

NEW PLAN

# 软考高级架构师

• 一段新征程 •





# 大刚介绍





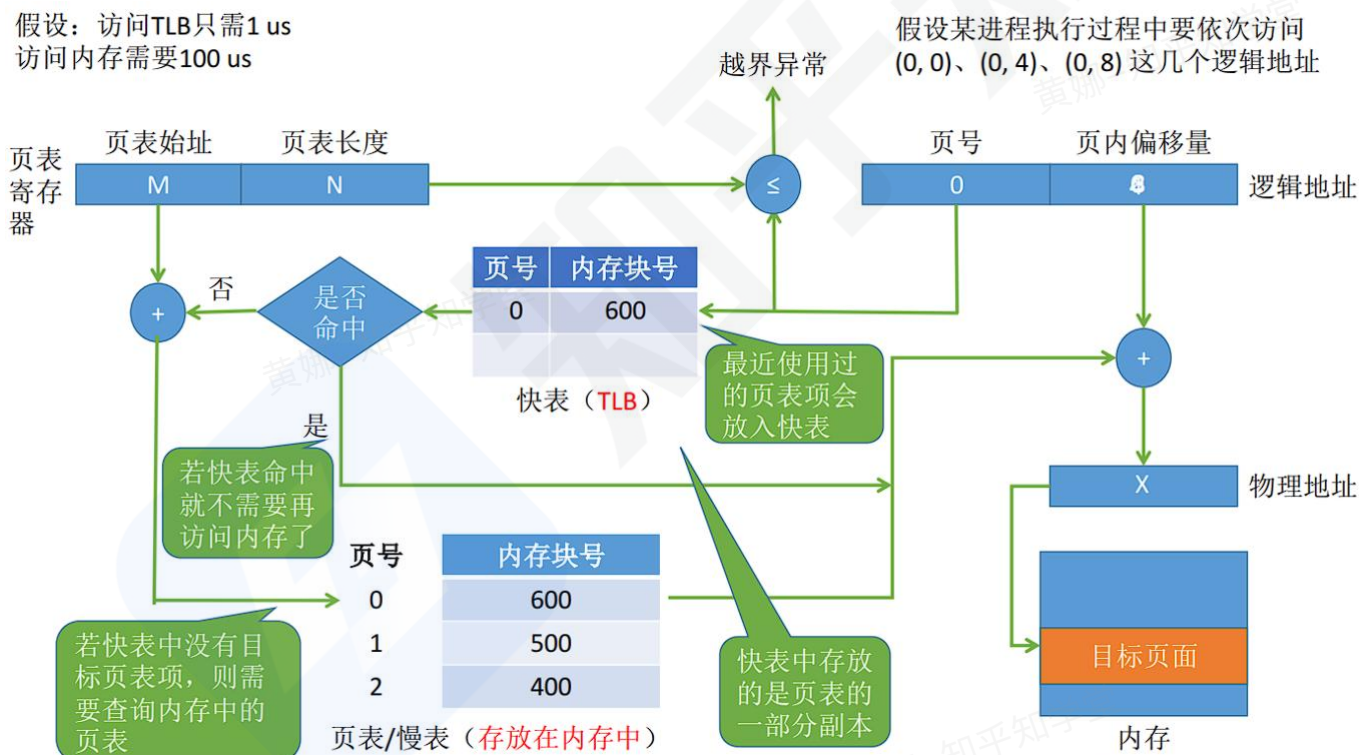
## 存储管理-快表

快表是一块小容量的相联存储器，由快速存储器组成，按内容访问，速度快，并且可以从硬件上保证按内容并行查找，一般用来存放当前访问最频繁的少数活动页面的页号(可以看成是页表的频繁访问数据的副本)。

快表是将页表存于Cache中

慢表是将页表存于内存中

因此慢表需要访问两次内存才能取出数据，而快表是访问一次Cache和一次内存，因此更快。

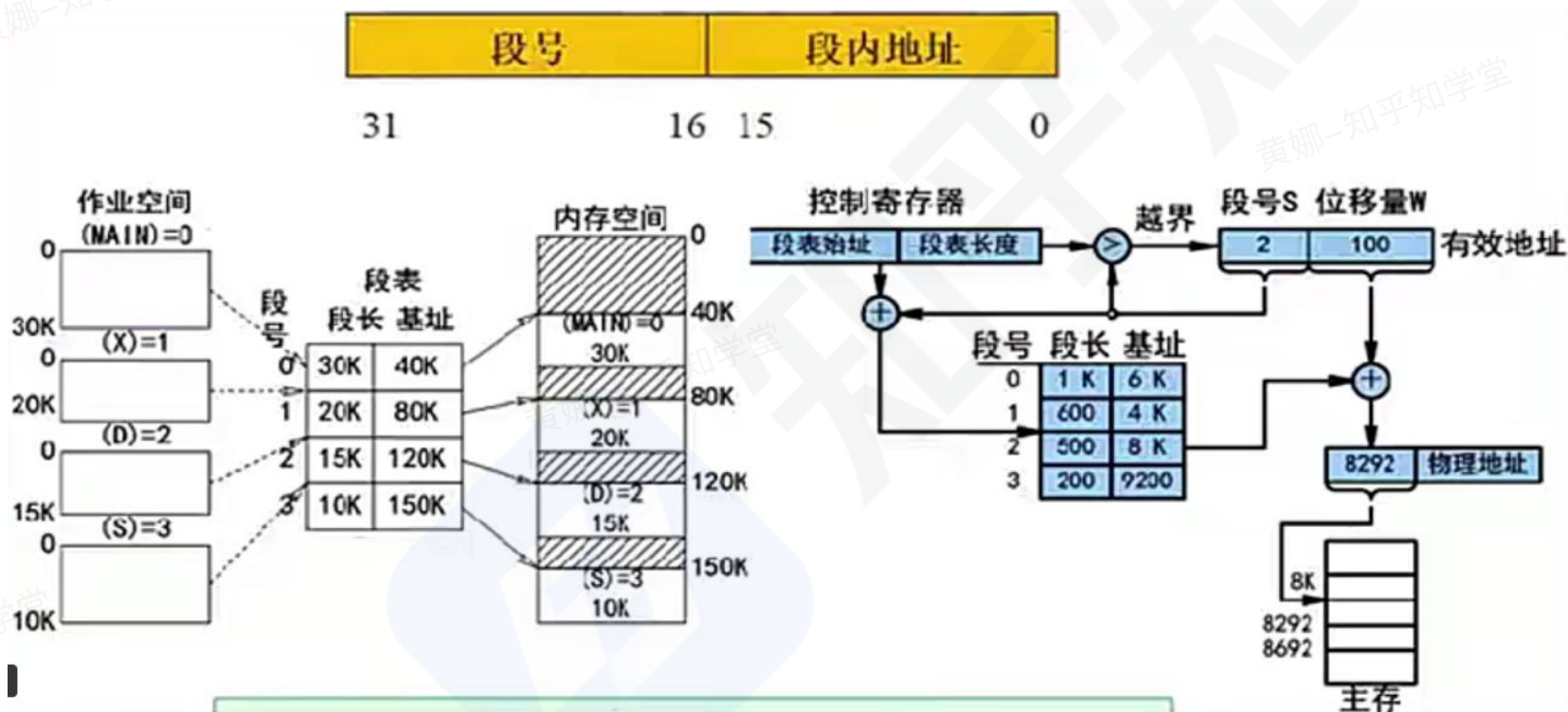




## 存储管理-段式存储

将进程空间分为一个个段，每段也有段号和段内地址，与页式存储不同的是，每段物理大小不同，分段是根据逻辑整体分段的。

地址表示: (段号, 段内偏移): 其中段内偏移不能超过该段号对应的段长, 否则越界错误, 而此地址对应的真正内存地址应该是: 段号对应的基地址+段内偏移。



优点: 程序逻辑完整, 修改互不影响

缺点: 内存碎片浪费大, 导致内存利用率低,



## 存储管理

设某进程的段表如下所示，逻辑地址( )可以转换为对应的物理位置。

A.(0, 1597)、(1, 30)和(3, 1390)

B.(0, 128)、(1, 30)和(3, 1390)

C.(0, 1597)、(2, 98)和(3, 1390)

D.(0, 128)、(2, 98)和(4, 1066)

| 段号 | 基地址  | 段长   |
|----|------|------|
| 0  | 1598 | 600  |
| 1  | 486  | 50   |
| 2  | 90   | 100  |
| 3  | 1327 | 2988 |
| 4  | 1952 | 960  |

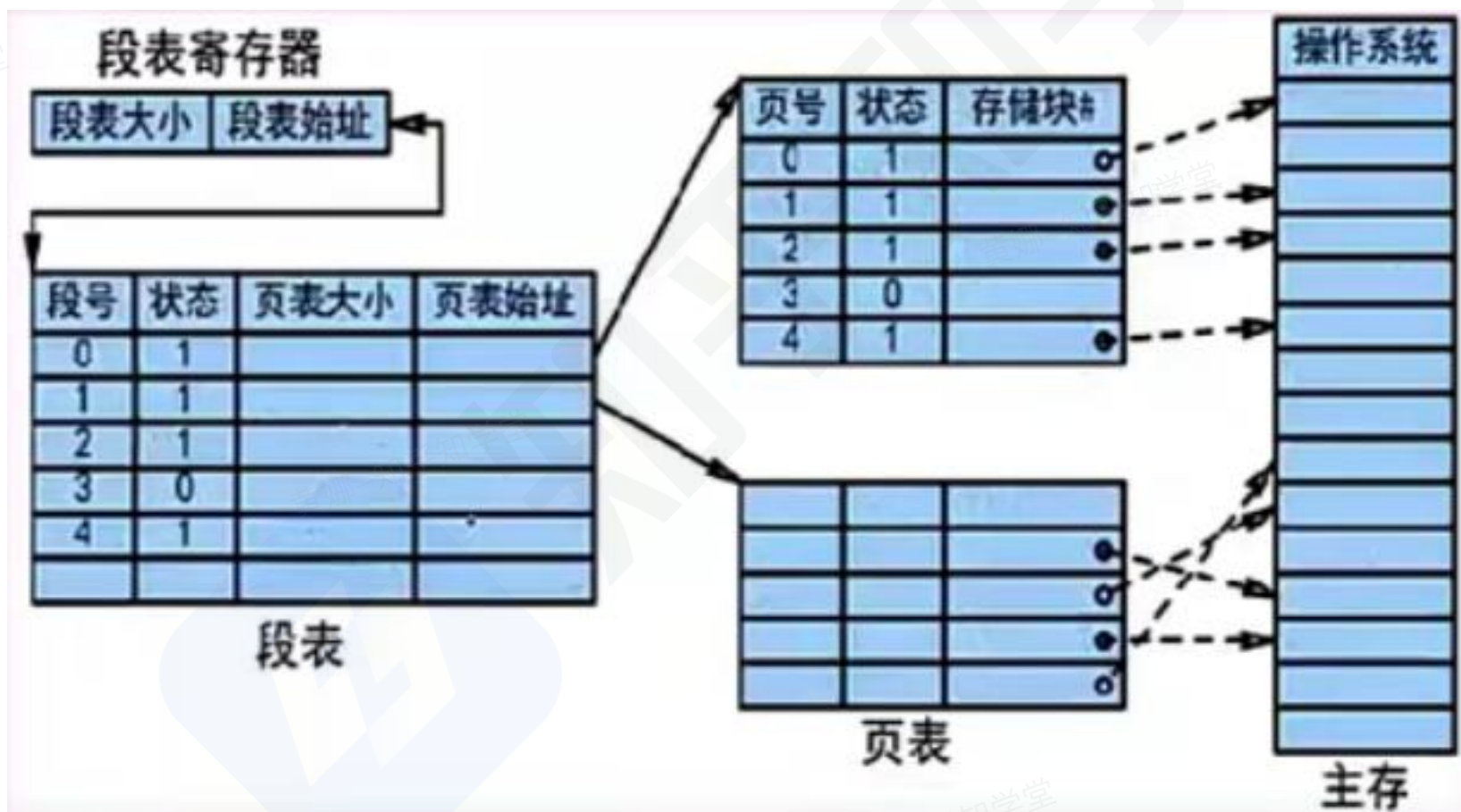


## 存储管理-段页存储

对进程空间先分段，后分页，具体原理图和优缺点如下：

优点：空间浪费小、存储共享容易、能动态连接。

缺点：由于管理软件的增加，复杂性和开销也增加，执行速度下降





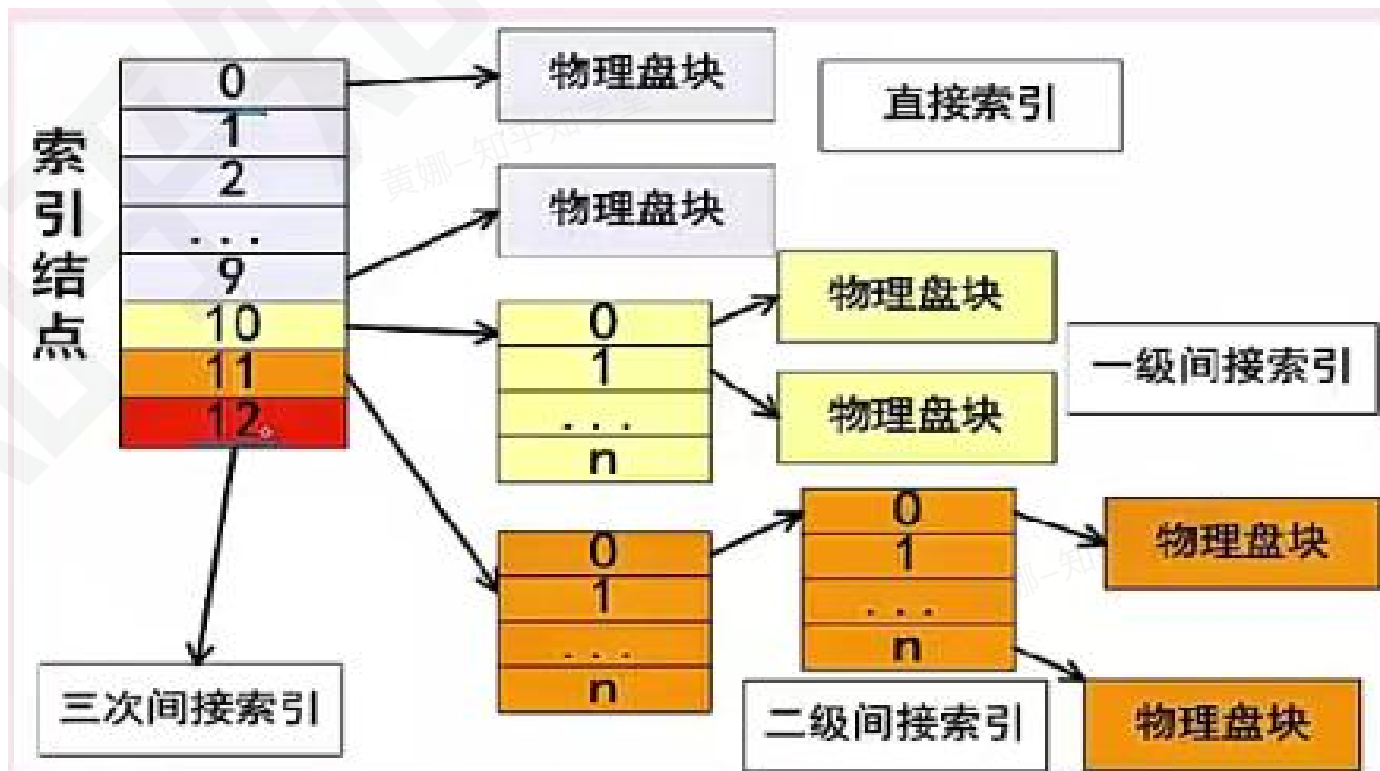
## 文件管理-文件结构

计算机系统中采用的索引文件结构如下图所示：

系统中有13个索引节点，0-9为直接索引，即每个索引节点存放的是内容，假设每个物理盘大小为4KB，共可存 $4KB \times 10 = 40KB$ 数据；

10号索引节点为一级间接索引节点，大小为4KB，存放的并非直接数据，而是链接到直接物理盘块的地址，假设每个地址占4B，则共有1024个地址，对应1024个物理盘，可存 $1024 \times 4KB = 4098KB$ 数据。

二级索引节点类似，直接盘存放一级地址，一级地址再存放物理盘快地址，而后链接到存放数据的物理盘块，容量又扩大了一个数量级，为 $1024 \times 1024 \times 4KB$ 数据。





## 文件管理

例：设文件索引节点中有8个地址项，每个地址项大小为4字节，其中5个地址项为直接地址索引，2个地址项是一级间接地址索引，1个地址项是二级间接地址索引，磁盘索引块和磁盘数据块大小均为1KB，若要访问文件的逻辑块号分别为5和518，则系统应分别采用\_27\_，而且可表示的单个文件最大长度是\_28\_KB。

- A. 直接地址索引和一级间接地址索引
- B. 直接地址索引和二级间接地址索引
- C. 一级间接地址索引和二级间接地址索引
- D. 一级间接地址索引和一级间接地址索引

A. 517

B. 1029

C. 16513

D. 66053



## 文件管理-树形文件

相对路径：是从当前路径开始的路径。

绝对路径：是从根目录开始的路径。

全文件名：绝对路径+文件名。

注意，绝对路径和相对路径是不加最后的文件名的，只是单纯的路径序列。

树形结构主要是区分相对路径和绝对路径，如下图所示：

文件属性

R只读文件属性

A存档属性

S系统文件

H隐藏文件

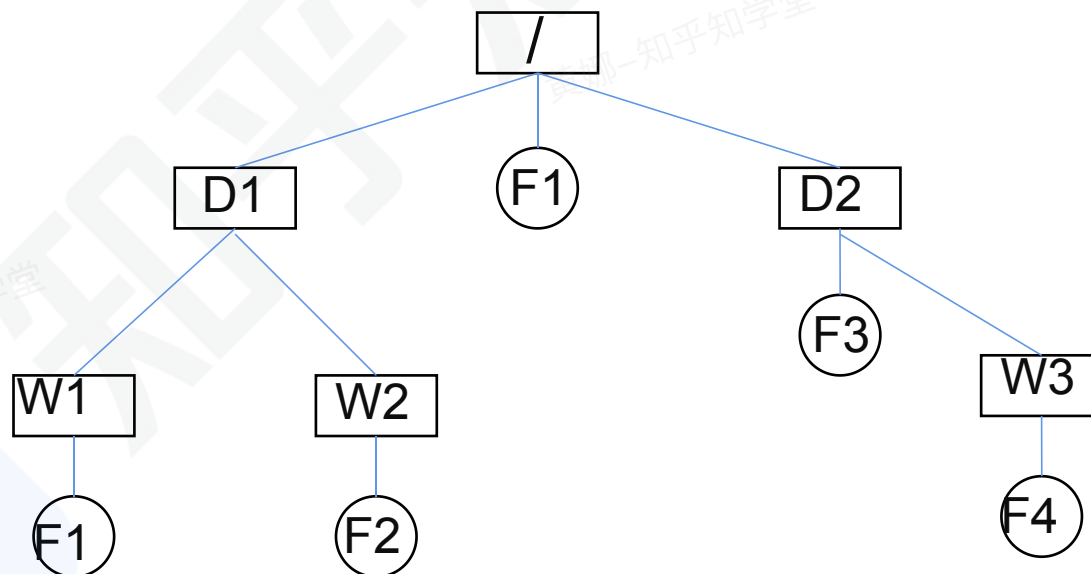
文件名的组成

驱动器号

路径

主文件名

扩展名





## 文件管理

例：若某文件系统的目录结构如下图所示，假设用户要访问文件Fault.swf，且当前工作目录为swshare，则该文件的全文件名为(27)，相对路径和绝对路径分别为(28)。

(27)

A.fault.swf

B.flash\fault.swf

C.swshare\flash\fault.swf

D.\swshare\flash\fault.swf

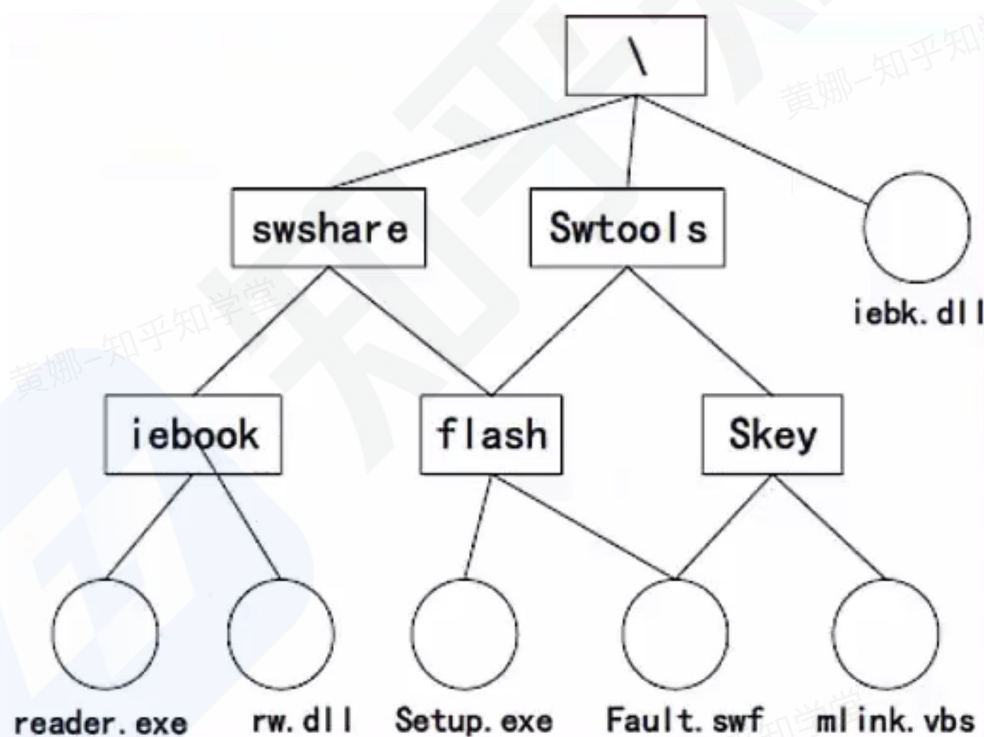
(28)

A.swshare\flash\和\flash\

B.flash和\swshare\flash\

C.\swshare\flash\和flash\

D.\flash\和\swshare\flash\





## 文件管理-空间存储

空闲区表法：将所有空闲空间整合成一张表，即空闲文件目录。

空闲链表法: 将所有空闲空间链接成一个链表, 根据需要分配。

成组链接法：既分组，每组内又链接成链表，是上述两种方法的综合。

位示图法：对每个物理空间用一位标识，为1则使用，为0则空闲，形成一张位示图。

[illegible]



## 文件管理

例：某文件管理系统在磁盘上建立了位示图(bitmap),记录磁盘的使用情况。若磁盘上的物理块依次编号为：0、1、2、...，系统中字长为32位，每一位对应文件存储器上的一个物理块，取值0和1分别表示空闲和占用，如下图所示。

| 31 | 30 | ... | 3 | 2 | 1 | 0 |   |
|----|----|-----|---|---|---|---|---|
| 0  | 1  | ... | 1 | 0 | 0 | 0 | 1 |

假设将4195号物理块分配给某文件，那么该物理块的使用情况在位示图中的第(1)个字中描述：系统应该将(2)

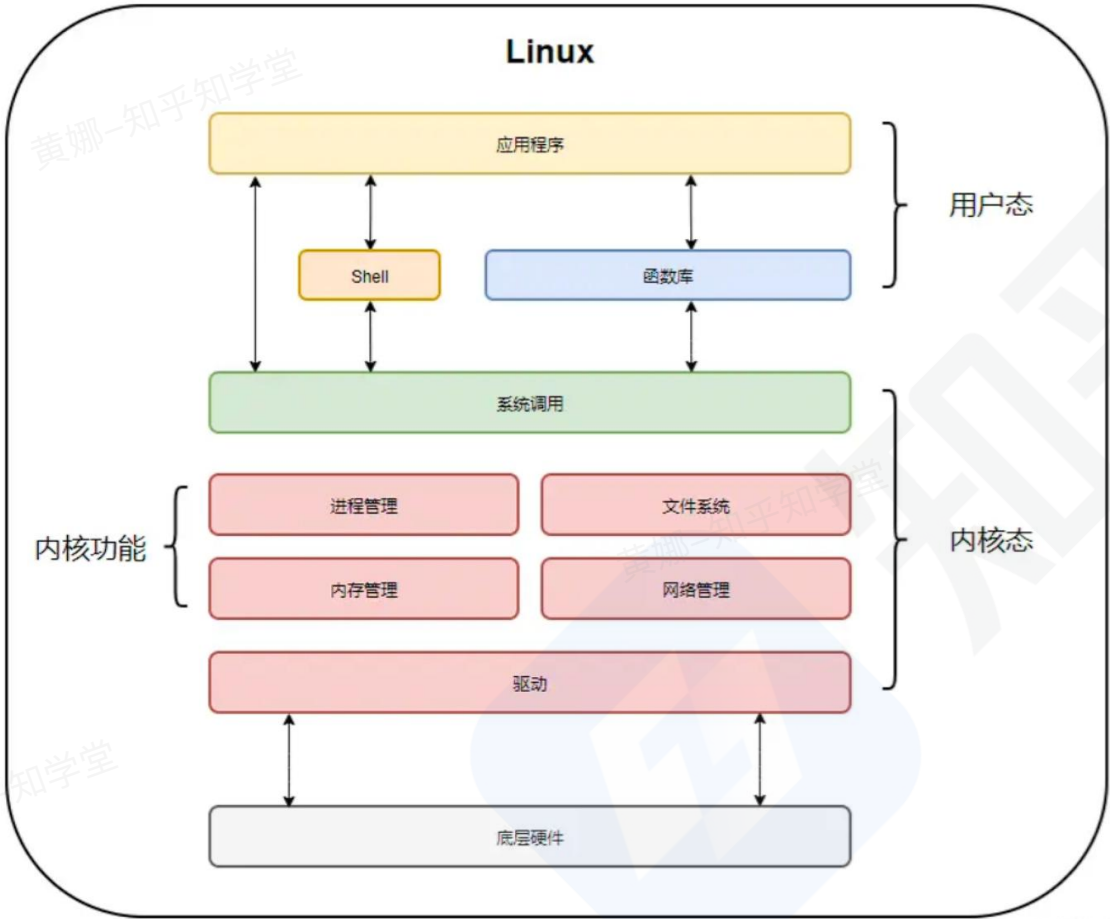
- A.129                      B.130                      C.131                      D.132

- A.该字的第3个位置为“0”  
B.该字的第3个位置为“1”  
C.该字的第4个位置为“0”  
D.该字的第4个位置为“1”



# 操作系统-微内核操作系统

微内核，顾名思义，就是尽可能的将内核做的很小，只将最为核心必要的东西放入内核中，其他能独立的东西都放入用户进程中，这样，系统就被分为了用户态和内核态



|      | 实质                                    | 优点                                                      | 缺点                             |
|------|---------------------------------------|---------------------------------------------------------|--------------------------------|
| 单体内核 | 将图形、设备驱动及文件系统等全部在内核中实现，运行在内核状态和同一地址空间 | 减少进程间通信和状态切换的系统开销，获得较高的运行效率                             | 内核庞大，占用资源较多且不易裁剪。系统的稳定性和安全性不好  |
| 微内核  | 只实现基本功能，将图形系统、文件系统、设备驱动及通信功能放在内核之外。   | 内核精炼，便于裁剪和移植。系统服务程序运行在用户地址空间，系统的可靠性、稳定性和安全性较高。可用于分布式系统。 | 用户状态和内核状态需要频繁切换，从而导致系统效率不如单体内核 |



## 操作系统-嵌入式操作系统

嵌入式操作系统特点：微型化、代码质量高、专业化、实时性强、可裁剪可配置

实时嵌入式操作系统的内核服务：异常和中断、计时器、I/O管理。

常见的嵌入式RTOS(实时操作系统)：VxWorks、RT-Linux、QNX、pSOS。

嵌入式系统初始化过程按照自底向上、从硬件到软件的次序依次为：  
芯片级初始化->板卡级初始化->系统级初始化。

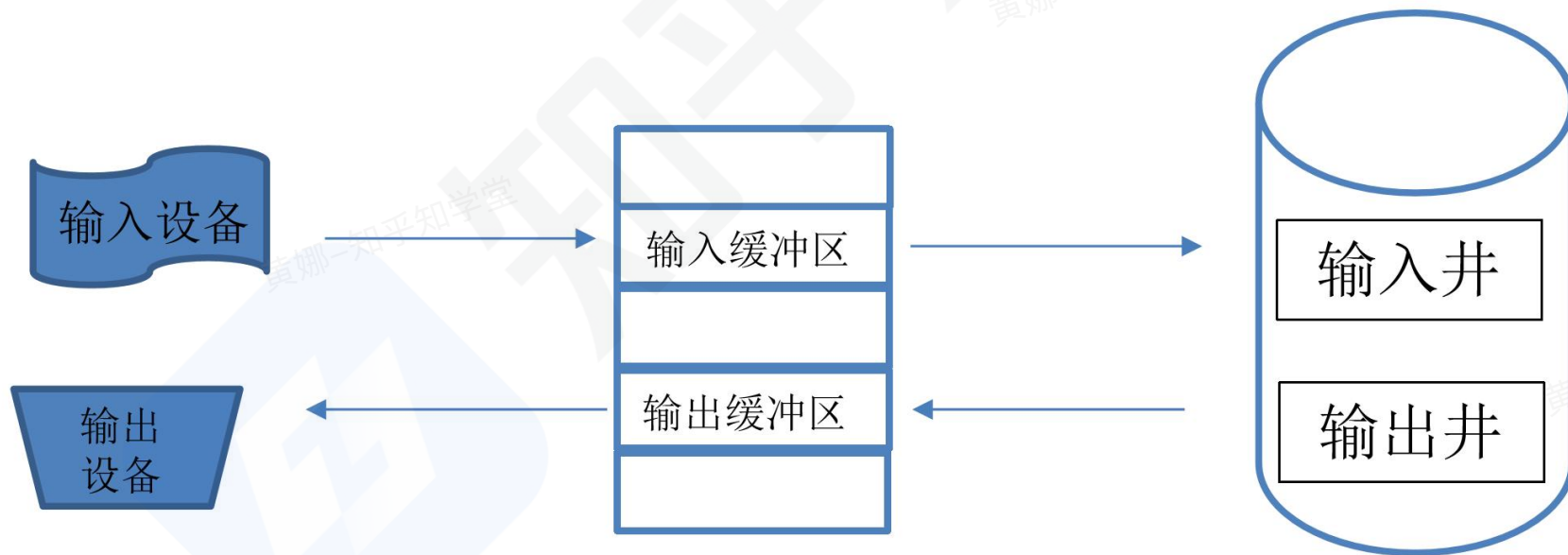
芯片级是微处理器的初始化，板卡级是其他硬件设备初始化，系统级初始化就是软件及操作系统初始化。



## 设备管理-虚设备和SPOOLING技术

一台实际的物理设备，例如打印机，在同一时间只能由一个进程使用，其他进程只能等待，且不知道什么时候打印机空闲，此时，极大的浪费了外设的工作效率。

引入SPOOLING技术，就是在外设上建立两个数据缓冲区，分别称为输入井和输出井，这样，无论多少进程，都可以共用这一台打印机，只需要将打印命令发出，数据就会排队存储在缓冲区中，打印机会自动按顺序打印，实现了物理外设的共享，使得每个进程都感觉在使用一个打印机，这就是物理设备的虚拟化。如下图所示：



T H E   E N D

功不唐捐，玉汝于成！

开   启   新   征   程

