**教材162页：**

2.

绝对装入方式：单道程序环境

重定位装入方式、动态运行时装入方式：多道程序环境

3.

静态链接：在程序运行之前，先将各目标模块及它们所需的库函数，链接成一个完整的装入模块（又称执行模块），以后不再拆开。

需解决：1）对相对地址进行修改；2）变换外部调用符号

5.

运行时动态链接：在执行过程中，若发现一个被调用模块尚未装入内存时，由OS去找该模块，将它装入内存，并链接到调用模块上。

优点：1）加快程序装入过程；2）节省大量内存空间

6.

在每个分区的起始部分，设置一些控制分区分配的信息，以及用于链接各分区所用的前向指针；在分区尾部设置一个后向指针，通过前后向链接指针，将所有空闲分区链成一个双向链。当分区分配出去后，把状态位由“0”改为“1”。

8.

为了实现动态分区分配，通常是将系统中的空闲分区链接成一个链。所谓顺序搜索，是指依次搜索空闲分区链上的空闲分区，去寻找一个其大小能满足要求的分区。基于顺序搜索的动态分区分配算法有如下四种:首次适应算法、循环首次适应算法、最佳适应算法和最坏适应算法。

9.

1）回收区与插入点的前一个分区相邻接，此时可将回收区与插入点的前一分区合并，不再为回收分区分配新表项，而只修改前邻接分区的大小；

2）回收分区与插入点的后一分区相邻接，此时合并两区，然后用回收区的首址作为新空闲区的首址，大小为两者之和；

3）回收区同时与插入点的前后两个分区邻接，此时将三个分区合并，使用前邻接分区的首址，大小为三区之和，取消后邻接分区的表项；

4）回收区没有邻接空闲分区，则应为回收区单独建立一个新表项，填写回收区的首址和大小，并根据其首址，插入到空闲链中的适当位置。

13.

为了提高系统的吞吐量，提高内存的利用率和处理机的利用率。

可分为整体对换（进程对换）和局部对换（分段或页面对换）。

15.

1）对换空间管理；2）程序换入；3）程序换出

17．

1）分页式存储管理；2）分段式存储管理；3）段页式存储管理

19.

页表是分页式存储管理使用的数据结构，描述该进程的各页面在内存中对应的物理块号。

作用：实现从页号到物理块号的地址映射。

20.

CPU的硬件支持（地址变换机构）

页表寄存器、物理地址寄存器、联想寄存器（快表）

21.

1）根据逻辑地址，计算出页号和页内偏移量；

2）从PTR中得到页表首址，然后检索页表，查找指定页面对应的页框号；

3）用页框号乘以页面大小获得其对应的起始地址，并将其送入物理地址的高端。

4）将页内偏移量送入物理地址低端，形成完整的物理地址。

24.

1）根据逻辑地址中的页号，查找快表中是否存在对应的页表项。

2）若快表中存在该表项，称为命中（hit），取出其中的页框号，加上页内偏移量，计算出物理地址。

3）若快表中不存在该页表项，称为命中失败，则再查找页表，找到逻辑地址中指定页号对应的页框号。同时，更新快表，将该表项插入快表中。并计算物理地址

26.

1）页是信息的物理单位，是为减少内存的碎片，出于系统管理需要；段是信息的逻辑单位，是出于用户的需要出发。

2）页的大小固定，由系统决定；段的长度不固定，由用户决定。

3）分页的地址空间是一维的，从0开始编址，而分段的地址空间是二维的，每个段的段内地址是连续的，也从0开始编址，但段间可以不连续，它们离散地分布在内存的各个分区中。

27.

1）连续分配是指为一个用户程序分配一个连续的地址空间，包括单一和分区两种分配方式。单一方式将内存分为系统区和用户区，最简单，只用于单用户单任务操作系统;分区方式分固定和动态分区。

2）离散分配方式分为分页、分段和段页式存储管理。分页式存储管理旨在提高内存利用率，分段式存储管理旨在满足用户（程序员）的需要，段页式存储管理则将两者结合起来，具有分段系统便于实现、可共享、易于保护和动态链接等优点，又能像分页系统很好解决外部碎片及为各段可离散分配内存等问题，是比较有效的存储管理方式;

**教材189页：**

2.

1）时间局限性

- 如果程序中的某条指令一旦执行，则不久以后该指令可能再次执行；如果某数据被访问过， 则不久以后该数据可能再次被访问。

- 产生时间局限性的典型原因，是由于在程序中存在着大量的循环操作。

2）空间局限性

- 一旦程序访问了某个存储单元，在不久之后，其附近的存储单元也将被访问。

- 程序在一段时间内所访问的地址，可能集中在一定的范围之内，其典型情况便是程序的顺序执行。

4.

1） 请求分页的页表机制；2）缺页中断机构；3）地址变换机构。

5.

1）在分页请求系统的基础上增加了请求调页和页面置换功能，形成了页式虚拟存储系统。它允许只装入少数页面程序（及数据），便可启动运行；

2）在请求分段系统的基础上增加了请求调段及分段置换功能，形成了段式虚拟存储系统。它允许只装入少数（而非所有）用户程序段和数据段，便可启动运行。

8.

1）当进程要访问某个逻辑地址中的数据时，分页地址变换机构会自动地将有效地址（相对地址）分为页号和页内地址两部分，再以页号为索引去检索页表。

2）查找操作由硬件执行。在执行检索之前，先将页号与页表长度进行比较，如果页号大于或等于页表长度，则表示本次所访问的地址已超越进程的地址空间。于是，这一错误将被系统发现，并产生一地址越界中断。若未出现越界错误，则将页表始址与页号和页表项长度的乘积相加，便得到该表项在页表中的位置，于是可从中得到该页的物理块号，将之装入物理地址寄存器中。

3）将有效地址寄存器中的页内地址送入物理地址寄存器的块内地址字段中。这样便完成了从逻辑地址到物理地址的变换。

11.

1）每当程序所要访问的页面未在内存时，便向CPU发出一缺页中断。

2）中断处理程序首先保留CPU环境，分析中断原因后，转入缺页中断处理程序。

3）如果内存已满，则须先按照某种置换算法从内存中选出一页准备换出；如果此页已被修改，则必须将它写回磁盘。

4） 然后再把所缺的页调入内存，并修改页表中的相应表项，置其存在位为“1”，并将此页表项写入快表中。

5）形成所要访问数据的物理地址，再去访问内存数据。

12.

1）最佳(优)置换算法

2）先进先出（FIFO）页面置换算法

3）最近最久未使用（LRU）置换算法

4）Clock置换算法

5）改进型Clock置换算法

6）其它置换算法

13.

块数为3：

缺页9次，缺页率=3/4

块数为4：

缺页10次，缺页率=5/6

比较结果：理论上，作业物理块越多，产生的缺页次数应越少，但上述结果里，物理块越多的反而产生更多的缺页，证明出现了Belady现象。

18.

1）进程分配的物理块太少；

2）置换算法选择不当；

3）全局置换使抖动传播。