**1、离散分配方式**

**---- 连续分配**方式（分区存储管理）要求把作业放在一个连续的存储区中，因而会**形成许多“碎片”**，固定分区会产生内部碎片，动态分区会产生外部碎片。虽然可通过“紧凑”方法将**许多碎片**拼接成可用的大块空间，但需为之付出很大开销。如果允许将**一个进程直接分散的装入到许多不相邻接的分区中**，则无需再进行“紧凑”。基于这一思想而产生了**离散分配方式**。如果**离散分配**的基本单位是**页**，则称为**分页存储管理方式**；如果离散分配的基本单位是**段**，则称为**分段存储管理方式。**

**----** 在分页存储管理方式中，如果**不具备页面对换功能**，则称为基本的分页存储管理方式，或称为**纯分页**存储管理方式。它**不具有支持实现虚拟存储器的功能**，它**要求把每个作业全部装入内存后方能运行**。

**2、页面与页表**

**---- 页面和物理块**

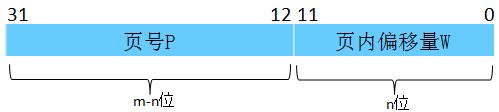
分页存储管理是将一个**进程的逻辑地址空间分成若干个大小相等的片，称为页面或页**，并为各页**加以编号**，从0开始，如第0页、第1页等。相应地，也把**内存空间分成与页面相同大小的若干个存储块**，称为**（物理）块或页框（frame）**,也加以编号，如0#块，1#块等等。在为进程分配内存时，**以块为单位将进程中的若干个页分别装入到多个可以不相邻接的物理块中。**由于进程的**最后一页经常装不满一块而形成了不可利用的碎片，称之为“页内碎片”。**

**---- 页面大小**

在分页系统中的页面大小应适中。**页面若太小**，一方面虽然可使**内存碎片减小**，有利于**提高内存利用率**。但是另一方面也会使每个进程占用较多的页面，从而导致进程的**页表过长**（页号->块号），占用大量内存，还会**降低页面换进换出**的效率。**若页面较大**，虽然可以减少页表的长度，提高页面换进换出的速度，却又会使**页内碎片增大**。因此，页面的大小选择应适中，且**页面大小**应是2的幂，通常为512B~8KB。

**---- 地址结构**

分页地址中的地址结构有两部分：前一部分是**页号P**，后一部分为页内**位移量W**（或称为**页内地址**）。如下图所示：如果逻辑地址空间是2的m次方，页面大小是2的n次方（字节），那么逻辑地址的高m-n位是页号，低n位是页内偏移量。



上述地址结构中，两部分的地址长度为32位，其中0～11位（计12位）为**页内偏移量**，即每页大小=2^12=4KB;12~31位（计20位）为页号，即（页表项的个数）最多的页数=2^20=1MB。页内偏移量也称为页内位移、页偏移或页内地址等。对于某**特定机器**，其**地址结构是一定的**。若给定一个逻辑地址空间中的地址为A，页面的大小为L，则页号P和页内地址W可按下式求得：  P=INT(A/L)        W=(A)MOD(L)     其中，INT是**整除**函数，MOD是**取余**函数。

例如，系统的页面大小是1KB（1024B），设A=2170B，则得出P=2，W=2170-2048=122.

**---- 页表**

在分页系统中，允许将进程的各个**页**离散地存储在**内存**不同的**物理块**中，但系统应能保证进程的正确运行，即能在内存中找到**每个页面所对应**的物理块，为此，系统又为每个进程建立了一张**页面映像表**，简称**页表**。

在进程地址空间内的所有页（0~n），依次在页表中有一页表项，其中记录了相应页在内存中对应的物理块号。在配置了页表后，进程执行时，通过**查找该页表**，即可找到每页在内存中的块号，页表的作用是**实现从页号到物理块号的地址映射**。即使在简单的分页系统中，也常在页表的表项中设置一**存取控制字段**，用于对该存储块中的内容加以**保护**。

**--** 当存取控制字段仅有一位时，可用来规定该存储块中的内容是允许读/写，还是只读。

**--** 当存取控制字段为二位，则可规定为读/写、只读和只执行等存取方式。

如果有一进程试图**去写一个只允许读**的存储块时，将引起操作系统的**一次中断**。如果要利用分页系统去实现虚拟存储器，还需在页表中增设状态位/修改位等**数据项**来供页面换进换出参考，具体可看**请求分页存储管理方式**。http://blog.csdn.net/dongyanxia1000/article/details/51727339

为了实现上面的方便，系统中设置了一个**页表寄存器**（**PTR，Page Table Register**），其中存放页表在内存的**起始地址F**和**页表的长度M**。**进程未执行时**，页表的起始地址和长度存放在**进程控制块（PCB）**中。当进程执行时，才将页表起始地址和长度存入页表寄存器中。

假设页面大小为L，页表长度（页表项的个数）为M，逻辑地址A通过地址变换得到物理地址E的过程如下：

**--1）**计算页号**P=(int)(A/L)**；页内偏移量**W=A%L**。

**--2）**比较页号P(理论值0~M-1)和页表长度M，若**P>= M**，则产生越界中断，否则转到下一步执行。

**--3）**页表中**页号P对应的页表项的地址**=页表起始地址F + 页号P x 页表项大小，取出该页表项中的**物理块号b**的内容。

**--4）**计算**E=bxL+W**（页面的大小就是物理块的大小，页内地址就是物理块内的地址）

**--5）**用得到的物理地址E去访问内存。

以上整个地址变换过程都是由**硬件自动完成**的。地址变换过程如下图所示：

