d 子目录

在两级目录结构中,目录被分成两级,第一级称为主目录,给出了用户名和用户子目录所在的物理位置。第二级称为用户目录,给出了该用户所有文件的文件控制块。故选 C。

1.37<sub>1</sub>

常用的文件存取方式有()种。

- a 一
- b 两

- c 三
- d 四

用户通过对文件的存取来完成对文件的各种操作，常用的文件存取方式有两种：顺序存取和随机存取。故选 B。

1.381

实现文件存储的几种方法中，随机存取缓慢，要获得文件的第 n 块，每一次都要从头开始读取的方法是()。

- a 连续分配
- b 使用磁盘链接表的分配
- c 使用内存的链接表分配
- d i-结点

连续分配，把每个文件作为一连串连续数据块存储在磁盘上，实现简单，但随着时间的推移，磁盘会变得零碎。删除文件所释放的簇形成“空洞”，故不选 A。使用磁盘链接表分配，为每个文件构造簇的链接表，每个文件可以存放在不连续的簇中，这一方法可以充分利用每个簇，不会因为磁盘碎片而浪费存储空间，缺点是随机存取缓慢，要获得文件的第 n 块，每一次都要从头开始读取前面的 n-1 块。故选 B。使用内存的链接表分配，访问文件时，只需从内存文件分配表中顺着某种链接关系查找簇的簇号，根据簇号查找到文件的所有块。故不选 C。i-结点为每个文件赋予一个被称为 i 结点的数据结构，其中列出了文件属性和文件块的磁盘地址。给定一个文件的 i 结点，就有可能找到文件的所有块。故不选 D。

1.39<sub>1</sub>

页表项中的字段，用来标识页是否在内存中的是()。

- a 状态位 P
- b 访问字段 A
- c 修改位 M
- d 保护位

状态位 P：标识页是否在内存中。可规定 0 不在，1 在。故选 A。访问字段 A：用于记录页最近被访问的情况。系统总是希望根据 A 的值把最近、最久未访问的页换出到外存。故不选 B。修改位 M：标识页最近是否被修改过。在换出页时，必须把最近修改过的页写回外存。故不选 C。保护位：标识页的访问权限，如 1 表示可读可写，0 表示只读。故不选 D。

1.40<sub>1</sub>

为每一个页框设一个表项，表项中存放进程号和页号的表是 ( )。

- a 页表          b 两级页表          c 多级页表          d 反置页表

页表是系统为进程建立的数据结构，作用是实现从页号到页框号的映射。在基本的分页机制中，每个进程有一个页表，进程的每一个页在页表中有一个对应的页表项。故不选 A。从系统性能考虑，不希望用大的连续地址空间存放页表，解决的办法是把页表再分页，形成两级或多级页表。故不选 BC。现代系统中可能存在大量进程，每个进程都允许很大的逻辑地址空间，因而进程可能拥有一个很大

的页表，这些页表会占用大量的物理内存空间。为了解决这个问题，可以使用反置页表，为每一个页框设一个表项，表项中存放进程号和页号。故选 D。

## 2.多选题

2.12

下面哪些算法属于动态分区分配算法()。

- a 首次适应算法
- b 平均分配算法
- c 循环首次适应算法
- d 按比例分配算法
- e 最佳适应算法

动态分区分配算法包含 3 种：首次适应算法、循环首次适应算法、最佳适应算法，故选 ACE。平均分配算法和按比例分配算法属于页框分配算法，故不选 BD。

2.22

实现文件存储的常用方式有()。

- a 连续分配
- b 使用磁盘链接表的分配
- c 使用内存的链接表分配
- d i-结点
- e 以上都是

实现文件存储的常用方式有：连续分配、使用磁盘链接表的分配、使用内存的链接表分配和 i-结点。故选 ABCDE。

2.32

虚拟存储系统的主要特征有()。

- a 离散性
- b 连续性
- c 多次性
- d 对换性
- e 虚拟性

虚拟存储系统具有以下几个主要特征：离散性。指进程可以分散地存储在物理内存中，离散性是实现虚拟存储管理的基础。故选 A，不选 B。多次性。指不必把进程一次性全部装入内存，可分多次装入。故选 C。对换性。指在内存中的进程可以换出，以腾出内存空间换入外存中的进程。故选 D。虚拟性。指虚拟存储系统为用户提供了比实际物理内存大的逻辑内存空间，程序员不必在编程时受物理内存空间大小的限制。虚拟性是实现虚拟存储系统的最重要目标。故选 E。

2.42

下面哪些属于动态链接的优点()。

- a 程序运行速度较快
- b 占用的内外存空间较大
- c 程序开发不够灵活
- d 节省内、外存空间



- e 方便程序开发

程序运行速度较快是静态链接的优点，故不选 A。占用的内外存空间较大，程序开发不够灵活是静态链接的缺点，故不选 BC。节省内、外存空间，方便程序开发是动态链接的优点，故选 DE。

2.52

目录结构包括()。

- a 单层目录
- b 两级目录
- c 三级目录
- d 多级目录
- e 树形目录

文件目录的组织和管理是文件管理的一个重要方面，包括单层目录、两级目录和树形目录。故选 ABE。

### 3.填空题

3.13

常用的文件存取方式有两种：\_\_\_\_\_和随机文件。

答案：顺序文件

3.23

目录是文件系统中实现\_\_\_\_\_的重要数据结构。

答案：按名存取

3.3<sub>3</sub>

最近最久未使用（LRU）置换算法选择的换出页是\_\_\_\_\_的页

答案：最近最久未使用

3.4<sub>3</sub>

CPU 的页表寄存器用于存放当前进程的\_\_\_\_\_

答案：页表起始地址

3.5<sub>3</sub>

在使用文件之前，必须先\_\_\_\_\_文件

答案：打开

3.6<sub>3</sub>

\_\_\_\_\_主要作用是在访问内存过程中发现缺页时产生缺页异常信息，使

CPU 中断当前控制流的执行，转去执行操作系统的缺页异常处理程序，完成请

求调页。

答案：缺页异常机构

3.7<sub>3</sub>

连续分配是指操作系统分配内存时，为每个进程分配一块\_\_\_\_\_的内存空

间，连续分配方式有 3 种，即单一连续区分配方式、固定分区分配方式和动态分

区分配方式。

答案：物理地址连续

3.8<sub>3</sub>

在分段存储管理系统中，段表项中包含的基本内容有段号、\_\_\_\_\_和段大

小。

答案：段基址

## 4.简答题

4.15.2

简述分段存储管理的原理。

答案：在分段的存储管理方式中，进程的地址空间被划分成若干个段。每个段定义了一组逻辑信息，每个段的大小由相应的逻辑信息组的长度确定，段的大小不一样，每个段的逻辑地址从 0 开始，采用一段连续的地址空间。系统为每个段分配一个连续的物理内存区域，各个不同的段可以离散地放入物理内存不同的区域。系统为每个进程建立一张段表，段表的每一个表项记录的信息包括段号、段长和该段的基址，段表存放在内存中。

4.25.2

简述 i-结点。

答案：为每个文件赋予一个被称为 i 结点的数据结构，其中列出了文件属性和文件块的磁盘地址。给定一个文件的 i 结点，就有可能找到文件的所有块。系统打开文件时，将文件的 i 结点从磁盘读入内存。当访问文件时，系统先根据文件名搜索文件所在的目录文件，从该文件对应的目录项中找到文件的 i 结点号，根据 i 结点号从磁盘中将 i 结点信息读入内存，文件在磁盘中的地址信息都存放在 i 结点中。如果每个 i 结点只能存储固定数量的磁盘地址，那么当一个文件比较大，所含簇的数目太多时，i 结点将无法记录所有的簇号。一个解决方案是采用间接地址，即使一个“磁盘地址”不存放数据块，而是存放簇号。

4.35.2

请解释抖动、抖动产生的原因和预防方法。

答案：多道程序度太高，使运行进程的大部分时间都用于进行页的换入、换出，而几乎不能完成任何有效工作的状态称为抖动。

引起抖动的原因是系统中的进程数量太多，每个进程能分配到的页框太少，以至于进程运行过程中频繁请求调页。

抖动的预防：

(1) 采取局部置换策略。仅在进程自己的内存空间范围内置换页，不允许从其他进程获得新的页框。

(2) 在 CPU 调度程序中引入工作集算法。只有当每个进程在内存中都有足够大的驻留集时，才能再从外存中调入新的作业。

(3) 挂起若干进程。为了预防抖动，挂起若干进程，腾出进程占用的空间。

4.45.2

**请解释静态链接程序和动态链接程序。**

答案：静态链接：在程序运行前，用链接程序将目标模块链接成一个完整的装入模块。静态链接程序的任务一是对逻辑地址进行修改，而是变换外部调用符号。

动态链接：可将某些目标模块的链接推迟到这些模块中的函数被调用执行时才进行。即在程序执行时，若发现一个被调用模块尚未链接，再把它链接到调用者模块上。

4.55.2

**简述请求分页系统的基本原理。**

答案：请求分页系统的基本原理是，把进程的逻辑地址空间分成大小相同的页，操作系统创建进程时只把进程的一部分页调入内存。进程运行过程中访问内存、

若发现所访问的页不在内存中，则产生一个缺页异常信号，系统响应缺页异常，请求调入缺页。