

操作系统题库汇总

一、选择题

1. 下面的 (B) 不是文件的存储结构

- A. 索引文件 B. 记录式文件
- C. 串联文件 D. 连续文件

- 顺序文件：包括连续文件和串联文件
- 顺序文件：记录按在其文件的中的逻辑顺序依次存入储存介质而建立的，即顺序文件中的物理记录和逻辑记录的顺序是一致的
- 连续文件：连续文件中次序相继的两个记录在存储介质中是相邻的（数组概念）
- 串联文件：串联文件中的物理机理次序通过指针的链接表示（链表）
- 索引文件：类似索引概念

2. 有一磁盘，共有 10 个柱面，每个柱面 20 个磁道，每个盘面分成 16 个扇区。采用位示图对其存储空间进行管理。如果字长是 16 个二进制位，那么位示图共需 (A) 字

- A. 200 B. 128 C. 256 D. 100

盘物理块： $20 \times 16 \times 100 = 3200$

采用 16 字长： $3200 / 16 = 200$ 字

3. 操作系统为每一个文件开辟一个存储区，在它的里面记录着该文件的有关信息。这就是所谓的 B

- A. 进程控制块 B. 文件控制块
- C. 设备控制块 D. 作业控制块

- PCB 进程控制块：描述进程外部特征，感知控制进程动态变化的数据结构
- FCB 文件控制块：存储文件的相关信息的数据结构
- DCB 设备控制块：记录硬件设备的特性，连接，使用情况等信息的数据结构
- JCB 作业控制块：描述作业状态等相关信息的数据结构

4. 文件控制块的英文缩写符号是 C

- A. File B. DCB C. FCB D. JCB

5. 一个文件的绝对路径名总是以 C 打头

- A. 磁盘名 B. 字符串 C. 分隔符 D. 文件名

6. 一个文件的绝对路径名是从 B 开始，逐步沿着每一级子目录向下，最后到达指定文件的整个通路上所有子目录名组成的一个字符串

- A . 当前目录 B . 根目录
- C . 多级目录 D . 二级目录

例如：要显示 YouKu 目录下的 youkuclient 目录中的 YoukuDesktop 命令，其绝对路径为 D:\ProgramFiles(x86)\YouKu\youkuclient\YoukuDesktop.exe,如果当前目录为 YouKu 的话，则其相对路径为 youkuclient\YoukuDesktop.exe

7 . 从用户的角度看，引入文件系统的主要目的是 D

- A . 实现虚拟存储 B . 保存用户和系统文档
- C . 保存系统文档 D . 实现对文件的按名存取

用户角度，方便用户

8 . 按文件的逻辑结构划分，文件主要有两类： A

- A . 流式文件和记录式文件 B . 索引文件和随机文件
- C . 永久文件和临时文件 D . 只读文件和读写文件

文件的逻辑结构分为两种：无结构的字符流式文件，有结构的记录式文件

- 无结构的字符流式文件：文件由字符序列组成，文件内信息不再划分结构
- 有结构的记录式文件：文件由记录组成组成，即文件内信息划分为多个记录，以记录为单位组织和使用信息

9 . 位示图用于 B

- A . 文件目录的查找 B . 磁盘空间的管理
- C . 主存空间的共享 D . 文件的保护和保密

位示图：利用二进制的一位来表示磁盘中盘块的使用情况 0 空闲，1 占用

10 . 用户可以通过调用 C 文件操作，来归还文件的使用权

- A . 建立 B . 打开 C . 关闭 D . 删除

11.与计算机硬件关系最密切的软件是（D）

- A.编译程序 B.数据库管理程序
- C.游戏程序 D.OS

12.现代 OS 具有并发性和共享性，是由（D）的引入而导致的

- A.单道程序 B.磁盘 C.对象 D.多道程序

多道程序:指在内存中同时存放几道互相独立的程序，使他们在程序管理控制下，相互穿插运行

13. (A) 不是多道程序系统

- A.单用户单任务 B.多道批处理系统
- C.单用户多任务 D.多用户分时系统

14. (B) 是多道操作系统不可缺少的硬件支持

- A.打印机 B.中断机构 C.软盘 D.鼠标

中断必不可少

中断：指出现需要时，cpu 暂时停止当前程序的执行转而执行新情况的程序和执行过程

15.系统的主要功能有 (c)

- A.进程管理、存储器管理、设备管理、处理机管理
- B.虚拟存储管理、处理机管理、进程调度、文件系统
- C.处理机管理、存储器管理、设备管理、文件系统
- D.进程管理、中断管理、设备管理、文件系统

操作系统 5 大功能：

处理机管理，储存器管理，设备管理，文件管理，以及作为用户与硬件系统之间的接口

16.单处理机计算机系统中，(A) 是并行操作的

- A.处理机操作和通道操作是并行的
- B.程序与程序
- C.主程序与子程序
- D.用户程序与操作系统程序

- 单处理机系统：只有 1 个运算器，运行状态最多一个
- 单处理机操作：理解为 cpu
- 通道：有自己的处理器，能自己执行指令，无需 cpu

17.处理机的所有指令可以在 (D) 中执行

- A.目态 B.浏览器中 C.任意时间 D.系统态

Cpu 工作状态分为：系统态和用户态（目态）

引入两个状态的原因：

为了避免用户程序错误的使用特权指令，保护 os 不被用户程序破坏，规定：cpu 处于用户态时，不允许执行特权指令，当 cpu 处于系统态时，可以执行包括特权指令在内的一切机器指令

18. (B) 功能不是操作系统直接完成的功能

- A.管理计算机硬盘 B.对程序进行编译
- C.实现虚拟存储器 D.删除文件

需要编译器，比如 GCC

19.要求在规定的时间内对外界的请求必须给予及时相应的 OS 是 (B)

- A.多用户分时系统 B.实时系统
- C.批处理系统时间 D.网络操作系统

- 多用户分时操作系统：一台计算机连接很多终端机，计算机按固定的时间片为每个终端机服务，由于计算机处理深速度快，用户感觉不到等待时间，似乎这台计算机专门为自己服务一样
- 批处理操作系统：采用批处理技术的 os，批处理：指用户将一批作业提交给 os 后就不在干预，有 os 控制他们自动运行，优点是用户不用一个一个提交作业，效率更高，没有人机交互

二者区别：批处理 os 没有人机交互，而分时 os 不仅有人机交互还允许多个用户同时使用，批处理 os 允许程序长时间占用 cpu，而分时 os 不允许

- 网络操作系统：向网络计算机提供服务的特殊的 os，网络的心脏和灵魂
- 实时操作系统：指当外界产生数据时，能足够快的处理数据，足够快的响应（规定时间内）分类：硬实时：规定时间必须完成，软实时：只要求按照任务优先级尽可能完成

20.分布式系统和网络系统的主要区别是 (C)。

- A.并行性 B.透明性 C.共享性 D.复杂性

分布式系统：分布式计算，利用更多的计算机，处理更多的数据

- 二者共同点：分布式 os 是建立在网络 os 上的，所以二者在物理结构上是基本相同的
- 二者区别：分布式 os 的设计思想和网络 os 是不同的，这样就决定了二者在结构和功能上的不同，网络 os 要求网络用户在使用网络资源的时候必须了解网络资源，用户必须要知道网络中各个计算机的功能配置，资源情况，比如网络 os 中用户如果要读一个共享文件，用户必须知道该文件在那个计算机的那个目录下面，而分布式 os 以全局是管理系统资源，可以为用户任意调度网络资源，且调度过程是透明的（用户看不见的），当用户提交一个作业时，分布式 os 能在系统中选择最合适的处理器，将用户的作业提交道处理程序，然后将结果传回给用户，在这个过程中，用户并不会感觉到有多个处理器的存在，这个系统就像是一个处理器

所以二者的共享的不同的共享，答案是共享性，

Ps：计算机中透明的概念：某过程对用户透明，即用户看不见，即黑盒

21. (C) 操作系统允许在一台主机上同时连接多台终端，多个用户可以通过各自的终端同时交互的使用计算机。

- A.网络 B.分布式 C.分时 D.实时

22.如果分时操作系统的时间片一定，那么（B），则相应时间越长

- A.用户数越少 B.用户数越多 C.内存越小 D.内存越大

时间片大小固定，用户越多，每个用户分配到的时间片就越少，需要等待响应的时间就越长

23.下面六个系统中，必须是实时操作系统的有（C）个

- (1)航空订票系统
- (2)过程控制系统
- (3)机器口语翻译系统
- (4)计算机辅助系统
- (5)办公自动化系统
- (6)计算机激光照排系统

- A.1 B.2 C.3 D.4

1, 2, 3

24.在下面关于并发性的叙述众正确的是（C）

- A.并发性是指若干事件在同一时刻发生
- B.并发性是指若干事件在不同时刻发生
- C.并发性是指若干事件在同一时间间隔发生
- D.并发性是指若干事件在不同时间间隔发生

- 并发：若干事件在同一时间间隔发生，（同一间隔）
- 并行：若干事件在同一时刻发生（同一时刻，注意理解）

具体：

- 并发是指在同一个人cpu上同时运行多个程序（不是真正的同时，而是看来的同时，因为cpu要在多个程序间切换，实质是物理cpu在若干道程序间的多路复用，是对有限的物理资源强制给多用户共享以提高效率）
- 并行：指若干事件可以在同一时刻发生

二者的异同：

并行是指两个同时发生的并发事件，具有并发的含义，而并发则不一定并行，也就是说并发事件不一定要同一时刻发生

25.一般来说，为了实现多道程序设计，计算机最需要（A）。

- A.更大的内存 B.更多的外设
- C.更快的CPU D.更先进的终端

多道程序设计：在内存中同时存放几道互相独立的程序，使得他们在管理控制程序下相互穿插运行。

多道程序设计按照一定的作业调度算法将作业队列中的作业调入到内存中，使他们共享cpu和各种资源，所以内存更大才能实现多道

26.当(B)时， 进程从执行状态转变为就绪状态。

- A.进程被调度程序选中
- B.时间片到
- C.等待某一事件
- D.等待的事件发生

进程被调度程序选中： 就绪->运行

等待某一事件： 执行/创建->等待（阻塞）

等待事件发生： 等待->运行

时间片到： 运行->就绪

27.在进程状态转换时， 下列(D)转换是不可能发生的。

- A.就绪态→运行态
- B.运行态→就绪态
- C.运行态→阻塞态
- D.阻塞态→运行态

阻塞只能到就绪， 然后由就绪到运行

阻塞： 等待除处理机资源以外的其他资源或事件

就绪： 只等待处理机资源， 其余资源事件都具备

28.下列各项工作步骤中， (B)不是创建进程所必需的步骤。

- A.建立一个 PCB
- B.作业调度程序为进程分配 CPU
- C.为进程分配内存等资源
- D.将 PCB 链入进程就绪队列

进程创建的四个步骤：

- 1.申请空白 pcb
- 2.为进程分配资源
- 3.初始化 pcb
- 4.将进程插入就绪队列

29.下列关于进程的叙述中， 正确的是(A)。

- A.进程通过进程调度程序而获得 CPU
- B.优先级是进行进程调度的重要依据， 一旦确定不能改变
- C.在单 CPU 系统中， 任一时刻都有 1 个进程处于运行状态
- D.进程申请 CPU 得不到满足时， 其状态变为等待状态

B 优先级可以改变

C 所有进程死锁的时候， 所有进程处于等待状态， cpu 空闲的时候在运行空闲进程， 不是没有进程运行

D 不是等待态， 而是就绪态， 等待态： 等待除 cpu 以外的其他资源或进程

30.从资源管理的角度看， 进程调度属于(C)。

- A.I/O 管理 B.文件管理
- C.处理机管理 D.存储器管理

处理机管理的主要功能：

进程控制， 进程同步， 进程通信， 进程调度

31.下列有可能导致一进程从运行变为就绪的事件是(D)。

- A.一次 I/O 操作结束
- B.运行进程需作 I/O 操作
- C.运行进程结束
- D.出现了比现运行进程优先权更高的进程

A： 等待->就绪

B:运行->等待

C： 进程结束

32.一个进程释放一种资源将有可能导致一个或几个进程(D)

- A.由就绪变运行 B.由运行变就绪
- C.由阻塞变运行 D.由阻塞变就绪

没有说明该资源是 cpu 资源， 如果是 cpu 资源的话， 选择 A 是对的

如果没有说明的话， 默认是 cpu 资源或者其他资源， 阻塞的进程在等待某种资源， 等待的资源到了以后， 由阻塞变为就绪

33.一次 I/O 操作的结束， 有可能导致(A)

- A.一个进程由睡眠变就绪 B.几个进程由睡眠变就绪
- C.一个进程由睡眠变运行 D.几个进程由睡眠变运行

考察设备的共享属性， 设备的共享属性分为两类：

- 独占设备进程间互斥的访问这类设备， 设备一旦被分配给某个进程， 便由该进程独占， 比如打印机， OI 操作后自然只有该进程由等待进入就绪
- 共享设备， 一段时间内运行多给进程同时访问设备（并发的概念）， 如磁带， 多个进程的读写可以交叉进行， 但是在一个极短的时间内 OI 操作只对一个进程进行服务， 对 OI 设备的访问是并发的， 不是并行的， 一次 OI 结束， 只是其对应进程的 OI 操作结束， 只会唤醒这一个进程

34.在下面的叙述中， 不正确的是(D)

- A.一个进程可创建一个或多个线程
- B.一个线程可创建一个或多个线程
- C.一个线程可创建一个或多个线程
- D.一个进程可创建一个或多个进程

线程是进程内部的一个执行单元

35.若系统中只有用户级线程，则处理机调度单位是(B)

- A.线程
- B.进程
- C.程序
- D.作业

只有用户级线程，那进程就是最小单位，因为用户级的无法访问内核资源，如果系统只有用户态线程，则线程对操作系统是不可见的，操作系统只能调度进程，如果系统中由内核态线程，则操作系统可以按照线程进行调度

36.一个进程是(C)

- A.由协处理机执行的一个程序
- B.一个独立的程序+数据集
- C.PCB 结构与程序和数据的组合
- D.一个独立的程序

进程的概念，pcb+程序+数据=进程

37.在下面的叙述中正确的是(B)

- A 线程是比进程更小的能独立运行的基本单位.
- B.引入线程可提高程序并发执行的程度，可进一步提高系统效率
- C.线程的引入增加了程序执行时时空开销
- D.一个进程一定包含多个线程

A 线程是操作系统进行调度的基本单位，但是不是独立运行的，必须依存在应用程序中，由应用程序提供多个线程执行控制

C 线程的引入简化了线程间的通信，节省了系统的时空开销，以小的开销来提供线程的并发程度

D 线程是进程内部的运行单元，一个进程不一定包含多个线程，而是至少包含一个线程

38.下面关于线程的叙述中，正确的是(C)

- A.不论是系统支持线程还是用户级线程，其切换都需要内核的支持
- B.线程是资源的分配单位，进程是调度和分配的单位
- C.不管系统中是否有线程，进程都是拥有资源的独立单位
- D.在引入线程的系统中，进程仍是资源分配和调度分派的基本单位

A 线程分为两类，用户级和内核级（系统级）

在用户级线程中，线程的管理由应用程序完成，内核意识不到线程的存在，内核级（系统级）线程中，线程的管理由内核完成，应用程序没有进行线程管理的代码，只有一个到内核级线程的编程接口

BD 传统的操作系统中，拥有资源和独立调度的基本单位都是进程

引入线程的操作系统中，线程是独立调度的基本单位，进程是资源分配拥有的基本单位

39.进程的控制信息和描述信息存放在(B)

A.JCB B.PCB C.AFT D.SFT

40.从执行状态挂起的进程解除挂起时进入(A)状态

A.就绪 B.执行 C.阻塞 D.挂起

cpu 资源暂停给该进程，该进程由执行变为就绪

就绪：只缺少 cpu 资源，其他什么都不缺

41.一个进程执行过程中不可能对应(C)

A.一个 PBC B.一个 JCB C.多个 PCB D.一个程序

Pcb：进程控制块

进程的 pcb 具有唯一性

作业：指在一次应用业务处理过程中，从输入开始到结束，用户要求计算机所做的由关该次作业处理的全部工作

Jcb：作业控制块：作业在系统中存在的标准，其内容是作业调度的依据

一个进程不仅可以对应一个程序，还可以对应多个程序，比如该进程的执行需要别的进程为其提供服务，这样就对应了多个程序

一个程序也可以对应多个进程，比如一个程序执行多次，就对应多个进程

42.用 P、V 操作管理临界区时，信号量的初值一般应定义为（C）

A. - 1 B.0 C.1 D.任意值

- 临界区：每个进程中访问临界资源的那段代码叫做临界区，每次只允许一个进程进入临界区，进入后，不允许其他进程进入
- 临界资源：是一次仅允许一个进程使用的共享资源，各进程采取互斥方式，属于临界资源的硬件：打印机等，属于临界资源的软件：消息队列，变量，数组，缓冲区等

管理临界区时，对互斥资源的访问设置为 1，对同步资源的访问设置为 0

43.在下面的叙述中正确的是（C）

- A.临界资源是非共享资源 B.临界资源是任意共享资源
- C.临界资源是互斥共享资源 D.临界资源是同时共享资源

44.对进程间互斥的使用临界资源，进程可以 (D)

- A.互斥的进入临界区 B.互斥的进入各自的临界区
- C.互斥的进入同一临界区 D.互斥的进入各自的同类资源临界区

- 临界区：每个进程中访问临界资源的那段代码
- 临界资源：一次只允许一个进程访问的共享资源，比如打印机

题目的意思是进程间互斥，针对同类临界资源的，而对应的代码应该是独立的，所以各个进程只能访问各自的代码空间，所以互斥的进入各自同类临界区，从另一个方向解释，临界区是使用临界资源的代码，同类资源临界区则是使用同一临界资源的代码，临界资源由多种，使他们不一定互斥，而使用同一临界资源自然是互斥的

A 不完整

B 同样不完整

C 应该是进入各自的临界区

45.设两个进程共用一个临界资源的互斥信号量 mutex，当 mutex = 1 时表示 (B)

- A.一个进程进入了临界区，另一个进程等待
- B.没有一个进程进入临界区
- C.两个进程都进入临界区
- D.两个进程都在等待

信号量表示资源数目，本题两个进程使用同一临界资源，所以初值为 1

信号量取值范围：-1，0，1

信号量为 1：代表资源可用，两个进程都为使用临界资源

信号量为 0：表示一个进程正在使用临界资源，另外一个需要等待

信号量为 -1：表示一个进程正在使用临界资源，另外一个进程因等待而阻塞在信号队列中，需要当前在临界资源中的进程退出时将其唤醒

46.设两个进程共用一个临界资源的互斥信号量 mutex，当 mutex = - 1 时表示 (A)

- A.一个进程进入了临界区，另一个进程等待
- B.没有一个进程进入临界区
- C.两个进程都进入临界区
- D.两个进程都在等待

47.当一进程因在记录型信号量 S 上执行 P (S) 操作而被阻塞后，S 的值为 (B)。

- A. > 0 B. < 0 C. ≥ 0 D. ≤ 0

- 记录型信号量：包含一个代表资源数目的整型变量和一个进程链表（等待队列）
- 整型信号量：只包含一个代表资源数目的整型变量

对于记录型信号量，采取了让权等待策略，当 $s < 0$ ，即不存在可用资源的时候，因为其存在进程链表等待队列，所以不会盲等，而是会阻塞

对于整型信号量，没有采取让权等待策略，当 $s \leq 0$ 的时候，即也是不存在可用资源的时候，因为其没有进程等待队列，所以不会阻塞，而是会陷入盲等状态

p 操作代表通过，v 操作代表释放

48.当一进程因在记录型信号量 S 上执行 V (S) 操作而导致唤醒另一进程后，S 的值为 (D)

- A. > 0 B. < 0 C. ≥ 0 D. ≤ 0

s 的取值可正可负，正数表示当前空闲资源数量，0 表示当前没有空闲资源，负数的绝对值表示正在等待进入临界区的进程个数，因为执行的是 v（唤醒进程操作），所以开始 s 应该是负数，加一之后肯定是 ≤ 0

49.若有 4 个进程共享同一程序段，而且每次最多允许 3 个进程进入该程序段，则信号量的变化范围是 (B)

- A. 3, 2, 1, 0
- B. 3, 2, 1, 0, -1
- C. 4, 3, 2, 1, 0
- D. 2, 1, 0, -1, -2

s 的取值可正可负，正数表示当前空闲资源数量，0 表示当前没有空闲资源，负数的绝对值表示正在等待进入临界区的进程个数

3,2,1 表示当前空闲的资源有 3, 2, 1 个

0 表示当前没有空闲资源

-1 表示有一个进程处于等待队列

50.若信号 S 的初值为 2，当前值为 - 1，则表示有 (B) 个等待进程

- A. 0 B. 1 C. 2 D. 3

2 代表有两个资源空闲

负数的绝对值表示在等待的进程数量

51. (A) 操作不是 P 操作可完成的

- A.为进程分配处理机 B.使信号量的值变小
- C.可用于进程的同步 D.使进程进入阻塞状态

p 原语操作的动作：

- s 减一
- 若 s 减一后仍然大于或等于 0，则进程继续执行
- 若 s 减一后小于 0，则该进程被阻塞后进入与该信号对于的队列中，然后转进程调度

v 原语操作的动作：

- s 加一
- 若相加结果大于 0，则进程继续执行
- 若相加结果小于或者等于 0，则从该信号的等待队列中唤醒一等待进程，然后返回原进程继续执行或者转进程调度

52.某系统采用了银行家算法，则下列叙述正确的是（B）

- A.系统处于不安全状态时一定会发生死锁
- B.系统处于不安全状态时可能会发生死锁
- C.系统处于安全状态时可能会发生死锁
- D.系统处于安全状态时一定会发生死锁

银行家算法：一种避免产生死锁的算法，在系统进行资源分配前，先计算此次分配资源的安全性，若分配会导致系统进入不安全状态，则不分配，否则分配，系统处于不安全状态时，可能发生死锁，系统处于安全状态时，一定不会发生死锁

53.银行家算法中的数据结构包括有可利用资源向量 Available、最大需求矩阵 Max、分配矩阵 Allocation、需求矩阵 Need，下列选项中表述正确的是（B）。

- A. $Max[i,j]=Allocation[i,j]+Need[i,j]$
- B. $Need[i,j]=Max[i,j]-Allocation[i,j]$
- C. $Max[i,j]=Allocation[i,j]-Need[i,j]$
- D. $Need[i,j]=Max[i,j]+Allocation[i,j]$

Available:可利用资源向量，每一类元素代表可利用资源的数目，比如 $Available[j]=k$,表示系统中 R[j]类资源现在有 k 个

Max:最大需求矩阵，定义了系统中 n 个进程中每一个进程对 m 类资源的最大需求，比如 $max[i,j]=k$,表示进程 i 需要 R[j]类资源 k 个

Allocation:分配矩阵，定义了系统中每一类资源分配给每一进程的资源数，比如 $Allocation[i,j]=k$,表示进程 i 当前已经得到的 R[j]类资源数目为 K

Need：需求矩阵，表示每一个进程尚需的各类资源数，比如 $Need[i,j]=k$,表示进程 i 还需要 R[j]类资源 k 个才可以完成任务

关系： $Need[i,j]=Max[i,j]-Allocation[i,j]$

54.在采用动态优先权的优先权调度算法中，如果所有进程都具有相同优先权初值，则此时的优先权调度算法实际上和（A）相同

- A.先来先服务调度算法
- B.短作业优先调度算法
- C.时间片轮转调度算法
- D.长作业优先调度算法

55.分页存储管理的存储保护是通过（A）完成的

- A.页表（页表寄存器） B.快表
- C.存储键 D.索引动态重定

页式管理为内存提供两种保护方式：

- 地址越界保护：由地址变换机构中控制寄存器的值-页表长度和所要访问的虚地址比较来完成
- 通过页表控制对内存信息的存取操作方式：在页表中增加相应的保护位

56.把作业地址空间中使用的逻辑地址变成内存中物理地址称为（B）。

A.加载 B.重定位 C.物理化 D.逻辑化

重定位又叫重定向，实现了从逻辑地址到物理地址的转换

分为两种：

- 静态重定位：在程序运行之前，由操作系统完成地址重定位工作
- 动态重定位：在程序运行中，完成重定位工作（需要硬件支持）

57.在存储管理中，采用覆盖与交换技术的目的是（A）

A.减少程序占用的主存空间

B.物理上扩充主存容量

C.提高 CPU 效率

D.代码在主存中共享

- 覆盖技术：把程序划分为若干给功能上相对独立的程序段，按照其自身逻辑结构使那些不会同时运行的程序段共享同一块内存区，程序段先保存在磁盘，当程序的前一部分结束后，把后续段调入内存，覆盖前面的程序段
- 交换技术：在分时系统中，用户的进程比内存能容纳的数量更多，系统将哪些不再允许的进程或者某一部分调出内存，暂时存放在外存上的一个后备存储区，通常称为交换区，当需要允许这些进程的时候，再将他们装入内存

理解了覆盖和交换的意思，就知道选 A，一切都是为了内存考虑

58.存储管理方法中，（A）用户可采用覆盖技术

A.单一连续区 B.可变分区存储管理

C.段式存储管理 D.段页式存储管理

覆盖技术是早期在单一连续储存管理中使用的扩大储存器容量的一种技术

在单道连续分配中，当储存空间容不下程序时，可用采用覆盖方法解决

覆盖基本思想：由于程序运行时并非各个部分都要访问，因此可用将用户空间分成一个固定区和多个覆盖区，将经常活跃的放入固定区，将那些将要用的段放入覆盖区，在需要时提前调入覆盖区，替换原有的段

59.在内存分配的”最佳适应法”中，空闲块是按（C）

A.始地址从小到大排序 B.始地址从大到小排序

C.块的大小从小到大排序 D.块的大小从大到小排序

最佳适应法：从全部空闲区中找出满足作业要求的且大小最小的空闲分区

60.用空白链记录内存空白块的主要缺点是（B）

- A.链指针占用了大量的空间 B.分配空间时可能需要一定的查链时间
- C.不好实现"首次适应法" D.不好实现"最佳适应法"

- 首次适应法：按照地址排序，然后依次找到满足需求的内存块即可
- 最佳适应法：空闲块从小到大排序，找到大小最接近的内存块

对于 C，D 选项，空间地址是递增的，每个空白块的大小不确定，所以两个算法只需要找到所需大小的空白块即可

对于 A，指针的大小是 4 个字节，和指针指向的空白块比，大小基本可忽略，就算由很多指针，但也说明了有很多空白块，所以对比一下，指针占用的空间大小基本可用忽略

对于 B，这个是由链表的特性决定了，查找要一个接着一个的查找

61.分区管理和分页管理的主要区别是（D）

- A.分区中的块比分页中的页要小
- B.分页有地址映射而分区没有
- C.分页有存储保护而分区没有
- D.分区要求一道程序存放在连续的空间内而分页没有这种要求

分区管理：

1. 固定分区（相当于固定数组）

优点：容易实训，开销小

缺点：存在内部碎片（分区内没有被利用的空间），分区总数固定，限制了并发执行的程序数量

2. 动态创建分区(相当于 c++stl 中的 vector，动态申请内存空间)

优点：没有内部碎片

缺点：有外部碎片（难以利用的小的空闲分区）

页式管理：

优点：没有外部碎片，最后一页可能有内部碎片但不大，程序不必连续存放，便于改变程序占用空间大小

缺点：程序仍需全部装入内存

核心：内存的离散和连续

62.静态重定位的时机是（C）。

- A.程序编译时 B.程序链接时
- C.程序装入时 D.程序运行时

静态重定位：是在目标程序装入内存时，由装入程序对目标程序中的指令和数据的地址进行修改，即完成逻辑地址到物理地址的转换，对每

个程序来说，这种地址变换只在装入时一次完成，在程序运行期间不再进行重定位

63.通常所说的"存储保护"的基本含义是（C）

- A.防止存储器硬件受损
- B.防止程序在内存丢失
- C.防止程序间相互越界访问
- D.防止程序被人偷看

常识，越界访问比如游戏外挂，修改游戏参数

64.能够装入内存任何位置的代码程序必须是（C）

- A.可重入的 B.可定位的
- C.可动态链接的 D.可静态链接的

将用户源程序变为内存中可执行的程序，需要三个步骤

- 编译：由编译程序将用户源代码编译成若干个目标模块
- 链接：将编译形成后的一组目标模块以及所需的库函数链接在一起，形成一个完整的装入模块
- 装入：由装入程序将装入模块装入内存中运行

链接的三种方式：

- 静态链接：在程序运行之前，先将各目标模块及他们所需的库函数链接成一个完整的可执行的程序，以后不再拆开
- 装入时动态链接：将编译得到的目标模块在装入内存时，采取边装入边链接的方式
- 运行时动态链接：对某些目标模块，是在程序执行中需要该目标模块时，才对它进行链接，其优点是便于修改和更新，便于实现对目标模块的共享

可重入代码（又叫纯代码）：允许多个进程同时访问的代码，但不允许进程修改它

可定位代码：即可重定位的代码

排除 AB，

D：静态链接的话，以后是不允许拆开的，排除，选 C

65.虚存管理和实存管理的主要区别是（C）

- A.虚存分逻辑地址和物理地址，实存不分
- B.实存要求一程序在内存必须连续，虚存不需要连续的内存
- C.实存要求一程序全部装入内存才开始运行，虚存允许程序在运行的过程中逐步装入
- D.虚存以逻辑地址执行程序，实存以物理地址执行程序

背景：现在的内存容量有限，但是外存容量发展很快，如果使用实存访问大外存中的大数据的话，计算机就不行了。。。所以有了虚存的产生

虚存概念：利用有限资源尽可能执行大程序，对用户程序和内存进行分块，将用户程序一块一块的拿到内存中来，一个执行完后释放内存，

下一个继续执行

虚拟存储通过将逻辑地址和物理地址分开，从而使得提供大于物理地址的逻辑地址空间成为可能

虚存管理和实存管理的区别：

虚存管理部分装入和部分对换，而实存管理不允许这样做，所谓部分装入：指一道应用程序不是全部装入内存以后才开始执行，而是装入其中一部分或者一点都不装入就开始执行，部分对换：指当内存已满而又有新的部分需要装入时，把已经在内存的某一部分替换出去，以腾出空间存放新来者，部分装入和部分对换的结果是可用用较小的内存允许较大的程序，而实存管理技不同，它要求整体装入

A，都分物理地址和逻辑地址

B，都不要求连续内存空间

D，都是以物理地址执行程序

66.在下列有关请求分页管理的叙述中，正确的是（D）

A.程序和数据是在开始执行前一次性装入的

B.产生缺页中断一定要淘汰一个页面

C.一个淘汰的页面一定要写回外存

D.在页表中要有"中断位"、"访问位"等信息

A，在请求分页系统中，只要求将当前一部分页面装入内存，便可以启动作业运行，并不需要一次全部装入，在作业执行的过程中，当访问的页面不存在的时，再通过调页功能将其调入，同时还可以通过置换功能将暂时不用的页面换出到外存，以便腾出内存空间

B，在请求分页系统中，当要访问的页面不存在的时，便会产生一个缺页中断，请求 os 将该页调入内存中，当内存中有空闲内存块的时候，将需要的页面直接调入空闲内存块，当内存中没有空闲内存块的时候，淘汰掉一个页面，然后将需要调入的页面调入

C，淘汰掉一个页面的时候，如果该页面没有被修改的话，便不用写回外存

D，请求分页系统为了解决和处理将要访问的页面不存在的情况，在请求页表项中增加了 4 个子段

- 页表构成：页号+物理块号+状态位 p+访问字段 A+修改位 M+外存地址
- 状态位 P：标记该页是否已被调入内存中，供程序访问时参考
- 访问字段位 A：记录本页在一段时间内被访问的次数，供页面淘汰释算法参考
- 修改位：标记该页面在调入内存后是否被修改
- 外存地址：该页在外存的地址，供写回外存和从外存中调入该页时参考

67.LRU 置换算法所基于的思想是（D）

A.在最近的过去用得少的在最近的将来也得少

B.在最近的过去用得多的在最近的将来也得多

C.在最近的过去很久未使用的在最近的将来会使用

D.在最近的过去很久未使用的在最近的将来也不会使用

LRU 算法（LeastRecentlyUsed）：近期最少使用算法，在过去很久没有使用的算法在最近的将来页不会使用，理想型算法，不可能实现

68.在下面关于虚拟存储器的叙述中， 正确的是（B）

- A.要求程序运行前必须全部装入内存且在运行过程中一直驻留在内存
- B.要求程序运行前不必全部装入内存且在运行过程中不必一直驻留在内存
- C.要求程序运行前不必全部装入内存但是在运行过程中必须一直驻留在内存
- D.要求程序运行前必须全部装入内存但在运行过程中不必一直驻留在内存

虚存概念： 利用有限资源尽可能执行大程序， 对用户程序和内存进行分块， 将用户程序一块一块的拿到内存中来， 一个执行完后释放内存， 下一个继续执行

69 在请求分页系统中， 页表中的改变位是供（C） 参考的

- A.页面置换
- B.内存分配
- C.页面换出， 决定是否写入磁盘
- D.页面调入

修改位（又叫改变位）： 标记该页面在调入内存后是否被修改， 从而决定是否需要写回外存， 没有被修改的话， 就没有必要写回外存

70.在请求分页系统中， 页表中的访问位是供（A） 参考的

- A.页面置换
- B.内存分配
- C.页面换出
- D.页面调入

访问字段位 A： 记录本页在一段时间内被访问的次数， 供页面淘汰释算法参考， 不太理解页面换出和页面置换的区别， 但 A 肯定是正确的， 至于 c 个人觉得也可以。。。。。

71.在请求分页管理中， 已修改过的页面再次装入时应来自（B）

- A.磁盘文件区
- B.磁盘对换区
- C.后备作业区
- D.I/O 缓冲区

请求分页系统缺页时， 如果页面从未被修改则从磁盘文件区调入， 对于那些被修改过的页面， 在将他们换出时， 便调出到对换区， 以后需要时， 再从对换区调入， 以为对换区的 IO 速度比文件区的快， 这样下次用这些被修改的文件的时候（被修改的文件再次被使用的概率大）速度就快一点

72.选择在最近的过去使用次数最少的页面予以淘汰的算法称为（D）

- A.Opt
- B.LRU
- C.MFU
- D.LFU

- OPT 算法（OptimalReplacement）： 淘汰的是以后不再使用或者未来长时间不再被访问的页面， 无法实现， 理想算法
- LRU 算法（LeastRecentlyUsed）： 最久未使用算法， 选择最近一段时间内最久没有被使用的页面进行置换
- LFU 算法（LeastFrequentlyUsed）： 最近最少使用算法， 选择最近一段时间使用得最少的页面置换

LRU 与 LFU 区别：

- LRU： 最久为使用， 和时间有关， 和使用次数无关
- LFU： 最不经常使用， 和使用次数有关， 和时间无关

73.选择在最近的过去很久未访问的页面予以淘汰的算法称为（B）。

A.Opt. B.LRU C.MFU D.LFU

74.程序动态链接的时刻是（B）

A.编译时 B.装入时，运行时 C.调用时 D.紧凑时

BC 都可以，只选 B 也没有错。。。。，因为单选，所以选 B

有装入时动态链接，也有运行时动态链接，调用时也是运行时

编译时：检查语法错误的时候

装入时，运行时，调用时：一个意思

紧凑时：预处理阶段，将声明引入的一些指令去掉，比如#include，#define

75.虚存的可行性基础是（C）

A.程序执行的离散性 B.程序执行的顺序性
C.程序执行的局部性 D.程序执行的并发性

虚存的可行性基础是计算机中著名的局部性

局部性表现为两个方面

- 时间局部性：程序中的某条指令一旦执行，不久之后该指令可能再次执行，因为程序中有大量循环操作
- 空间局部性：一旦程序访问了某个存储单元，在不久之后，其附件的存储单元也将会被访问，即程序一段时间内，访问的地址可能集中在一定范围内，因为指令一般是顺序存放，顺序执行的，数据一般页式以数组，向量，表等聚簇方式储存的

虚存概念：利用有限资源尽可能执行大程序，对用户程序和内存进行分块，将用户程序一块一块的拿到内存中来，一个执行完后释放内存，下一个继续执行

虚拟存储通过将逻辑地址和物理地址分开，从而使得提供大于物理地址的逻辑地址空间成为可能

所以选 C

76.实现虚存最主要的技术是（C）

A.整体覆盖 B.整体对换 C.部分对换 D.多道程序设计

考察虚存概念

虚存概念：利用有限资源尽可能执行大程序，对用户程序和内存进行分块，将用户程序一块一块的拿到内存中来，一个执行完后释放内存，下一个继续执行，所以虚存最主要的技术是部分对换

77.首次适应算法的空闲区是（A）

A.按地址递增顺序链在一起 B.始端指针表指向最大空闲区

C.按大小递增顺序链在一起 D.寻找从最大空闲区开始

首次适应算法:要求空闲分区以地址递增的次序连接，在分配内存时，从首地址开始找，找到第一个满足要求的空闲分区，分配出去

78.采用（B）不会产生内部碎片

A.分页式存储管理 B.分段式存储管理

C.固定分区式存储管理 D.段页式存储管理

- 内部碎片：指已经分配出去的，但是分配的大于实际需要的，比如需要 1，你给了 10，那么浪费了 9
- 外部碎片：指没有被分配出去的，由于太小而无法分配出去，比如一个块大小为 1，最小的都需要 2，1 就没有人要了
- 固定分区：处理作业之前就已经被划分为若干个分区，每个分区大小可以相同或者不同，且一旦划分好之后，分区大小不可以再改变，存在内部碎片
- 可变式分区：（动态划分，你要多大内存我就给你多大内存，c++stl 中的 vector）存在外部碎片
- 分页式储存管理：划分成一页一页的，也有动态和静态之分，都存在内部碎片
- 分段式：存在外部碎片
- 段页式：为了有效利用内存，产生更少碎片，分页的话内存以页为单位来使用，最后一页往往装不满，所以有内部碎片，为了共享要分段，在段的换入换出时形成了外部碎片，比如 5k 段换出后，换入一个 4k 段，于是就形成了 1k 的外部碎片

79.下面哪种内存管理方法有利于程序的动态链接？（A）

A.分段存储管理 B.分页存储管理

C.可变分区分配 D.固定分区分配

动态链接：指在作业运行前，不将目标程序段链接起来而是运行时，先将程序对应的目标程序段装入内存并启动运行，在运行的过程中，需要调用某段的时候，才将目标段调入内存并进行链接，所以动态链接是以段作为管理的单位

80.在虚拟存储系统中，若进程在内存中占三块(开始时为空)，采用先进先出页面淘汰算法，当执行访问页号序列为 1、2、3、4、1、2、5、1、2、3、4、5、6 时，将产生（D）次缺页中断

A.7 B.8 C.9 D.10

开始三个都是空的，所以一开始会产生三次，装入 123 之后，会产生 7 次，所以总共 10 次

81.系统"抖动"现象的发生是由（A）引起的？

A.置换算法选择不当 B.交换的信息量过大

C.内存容量充足 D.请求页式管理方案

置换算法选择不当，导致频繁置换，消耗资源，这种现象叫系统抖动

82 采用段式存储管理的系统中，若地址用 24 位表示，其中 8 位表示段号，则允许每段的最大长度是（B）。

- A.2²⁴
- B.2¹⁶
- C.2⁸
- D.2³²

长度位数：24-8=16，所以是 2 的 16 次方

83.在请求分页存储管理的系统中，若采用 FIFO 页面淘汰算法，则当进程分配到的页面数增加时，缺页中断的次数（D）。

- A.减少
- B.增加
- C.无影响
- D.可能增加也可能减少

可能会有抖动现象，可能没有，所以可能增加可能减少

84.在固定分区分配中，每个分区的大小是（C）

- A.相同
- B.随作业长度变化
- C.可以不同但预先固定
- D.可以不同但根据作业长度变化

固定分区：处理作业之前就已经被划分为若干个分区，每个分区大小可以相同或者不同，且一旦划分好之后，分区大小不可以再改变，存在内部碎片

85.在页式存储管理系统中，页表内容如表所列。

页表：页号 块号

0	2
1	1
2	6
3	3
4	7

若页的大小为 4KB，则地址转换机构将逻辑地址 0 转换成物理地址为（A）

- A.8192
- B.4096
- C.2048
- D.1024

页号为 0 表示逻辑地址为 0，对应的块号为 2，所以物理地址：1024*4*2=8192

86.某段表的内容如表所示

段表：段号 段首址/KB 段长度/KB

0	120	40
1	760	30
2	480	20
3	370	20

现执行某条指令 Load1， 2/154， 逻辑地址 2/154（其中段号为 2， 段内地址为 154）， 它对应的物理地址为（B）。

- A.120KB+2 B.480KB+154B C.30KB+154 D.480KB+2

段号为 2， 查表得首地址是 480KB， 段内地址是 154B， 所以物理地址： 480KB+154B

记住， 段长是 20KB， 而段内偏移地址是 154B， 所以没有产生越界

87.下述（a） 页面淘汰算法会产生 Belady 现象

- A.先进先出 B.最近最少使用 C.最不经常使用 D.最佳

所谓 Belady 现象： 在分页式虚拟存储器中， 发生缺页时采用 FIFO 置换算法， 出现的分配的分配页面数增多而缺页次数反而多的异常现象（比如你刚刚淘汰了 x， 后面又需要 x， 你刚刚淘汰 y， 后面马上又需要 y）

88.下面（B） 内存管理方法不能实现扩大内存？

- A.动态页式管理 B.静态页式管理
C.分段存储管理 D.段页式存储管理

- 静态页式管理： 在作业或者进程执行前， 把作业或者进程全部装入内存
- 动态页式管理： 不会一次性全部装入， 只装入反复被调用的部分， 其余部分需要的时候再装入， 无形中扩大了内存

89.设主存容量为 1MB， 辅存容量为 400MB， 计算机系统的地址寄存器有 24 位， 那么虚存的最大容量是（D）

- A.1MB B.401MB C.1MB+2²⁴ B D.2²⁴B

虚存： min（主存+辅存， 逻辑地址）

逻辑地址跟索引有关， 索引跟寄存器有关

所以虚存=min（401MB， 2 的 24 次方 B） =2 的 24 次方 B

90.在可变式分区分配方案中， 只需要进行一次比较就可以判定系统是否能满足作业对主存空间要求的算法是（C）

- A.最先适应算法 B.最佳适应算法 C.最差适应算法 D.固定式分区算法

最差适应算法： 每次选择最大的

判断一次， 如果最大的都不适合的话， 那么就没有适合的了

91.页式虚拟存储管理的主要特点是（B）

- A.不要求将作业装入到主存的连续区域
B.不要求将作业同时全部装入到主存的连续区域
C.不要求进行缺页中断处理
D.不要求进行页面置换

A 没有 B 全面

页式虚拟必须是请求的，说明这是请求页式虚拟储存

- 请求：不要求一次性全部装入
- 页式：存入的主存可以不连续

C 如果缺页不处理的话，程序运行不下去

D 页面不置换的话那没得玩了

92. 在请求分页存储管理中，当访问的页面不在内存时，便产生缺页中断，缺页中断是属于 (B)

A.I/O 中断 B.程序中断 C.访管中断 D.外中断

外中断：外设发出的中断，可屏蔽

内中断：硬件出错或者运行出错，不可屏蔽

中断分类:

一，强迫性中断（正在运行的程序没有想到的，来自硬件故障或者外部请求）

- IO 中断（外部 IO 设备给的，不是程序所预料到的，程序没有想到此刻会有 IO 中断）
- 程序性中断（溢出，缺页，缺段，地址越界，除 0）
- 时钟中断
- 控制台中断
- 硬件中断

二、自愿性中断（编程者预期的）

用户在编程时要求操作系统提供服务，使用访管指令或系统调用使中断发生，叫做访管中断，包括执行 OI，创建进程，分配内存，信号量操作，发生接收消息

93. 在下面的 I/O 控制方式中，需要 CPU 干预最少的方式是 (D)

A.程序 I/O 方式 B.中断驱动 I/O 控制方式

C.直接存储器访问 DMA 控制方式 D.I/O 通道控制方式

- 程序 IO: cpu 采用轮询方式询问数据----效果最差, 此刻 cpu 只能做这个, 干不了其他事情
- 中断驱动 IO: 设备控制器取出一个数据后向 cpu 发送一个中断, 然后 cpu 将数据从设备控制器取到 cpu 寄存器, 然后转移到内存中
- DMA 方式: cpu 控制 DMA, DMA 自己进行 IO
- IO 通道: IO 通道相当于一个简单处理机, 有自己的指令, 可以执行指令, 不用 cpu 管了, 大大的解放了 cpu, cpu 干预最少

94.CPU 输出数据的速度远远超过打印机的打印速度，影响程序执行速度，为解决这一问题，可以采用 (D)。

A.通道技术 B.虚拟存储器 C.并行技术 D.缓冲技术

缓冲技术：协调吞吐速度差异很大的的设备的技术，比如 cpu 和输入设备，输入的数据先放到缓冲区，然后 cpu 自己拿，决定拿多少，这样 cpu 就不用等待输入设备输数据

95.利用通道实现了（C）之间数据的快速传输。

- A.CPU 和外设 B.内存和 COU C.内存和外设 D.外设和外设

通道：独立于 cpu 外的专门 IO 的处理机，完成外设和内存间数据传输

96.设从磁盘将一块数据传送到缓冲区所用实际为 80μs，将缓冲区中数据传送到用户区所用时间为 40μs，CPU 处理数据所用的时间为 30μs。

则处理该数据，采用单缓冲传送磁盘数据，系统所用总时间为（A）

- A.120μs B.110μs C.150μs D.70μs

80+max(40,30)=120

缓冲区存在的意义：

使用户输入输出数据的速度与 cpu 处理数据的速度协调

因而用户输入数据和 cpu 处理数据的过程可以并行，所以二者取 max 即可：80+max（40，30）=120

97.下列关于通道、设备、设备控制器三者间关系的叙述中正确的是（C）。

- A.设备控制器和通道可以分别控制设备
B.设备控制器控制通道和设备一起工作
C.通道控制设备控制器，设备控制器控制设备
D.设备控制器控制通道，通道控制设备

储存器和设备交互时，先选择通道-->设备控制器-->设备

98.假设就绪队列中有 10 个进程，以时间片轮转方式进行进程调度，时间片大小为 300ms，CPU 进行进程切换需要花费 10ms，则系统开销所

占比率和进程数目增加到 30 个，其余条件不变时系统开销所占的比率相比，其结果是（B）

- A.大于 B.等于 C.小于 D.不能确定

系统开销比率=调度耗时（进程切换耗时）/时间片长度，所以不变

99.在段页式存储管理系统中其虚拟地址空间是()的

- A.一维
B.二维，段号，页号
C.三维，段号，页号，页内偏移量
D.四维

没有正确答案

三个量，段号，页号，页内偏移量，属于二维：段号加页号组成的二维空间，页内偏移量和页号属于同一维

页式是一维

段式是二维

段页式仍然是二维

100.下面关于请求分段存储管理的叙述中说法正确是（B）。

- A.分段尺寸受内存空间的限制，且作业总的尺寸也受内存空间的限制。
- B.分段尺寸受内存空间的限制，但作业总的尺寸不受内存空间的限制。
- C.分段尺寸不受内存空间的限制，且作业总的尺寸不受内存空间的限制。
- D.分段尺寸不受内存空间的限制，但作业总的尺寸受内存空间的限制。

分段对应的是内存具体存储的一种方式，是对具体内存进行管理，段号+基址，分段尺寸最大为具体存储，在动态链接时先将主程序所对应的目标程序装入内存并启动，运行过程中需要调用某段时才将该段内存合并进行链接

作业的大小不受内存大小的限制，虚拟存储器可以解决内存不够的问题，允许作业装入的时候只装入一部分，另一部分放在磁盘上，当需要的时候再装入内存，这样一来，在一个小的内存空间就可以运行一个比它大的作业，同时，用户编程的时候摆脱了一定要编写小于主存容量作业的限制

101.通道是一种特殊的处理机，具有（A.）能力。

- A.执行 I/O 指令集。B.执行 CPU 指令集。
- C.传送 I/O 命令。D.运行 I/O 进程。

通道的目的是为了使得原来一些由 cpu 处理的 IO 任务交由通道来承担，从而解脱 cpu，通道所能执行的命令局限于 IO 操作的指令，也就是 IO 指令集

102.假定把磁盘上的一个数据块中的信息输入到一个双缓冲区的时间为 $T=0.1\text{ms}$ ，将缓冲区中的数据传送到用户区的时间 $M=0.05\text{ms}$ ，而 CPU 对数据的处理时间 $C=0.05\text{ms}$ ，这样系统对每个数据块的处理时间为（B）

- A.0.05ms B.0.1ms C.0.15ms D.0.25ms

单缓冲：缓冲区数据传输到用户区之后，才能再从磁盘读数据

处理时间= $M+\text{Max}(C,T)$

双缓冲：缓冲区的数据传输到用户区的同时，也能从磁盘读数据（理解为两个缓存工作）

处理时间= $\text{Max}(C,T,M)$

103.进程和程序的一个本质区别是（A.）

- A.前者为动态的，后者为静态的
- B.前者存储在内存，后者存储在外存
- C.前者在一个文件中，后者在多个文件中
- D.前者分时使用 CPU,后者独占 CPU

进程就是一个正在进行的程序，描述的是程序在处理机上的一次执行过程，所以最本质的区别肯定是动态和静态的区别

104.进程在系统中是否存在的惟一标志是（D）

- A.数据集合
- B.目标程序
- C.源程序
- D.进程控制块

105.虚拟存储器的最大容量是由（A.）决定的

- A.计算机系统的地址结构
- B.页表长度
- C.内存空间
- D.逻辑空间

虚拟存储器的最大容量=Min(内存+外存, 2 的 n 次方) n 为计算机的地址总线位数

题目问的是决定，决定这个词用的不好，因为内存也是决定因素之一，最大虚拟储存有两个概念，一个是理论上能达到最大的和实际允许最大的

理论上最大的话选 A，实际上最大的话选 C

但是因为内存大小也取决于地址总线，所以选 A

106.在段式存储管理中，一个段是一个（B.）区域

- A.定长的连续
- B.不定长的连续
- C.定长的不连续
- D.不定长的不连续

分段后，段表包含以下信息：段号+段长+段基地址+存取控制信息

所以段长是可以不固定的，但每个段内地址是连续的，这跟分页有点区别，分页中每个页面大小是固定的，可能连续可能不连续

107.外存上存放的数据（D）

- A.CPU 可直接访问
- B.CPU 不可访问
- C.是高速缓冲器中的信息
- D.必须在访问前先装入内存

常识，cpu 都是去内存拿数据，就算中断驱动 OI 的话，也是设备控制器取出一个数据后向 cpu 发送一个中断，然后 cpu 将数据从设备控制器取到 cpu 寄存器，然后转移到内存中，而不是直接在 cpu 寄存器拿数据

107.某进程在运行过程中等待的事件已发生，例如，打印结束，此时该进程的状态将（C.）。

- A.从就绪变为运行
- B.从运行变为就绪
- C.从运行变为阻塞
- D.从阻塞变为就绪

进程主要的三个状态：运行，就绪，阻塞，某进程在运行中等待的时间已经发生，注意是在运行中，运行中是可以等待 OI 来中断这个进程，此时开始打印，但是 cpu 就会去处理其他进程，这样这个进程就处于阻塞态

108.下述 (B) 不属于多道程序运行的特征。

- A.多道
- B.运行速度快
- C.宏观上并行
- D.实际上多道程序是串插运行的

- 多道程序的特点：多道，宏观上并行，微观上串行
- 多道程序优点：cpu 利用率高，设备利用率高，系统吞吐量大

注意多道并没有增加系统的运行速度，只是让 cpu 的利用率得到提升

109.在请求分页存储管理方案中，若某用户空间为 16 个页面，页长 1KB，现有页表如下，则逻辑地址 102B (H) 所对应的物理地址为 (C)。

- 0 1
- 1 5
- 2 3
- 3 7
- 4 2
- A.312B (H)
- B.202B (H)
- C.082B (H)
- D.282B (H)

分页储存中

逻辑地址：页号+页内偏移

物理地址：块号+页内偏移

16 个页面，是 2 的四次方，所以页号占 4 位，页长为 1k，所以页面大小（页内偏移）占 10 位

102B(H)是 16 进制，变成 2 进制

0001000000101011，16 位

变成 14 位

01000000101011，前面 4 位是页号，后面 10 位是页内偏移地址

页号：0100 对应 10 进制 4，所以页号是 4，查表得到页号 4 对应的块号是 2

所以物理地址是 $2 \times 1K + 00101011 = 100000101011$

转换为 16 进制：082B(H)

110.原语是一种特殊的系统调用命令，它的特点是 (D.)

- A.功能强
- B.自己调用自己
- C.可被外层调用
- D.执行时不可中断

原语具有不可分割性，所以执行的时候不可中断

111.一个进程被唤醒意味着 (B)

- A.该进程重新占有了 cpu

B.进程状态变为就绪
- C.它的优先权变为最大

D.其 pcb 移至就绪队列的队首

对于 A： 进程被唤醒， 并不意味这该进程一定会在 cpu 上执行

对于 B： 进程被唤醒， 就是上该进程从阻塞变为就绪（联系信号量）

对于 C： 优先权可能边， 可能不变

对于 D： os 无此功能

112.现代操作系统的两个基本特征是（C.）和资源共享

- A.多道程序设计

B.中断处理
- C.程序的并发执行

D.实现分时与实时处理

并发和共享是操作系统最基本的两个特性， 他们又是互为存在条件， 一方面资源共享是以进程的并发性为执行条件的， 若系统不允许并发， 那么资源共享就没有任何意义， 另一方面系统若不能对共享的资源实现有效的管理， 则也必将影响进程的并发执行

113.时间片轮转法进行进程调度是为了(A)。

- A.多个终端都能得到系统的及时响应
- B.先来先服务
- C.优先级较高的进程得到及时响应
- D.需要 cpu 最短的进程先做

时间片轮转就是保证每个都能得到等概率的执行

114.若处理器有 32 位地址， 则它的虚拟地址空间为（B） 字节。

- A.2G
- B.4G
- C.100K
- D.640K

$2^{32} = 4G$

换算如下：

$2^{10}b = 1kb$

$2^{20}b = 1mb = 1kb*2^{10}$

$2^{30}b = 1gb = 1mb*2^{10}$

$2^{32}b = 2^2*1gb = 4gb$

115.操作系统内核与用户程序、应用程序之间的接口是（C）

- A.shell 命令
- B.图形界面
- C.系统调用
- D.C 语言函数

Unix 操作系统体系结构， 自底向上为： 硬件， 内核， 系统调用， shell 和库函数， 应用程序

个人理解：

自底向上：硬件，驱动，内核，用户接口程序，应用程序

用户接口程序：基于文件的叫 shell，基于图标的图形化界面叫 GUI

这里选 C，系统调用包括了 shell 和 GUI

116 在一段时间内，只允许一个进程访问的资源称为（C）。

- A.共享资源 B.独占临源 C.临界资源 D.共享区

临界资源：一段时间内，一次仅允许一个进程使用的资源

117.操作系统采用缓冲技术，能够减少对 CPU 的（A）次数，从而提高资源的利用率。

- A.中断 B.访问 C.控制 D.依赖

引入缓冲的原因：缓和 cpu 和 IO 设备速度不匹配的矛盾，减少对 cpu 的中断频率，放宽对中断响应时间的限制，提供 cpu 和 IO 设备间的并行性，所以采用缓冲技术，可减少 cpu 的中断次数，从而提高系统效率

118.文件的存储方法依赖于（C）

- A.文件的物理结构
B.存放文件的存储设备的特性
C.A 和 B.
D.文件的逻辑结构

文件的结构就是文件的形式，从用户的观点所看到的文件组织形式就是文件的逻辑结构，从实现的观点看到的文件在外存上的存放方式就是文件的物理结构，文件的逻辑结构和储存设备特性无关，但文件的物理结构于储存设备的特性有很大的关系

119.目录文件所存放的信息是（D.）

- A.某一文件存放的数据信息 B.某一个文件的文件目录
C.该目录中所有数据文件目录 D.该目录中所有子目录文件和数据文件的目录

- 文件目录：把所有的 FCB（文件控制块）组织在一起，就构成了文件目录，即文件控制块的有序集合
- 目录文件：为了实现对目录文件的管理，通常将文件目录以文件的形式保存在外存，这个文件就叫做目录文件

120.在现代操作系统中采用缓冲技术的主要目的是（C）。

- A.改善用户编程环境 B.提高 CPU 的处理速度
C.提高 CPU 和设备之间的并行程度 D.实现与设备无关性

为了缓和 cpu 和 IO 设备速度的不匹配，提高 cpu 与 IO 设备的并行性，在现代 OS 中，几乎所有的 IO 设备在于处理机交互数据时都采用了缓冲技术，并提高和活动释放缓冲区的手段，总结来说，缓冲技术用到了缓冲区，而缓冲区的引入是为了缓和 cpu 与 IO 设备速度的不匹配，减少 cpu 的中断频率，提高 cpu 和 IO 设备的并行性