

第一章

1. 设计现代 OS 的主要目标是什么？

答：（1）有效性 （2）方便性 （3）可扩充性 （4）开放性

2. OS 的作用可表现在哪几个方面？

答：（1）OS 作为用户与计算机硬件系统之间的接口

（2）OS 作为计算机系统资源的管理者

（3）OS 实现了对计算机资源的抽象

3. 为什么说 OS 实现了对计算机资源的抽象？

答：OS 首先在裸机上覆盖一层 I/O 设备管理软件，实现了对计算机硬件操作的第一层次抽象；在第一层软件上再覆盖文件管理软件，实现了对硬件资源操作的第二层次抽象。OS 通过在计算机硬件上安装多层系统软件，增强了系统功能，隐藏了对硬件操作的细节，由它们共同实现了对计算机资源的抽象。

4. 试说明推动多道批处理系统形成和发展的主要动力是什么？

答：主要动力来源于四个方面的社会需求与技术发展：

（1）不断提高计算机资源的利用率；

（2）方便用户；

（3）器件的不断更新换代；

（4）计算机体系结构的不断发展。

5. 何谓脱机 I/O 和联机 I/O？

答：脱机 I/O 是指事先将装有用户程序和数据的纸带或卡片装入纸带输入机或卡片机，在外围机的控制下，把纸带或卡片上的数据或程序输入到磁带上。该方式下的输入输出由外围机控制完成，是在脱离主机的情况下进行的。

而联机 I/O 方式是指程序和数据的输入输出都是在主机的直接控制下进行的。

6. 试说明推动分时系统形成和发展的主要动力是什么？

答：推动分时系统形成和发展的主要动力是更好地满足用户的需要。主要表现在：CPU 的分时使用缩短了作业的平均周转时间；人机交互能力使用户能直接控制自己的作业；主机的共享使多用户能同时使用同一台计算机，独立地处理自己的作业。

7. 实现分时系统的关键问题是什么？应如何解决？

答：关键问题是当用户在自己的终端上键入命令时，系统应能及时接收并及时处理该命令，在用户能接受的时延内将结果返回给用户。

解决方法：针对及时接收问题，可以在系统中设置多路卡，使主机能同时接收用户从各个终端上输入的数据；为每个终端配置缓冲区，暂存用户键入的命令或数据。针对及时处理问题，应使所有的用户作业都直接进入内存，并且为每个作业分配一个时间片，允许作业只在自己的时间片内运行，这样在不长的时间内，能使每个作业都运行一次。

8. 为什么要引入实时 OS?

答：实时操作系统是指系统能及时响应外部事件的请求，在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行。引入实时 OS 是为了满足应用的需求，更好地满

足实时控制领域和实时信息处理领域的需要。

9. 什么是硬实时任务和软实时任务？试举例说明。

答：硬实时任务是指系统必须满足任务对截止时间的要求，否则可能出现难以预测的结果。

举例来说，运载火箭的控制等。

软实时任务是指它的截止时间并不严格，偶尔错过了任务的截止时间，对系统产生的影响不大。举例：网页内容的更新、火车售票系统。

10. 在 8 位微机和 16 位微机中，占据了统治地位的是什么操作系统？

答：单用户单任务操作系统，其中最具代表性的是 CP/M 和 MS-DOS.

11. 试列出 Windows OS 中五个主要版本，并说明它们分别较之前一个版本有何改进。

答：

(1) Microsoft Windows 1.0 是微软公司在个人电脑上开发图形界面的首次尝试。

(2) Windows 95 是混合的 16 位/32 位系统，第一个支持 32 位。带来了更强大、更稳定、更实用的桌面图形用户界面，结束了桌面操作系统间的竞争。

(3) Windows 98 是微软公司的混合 16 位/32 位 Windows 操作系统，改良了硬件标准的支持，革新了内存管理，是多进程操作系统。

(4) Windows XP 是基于 Windows2000 的产品，拥有新用户图形界面月神 Luna。简化了用户安全特性，整合了防火墙。

(5) Windows Vista 包含了上百种新功能；特别是新版图形用户界面和 WindowsAero

全新界面风格、加强的搜寻功能（Windows IndexingService）、新媒体创作工具以及重新设计的网络、音频、输出（打印）和显示子系统。。

12. 试从交互性、及时性以及可靠性方面，将分时系统与实时系统进行比较。

答：（1）及时性：实时信息处理系统对实时性的要求与分时系统类似，都是以人所能接受的等待时间来确定；而实时控制系统的及时性，是以控制对象所要求的开始截止时间或完成截止时间来确定的，一般为秒级到毫秒级，甚至有的要低于 100 微妙。

（2）交互性：实时信息处理系统具有交互性，但人与系统的交互仅限于访问系统中某些特定的专用服务程序。不像分时系统那样能向终端用户提供数据和资源共享等服务。

（3）可靠性：分时系统也要求系统可靠，但相比之下，实时系统则要求系统具有高度的可靠性。因为任何差错都可能带来巨大的经济损失，甚至是灾难性后果，所以在实时系统中，往往都采取了多级容错措施保障系统的安全性及数据的安全性。

13. OS 有哪几大特征？其最基本的特征是什么？

答：并发性、共享性、虚拟性和异步性四个基本特征；最基本的特征是并发性。

14. 处理机管理有哪些主要功能？它们的主要任务是什么？

答：处理机管理的主要功能是：进程管理、进程同步、进程通信和处理机调度；

进程管理：为作业创建进程，撤销已结束进程，控制进程在运行过程中的状态转换。

进程同步：为多个进程（含线程）的运行_____进行协调。

通信：用来实现在相互合作的进程之间的信息交换。

处理机调度：

（1）作业调度。从后备队里按照一定的算法，选出若干个作业，为他们分配运行所需的资源（首选是分配内存）。

（2）进程调度：从进程的就绪队列中，按照一定算法选出一个进程，把处理机分配给它，并设置运行现场，使进程投入执行。

15. 内存管理有哪些主要功能？他们的主要任务是什么？

答：内存管理的主要功能有：内存分配、内存保护、地址映射和内存扩充。

内存分配：为每道程序分配内存。

内存保护：确保每道用户程序都只在自己的内存空间运行，彼此互不干扰。

地址映射：将地址空间的逻辑地址转换为内存空间与对应的物理地址。

内存扩充：用于实现请求调用功能，置换功能等。

16. 设备管理有哪些主要功能？其主要任务是什么？

答：主要功能有：缓冲管理、设备分配和设备处理以及虚拟设备等。

主要任务：完成用户提出的 I/O 请求，为用户分配 I/O 设备；提高 CPU 和 I/O 设备的利用率；提高 I/O 速度；以及方便用户使用 I/O 设备。

17. 文件管理有哪些主要功能？其主要任务是什么？

答：文件管理主要功能：文件存储空间的管理、目录管理、文件的读/写管理和保护。

文件管理的主要任务：管理用户文件和系统文件，方便用户使用，保证文件安全性。

18. 是什么原因使操作系统具有异步性特征？

答：操作系统的异步性体现在三个方面：一是进程的异步性，进程以人们不可预知的速度向前推进，二是程序的不可再现性，即程序执行的结果有时是不确定的，三是程序执行时间的不可预知性，即每个程序何时执行，执行顺序以及完成时间是不确定的。

19. 模块接口法存在哪些问题？可通过什么样的途径来解决？

答：（1）模块接口法存在的问题：①在 OS 设计时，各模块间的接口规定很难满足在模块完

成后对接口的实际需求。②在 OS 设计阶段，设计者必须做出一系列的决定，每一个决定必

须建立在上一个决定的基础上。但模块化结构设计的各模块设计齐头并进，无法寻找可靠的顺序，造成各种决定的无序性，使程序设计人员很难做到设计中的每一步决定都建立在可靠的基础上，因此模块接口法被称为“无序模块法”。

（2）解决途径：将模块接口法的决定顺序无序变有序，引入有序分层法。

20. 在微内核 OS 中，为什么要采用客户/服务器模式？

答：C/S 模式具有独特的优点：(1)数据的分布处理和存储。(2)便于集中管理。(3)灵活性和可扩充性。(4)易于改编应用软件。

21. 试描述什么是微内核 OS。

答：1) 足够小的内核 2) 基于客户/服务器模式

3) 应用机制与策略分离原理 4) 采用面向对象技术。

22. 在基于微内核结构的 OS 中，应用了哪些新技术？

答：在基于微内核结构的 OS 中，采用面向对象的程序设计技术。

23. 何谓微内核技术？在微内核中通常提供了哪些功能？

答：把操作系统中更多的成分和功能放到更高的层次（即用户模式）中去运行，而留下一个尽量小的内核，用它来完成操作系统最基本的核心功能，称这种技术为微内核技术。在微内核中通常提供了进程（线程）管理、低级存储器管理、中断和陷入处理等功能。

24. 微内核操作系统具有哪些优点？它为何能有这些优点？

答：1) 提高了系统的可扩展性

2) 增强了系统的可靠性

3) 可移植性

4) 提供了对分布式系统的支持

5) 融入了面向对象技术

第二章

1. 什么是前趋图？为什么要引入前趋图？

答：前趋图(Precedence Graph)是一个有向无循环图，记为 DAG(DirectedAcyclic Graph)，用于描述进程之间执行的前后关系。

2. 画出下面四条语句的前趋图：

S1=a: =x+y; S2=b: =z+1; S3=c: =a - b; S4=w: =c+1;

答：其前趋图为：

3. 什么程序并发执行会产生间断性特征？

答：程序在并发执行时，由于它们共享系统资源，为完成同一项任务需要相互合作，致使这些并发执行的进程之间，形成了相互制约关系，从而使得进程在执行期间出现间断性。

4. 程序并发执行时为什么会失去封闭性和可再现性？

答：程序并发执行时，多个程序共享系统中的各种资源，因而这些资源的状态由多个程序改变，致使程序运行失去了封闭性，也会导致其失去可再现性。

5. 在操作系统中为什么要引入进程概念？它会产生什么样的影响？

答：为了使程序在多道程序环境下能并发执行，并对并发执行的程序加以控制和描述，在操作系统中引入了进程概念。

影响：使程序的并发执行得以实行。

6. 试从动态性，并发性和独立性上比较进程和程序？

答：(1)动态性是进程最基本的特性，表现为由创建而产生，由调度而执行，因得不到资源而暂停执行，由撤销而消亡。进程有一定的生命期，而程序只是一组有序的指令集合，是静态实体。

(2)并发性是进程的重要特征，同时也是 OS 的重要特征。引入进程的目的正是为了使其程序能和其它进程的程序并发执行，而程序是不能并发执行的。

(3)独立性是指进程实体是一个能独立运行的基本单位，也是系统中独立获得资源和独立调度的基本单位。对于未建立任何进程的程序，不能作为独立单位参加运行。

7. 试说明 PCB 的作用，为什么说 PCB 是进程存在的惟一标志？

答：PCB 是进程实体的一部分，是操作系统中最重要的记录型数据结构。作用是使一个在多道程序环境下不能独立运行的程序，成为一个能独立运行的基本单位，成为能与其它进程并发执行的进程。OS 是根据 PCB 对并发执行的进程进行控制和管理的。

8. 试说明进程在三个基本状态之间转换的典型原因。

答：（1）就绪状态→执行状态：进程分配到 CPU 资源

（2）执行状态→就绪状态：时间片用完

（3）执行状态→阻塞状态：I/O 请求

（4）阻塞状态→就绪状态：I/O 完成

9. 为什么要引入挂起状态？该状态有哪些性质？

答：引入挂起状态处于五种不同的需要：终端用户需要，父进程需要，操作系统需要，对换

北京石油化工学院信息工程学院计算机系 5/48

《计算机操作系统》习题参考答案余有明与计 07 和计 G09 的同学们编著 5/48

需要和负荷调节需要。处于挂起状态的进程不能接收处理机调度。

10. 在进行进程切换时，所要保存的处理机状态信息有哪些？

答：进行进程切换时，所要保存的处理机状态信息有：

（1）进程当前暂存信息

（2）下一指令地址信息

（3）进程状态信息

(4) 过程和系统调用参数及调用地址信息。

11. 试说明引起进程创建的主要事件。

答：引起进程创建的主要事件有：用户登录、作业调度、提供服务、应用请求。

12. 试说明引起进程被撤销的主要事件。

答：引起进程被撤销的主要事件有：正常结束、异常结束（越界错误、保护错、非法指令、特权指令错、运行超时、等待超时、算术运算错、I/O 故障）、外界干预（操作员或操作系统干预、父进程请求、父进程终止）。

13. 在创建一个进程时所完成的主要工作是什么？

答：

- (1) OS 发现请求创建新进程事件后，调用进程创建原语 `Creat()`；
- (2) 申请空白 PCB；
- (3) 为新进程分配资源；
- (4) 初始化进程控制块；
- (5) 将新进程插入就绪队列。

14. 在撤销一个进程时所完成的主要工作是什么？

答：

- (1) 根据被终止进程标识符，从 PCB 集中检索出进程 PCB，读出该进程状态。
- (2) 若被终止进程处于执行状态，立即终止该进程的执行，置调度标志真，指示该进程被终止后重新调度。
- (3) 若该进程还有子进程，应将所有子孙进程终止，以防它们成为不可控进程。
- (4) 将被终止进程拥有的全部资源，归还给父进程，或归还给系统。
- (5) 将被终止进程 PCB 从所在队列或列表中移出，等待其它程序搜集信息。

15. 试说明引起进程阻塞或被唤醒的主要事件是什么？

答：a. 请求系统服务；b. 启动某种操作；c. 新数据尚未到达；d. 无新工作可做。

16. 进程在运行时存在哪两种形式的制约？并举例说明之。

答：

- (1) 间接相互制约关系。举例：有两进程 A 和 B，如果 A 提出打印请求，系统已把唯一的

一台打印机分配给了进程 B，则进程 A 只能阻塞；一旦 B 释放打印机，A 才由阻塞改为就绪。

(2) 直接相互制约关系。举例：有输入进程 A 通过单缓冲向进程 B 提供数据。当缓冲空时，

计算进程因不能获得所需数据而阻塞，当进程 A 把数据输入缓冲区后，便唤醒进程 B；反之，当缓冲区已满时，进程 A 因没有缓冲区放数据而阻塞，进程 B 将缓冲区数据取走后便唤醒 A。

17. 为什么进程在进入临界区之前应先执行“进入区”代码？而在退出前又要执行“退出区”代码？

答：为了实现多个进程对临界资源的互斥访问，必须在临界区前面增加一段用于检查欲访问的临界资源是否正被访问的代码，如果未被访问，该进程便可进入临界区对资源进行访问，并设置正被访问标志，如果正被访问，则本进程不能进入临界区，实现这一功能的代码为“进入区”代码；在退出临界区后，必须执行“退出区”代码，用于恢复未被访问标志，使其它进程能再访问此临界资源。

18. 同步机构应遵循哪些基本准则？为什么？

答：同步机构应遵循的基本准则是：空闲让进、忙则等待、有限等待、让权等待

原因：为实现进程互斥进入自己的临界区。

19. 试从物理概念上说明记录型信号量 wait 和 signal。

答：wait(S)：当 $S.value > 0$ 时，表示目前系统中这类资源还有可用的。执行一次 wait 操作，意味着进程请求一个单位的该类资源，使系统中可供分配的该类资源减少一个，因此描述为 $S.value := S.value - 1$ ；当 $S.value < 0$ 时，表示该类资源已分配完毕，进程应调用 block 原语自我阻塞，放弃处理机，并插入到信号量链表 S.L 中。

signal(S)：执行一次 signal 操作，意味着释放一个单位的可用资源，使系统中可供分配的该类资源数增加一个，故执行 $S.value := S.value + 1$ 操作。若加 1 后 $S.value \leq 0$ ，则表示在该信号量链表中，仍有等待该资源的进程被阻塞，因此应调用 wakeup 原语，将 S.L 链表中的第一个等待进程唤醒。

20. 你认为整型信号量机制是否完全遵循了同步机构的四条准则？

答：整型信号量机制不完全遵循同步机制的四条准则，它不满足“让权等待”准则。

21. 如何利用信号量机制来实现多个进程对临界资源的互斥访问？并举例说明之。

答：为使多个进程互斥访问某临界资源，只需为该资源设置一互斥信号量 **mutex**，并设其初值为 1，然后将各进程访问该资源的临界区 CS 置于 **wait(mutex)**和 **signal(mutex)**操作之间即可。这样，每个欲访问该临界资源的进程在进入临界区之前，都要先对 **mutex** 执行 **wait** 操作，若该资源此刻未被访问，本次 **wait** 操作必然成功，进程便可进入自己的临界区，这时若再有其他进程也欲进入自己的临界区，此时由于对 **mutex** 执行 **wait** 操作定会失败，因而该进程阻塞，从而保证了该临界资源能被互斥访问。当访问临界资源的进程退出临界区后，应对 **mutex** 执行 **signal** 操作，释放该临界资源。利用信号量实现进程互斥的进程描述如下：

```
Var mutex: semaphore:=1;
```

```
begin
```

```
parbegin
```

```
process 1: begin
```

```
repeat
```

```
wait(mutex);
```

```
critical section
```

```
signal(mutex);
```

```
remainder section
```

```
until false;
```

```
end
```

```
process 2: begin
```

```
repeat
```

```
wait(mutex);
```

```
critical section
```

```
signal(mutex);
```

```
remainder section
```

```
until false;
```

```
end
```

parend

22. 试写出相应的程序来描述图 2-17 所示的前驱图。

答：（a）Var a, b, c, d, e, f, g, h; semaphore:= 0, 0,0, 0, 0, 0, 0, 0;

begin

parbegin

begin S1; signal(a); signal(b); end;

begin wait(a); S2; signal(c); signal(d); end;

begin wait(b); S3; signal(e); end;

begin wait(c); S4; signal(f); end;

begin wait(d); S5; signal(g); end;

begin wait(e); S6; signal(h); end;

begin wait(f); wait(g); wait(h); S7; end;

parend

end

（b）Var a, b, c, d, e, f, g, h,i,j; semaphore:= 0,0, 0, 0, 0, 0, 0,0,0, 0;

begin

parbegin

begin S1; signal(a); signal(b); end;

begin wait(a); S2; signal(c); signal(d); end;

begin wait(b); S3; signal(e); signal(f); end;

begin wait(c); S4; signal(g); end;

begin wait(d); S5; signal(h); end;

begin wait(e); S6; signal(i); end;

begin wait(f); S7; signal(j); end;

begin wait(g);wait(h); wait(i); wait(j); S8;end;

parend

end

23. 在生产者消费者问题中，如果缺少了 `signal(full)`或 `signal(empty)`,对执行结果有何影响？

答：

如果缺少 `signal(full)`，那么表明从第一个生产者进程开始就没有改变信号量 `full` 值，即使缓冲池产品已满，但 `full` 值还是 0，这样消费者进程执行 `wait(full)`时认为缓冲池是空而取不到产品，消费者进程一直处于等待状态。

如果缺少 `signal(empty)`，在生产者进程向 `n` 个缓冲区投满产品后消费者进程才开始从中取产品，这时 `empty=0`，`full=n`，那么每当消费者进程取走一个产品 `empty` 值并不改变，直到缓冲池取空了，`empty` 值也是 0，即使目前缓冲池有 `n` 个空缓冲区，生产者进程要想再往缓冲池中投放产品也会因为申请不到空缓冲区被阻塞。

24. 在生产消费者问题中，如果将两个 `wait` 操作即 `wait(full)`和 `wait(mutex)`互换位置，或者将 `signal(mutex)`与 `signal (full)` 互换位置，结果如何？

答：将 `wait(full)`和 `wait(mutex)`互换位置后，可能引起死锁。考虑系统中缓冲区全满时，若一生产者进程先执行了 `wait(mutex)`操作并获得成功，则当再执行 `wait(empty)`操作时，它将因失败而进入阻塞状态，它期待消费者进程执行 `signal(empty)`来唤醒自己，在此之前，它不可能执行 `signal(mutex)`操作，从而使试图通过执行 `wait(mutex)`操作而进入自己的临界区的其他生产者和所有消费者进程全部进入阻塞状态，这样容易引起系统死锁。

若 `signal(mutex)`和 `signal(full)`互换位置后只是影响进程对临界资源的释放次序，而不会引起系统死锁，因此可以互换位置。

25. 我们在为某一临界资源设置一把锁 `W`，当 `W=1` 时表示关锁，当 `W=0` 时表示锁已打开。试写出开锁和关锁的原语，并利用他们实现互斥。

答：整型信号量：lock(W): while W=1 do no-op

W:=1;

unlock(W): W:=0;

记录型信号量：lock(W): W:=W+1;

if(W>1) then block(W, L)

```
unlock(W): W:=W-1;
```

```
if(W>0) then wakeup(W, L)
```

例子:

```
Var W:semaphore:=0;
```

```
begin
```

```
repeat
```

```
lock(W);
```

```
critical section
```

```
unlock(W);
```

```
remainder section
```

```
until false;
```

```
end
```

26. 试修改下面生产者—消费者问题解法中的错误:

答: producer:

```
begin
```

```
repeat
```

```
...
```

```
producer an item in nextp;
```

```
wait(mutex);
```

```
wait(full);
```

```
buffer(in):=nextp;
```

```
signal(mutex);
```

```
until false;
```

```

end
consumer:
begin
repeat
wait(mutex);
wait(empty);
nextc:=buffer(out);
out:=out+1;
signal(mutex);
consumer item in nextc;
until false;
end

```

27. 试利用记录型信号量写出一个不会出现死锁的哲学家进餐问题的算法.

答: Var chopstick:array[0,...,4] of semaphore;

所有信号量均被初始化为 1, 第 i 位哲学家的活动可描述为:

```

Repeat
Wait(chopstick[i]);
Wait(. chopstick[(i+1) mod 5]);
...
Eat ;
...
Signal(chopstick[i]);
Signal(chopstick[(i+1) mod 5])
Eat ;
...
Think;
Until false;

```

28. 在测量控制系统中的数据采集任务，把所采集的数据送一单缓冲区；计算任务从该单缓冲中取出数据进行计算.试写出利用信号量机制实现两者共享单缓冲的同步算法。

答：

a. Var mutex, empty, full: semaphore:=1, 1, 0;

gather:

begin

repeat

.....

gather data in nextp;

wait(empty);

wait(mutex);

buffer:=nextp;

signal(mutex);

signal(full);

until false;

end

compute:

begin

repeat

.....

wait(full);

wait(mutex);

nextc:=buffer;

signal(mutex);

signal(empty);

compute data in nextc;

until false;

```

end

b. Var empty, full: semaphore:=1, 0;

gather:

begin

repeat

.....

gather data in nextp;

wait(empty);

buffer:=nextp;

signal(full);

until false;

end

compute:

begin

repeat

.....

wait(full);

nextc:=buffer;

signal(empty);

compute data in nextc;

until false;

end

```

29. 画图说明管程由哪几部分组成，为什么要引入条件变量？

答：管程由四部分组成：①管程的名称；②局部于管程内部的共享数据结构说明；③对该数据结构进行操作的一组过程；④对局部于管程内部的共享数据设置初始值的语句；

当一个进程调用了管程，在管程中时被阻塞或挂起，直到阻塞或挂起的原因解除，而在此期间，如果该进程不释放管程，则其它进程无法进入管程，被迫长时间地等待。为了解决这个

问题，引入了条件变量 `condition`。

30. 如何利用管程来解决生产者与消费者问题？

答：首先建立一个管程，命名为 `ProclucerConsumer`，包括两个过程：

(1) `Put (item)` 过程。生产者利用该过程将自己生产的产品放到缓冲池，用整型变量 `count` 表示在缓冲池中已有的产品数目，当 `count ≥ n` 时，表示缓冲池已满，生产者须等待。

(2) `get (item)` 过程。消费者利用该过程从缓冲池中取出一个产品，当 `count ≤ 0` 时，表示缓冲池中已无可取的产品，消费者应等待。

PC 管程可描述如下：

```
type producer-consumer =monitor
Var in,out,count:integer;
buffer:array[0,...,n-1]of item;
notfull, notempty:condition;
procedure entry dot(item)
begin
if count>=n then not full.wait;
buffer(in):=nextp;
in:=(in+1)mod n;
count:=count+1;
if notempty.queue then notempty.signal;
end
procedure entry get(item)
begin
if count<=0 then not full.wait;
nextc:=buffer(out);
out:=(out+1)mod n;
count:=count-1;
if notfull.quene then notfull.signal;
```


end

begin in:=out:=0;

count:=0

end

在利用管程解决生产者—消费者问题时，其中的生产者和消费者可描述为：

producer: begin

repeat

produce an item in nextp

PC.put(item);

until false;

end

consumer: begin

repeat

PC.get(item);

consume the item in nextc;

until false;

end

31. 什么是 AND 信号量？试利用 AND 信号量写出生产者—消费者问题的解法。

答：为解决并行带来的死锁问题，在 wait 操作中引入 AND 条件，其基本思想是将进程在整个运行过程中所需要的所有临界资源，一次性地全部分配给进程，用完后一次性释放。

解决生产者—消费者问题可描述如下：

var mutex,empty,full: semaphore:=1,n,0;

buffer: array[0,...,n-1] of item;

in,out: integer:=0,0;

begin

parbegin

producer: begin

repeat

...

produce an item in nextp;

...

wait(empty);

wait(s1,s2,s3,...,sn); //s1,s2,...,sn 为执行生产者进程除 empty 外其余的条件

wait(mutex);

buffer(in):=nextp;

in:=(in+1) mod n;

signal(mutex);

signal(full);

signal(s1,s2,s3,...,sn);

until false;

end

consumer: begin

repeat

wait(full);

wait(k1,k2,k3,...,kn); //k1,k2,...,kn 为执行消费者进程除 full 外其余的条件

wait(mutex);

nextc:=buffer(out);

out:=(out+1) mod n;

signal(mutex);

signal(empty);

signal(k1,k2,k3,...,kn);

consume the item in nextc;

until false;

end

parend

end

32. 什么是信号量集？试利用信号量集写出读者—写者问题的解法。

答：对 AND 信号量加以扩充，形成的信号量集合的读写机制。

解法：Var RN integer;

L,mx: semaphore:=RN,1;

begin

parbegin

reader:begin

repeat

Swait(L,1,1);

Swait(mx,1,1);

...

perform read operation;

...

Ssignal(L,1);

until false

end

writer:begin

repeat

Swait(mx,1,1;L,RN,0);

perform write operation;

Ssignal(mx,1);

until false

end

parend

end

33. 试比较进程间的低级与高级通信工具。

答：用户用低级通信工具实现进程通信很不方便，效率低，通信对用户不透明，所有操作都必须由程序员来实现，而高级通信工具弥补了这些缺陷，用户直接利用操作系统提供的一组

通信命令，高效地传送大量的数据。

34. 当前有哪几种高级通信机制？

答：共享存储器系统、消息传递系统以及管道通信系统。

35. 消息队列通信机制有哪几方面的功能？

答：（1）构成消息（2）发送消息（3）接收消息（4）互斥与同步。

36. 为什么要在 OS 中引入线程？

答：在操作系统中引入线程，则是为了减少程序在并发执行时所付出的时空开销，使 OS 具有更好的并发性，提高 CPU 的利用率。进程是分配资源的基本单位，而线程则是系统调度的基本单位。

37. 试说明线程具有哪些属性？

答：（1）轻型实体（2）独立调度和分派的基本单位（3）可并发执行（4）共享进程资源。

38. 试从调度性，并发性，拥有资源及系统开销方面对进程和线程进行比较。

答：

（1）调度性。线程在 OS 中作为调度和分派的基本单位，进程只作为资源拥有的基本单位。

（2）并发性。进程可以并发执行，一个进程的多个线程也可并发执行。

（3）拥有资源。进程始终是拥有资源的基本单位，线程只拥有运行时必不可少的资源，本身基本不拥有系统资源，但可以访问隶属进程的资源。

（4）系统开销。操作系统在创建、撤消和切换进程时付出的开销显著大于线程。

39. 为了在多线程 OS 中实现进程之间的同步与通信，通常提供了哪几种同步机制？

答：同步功能可以控制程序流并访问共享数据，从而并发执行多个线程。共有四种同步模型：

互斥锁、读写锁、条件变量和信号。

40. 用于实现线程同步的私用信号量和公用信号量之间有何差别？

答：

（1）私用信号量。当某线程需利用信号量实现同一进程中各线程之间的同步时，可调用创建信号量的命令来创建一个私用信号量，其数据结构存放在应用程序的地址空间中。

（2）公用信号量。公用信号量是为实现不同进程间或不同进程中各线程之间的同步而设置的。其数据结构是存放在受保护的系统存储区中，由 OS 为它分配空间并进行管理。

41. 何谓用户级线程和内核支持线程？

答：

（1）用户级线程：仅存在于用户空间中的线程，无须内核支持。这种线程的创建、撤销、线程间的同步与通信等功能，都无需利用系统调用实现。用户级线程的切换通常发生在一个应用进程的诸多线程之间，同样无需内核支持。

（2）内核支持线程：在内核支持下运行的线程。无论是用户进程中的线程，还是系统进程中的线程，其创建、撤销和切换等都是依靠内核，在内核空间中实现的。在内核空间里还为每个内核支持线程设置了线程控制块，内核根据该控制块感知某线程的存在并实施控制。

42. 试说明用户级线程的实现方法。

答：用户级线程是在用户空间中的实现的，运行在“运行时系统”与“内核控制线程”的中间系统上。运行时系统用于管理和控制线程的函数的集合。内核控制线程或轻型进程 LWP 可通过系统调用获得内核提供服务，利用 LWP 进程作为中间系统。

43. 试说明内核支持线程的实现方法。

答：系统在创建新进程时，分配一个任务数据区 PTDA，其中包括若干个线程控制块 TCB 空间。创建一个线程分配一个 TCB，有关信息写入 TCB，为之分配必要的资源。当 PTDA 中的 TCB 用完，而进程又有新线程时，只要所创建的线程数目未超过系统允许值，系统可在为之分配新的 TCB；在撤销一个线程时，也应回收线程的所有资源和 TCB。

第三章

第三章处理机调度与死锁

1. 高级调度与低级调度的主要任务是什么？为什么要引入中级调度？

答：高级调度的主要任务是根据某种算法，把外存上处于后备队列中的那些作业调入内存。

低级调度是保存处理机的现场信息，按某种算法先取进程，再把处理器分配给进程。

引入中级调度的主要目的是为了提高内存利用率和系统吞吐量。使那些暂时不能运行的进程不再占用内存资源，将它们调至外存等待，把进程状态改为就绪驻外存状态或挂起状态。

2. 何谓作业、作业步和作业流？

答：作业包含通常的程序和数据，还配有作业说明书。系统根据该说明书对程序的运行进行控制。批处理系统中是以作业为基本单位从外存调入内存。

作业步是指每个作业运行期间都必须经过若干个相对独立相互关联的顺序加工的步骤。

作业流是指若干个作业进入系统后依次存放在外存上形成的输入作业流；在操作系统的控制下，逐个作业进程处理，于是形成了处理作业流。

3. 在什么情况下需要使用作业控制块 JCB？其中包含了哪些内容？

答：每当作业进入系统时，系统便为每个作业建立一个作业控制块 JCB，根据作业类型将它插入到相应的后备队列中。

JCB 包含的内容通常有：1) 作业标识 2) 用户名称 3) 用户账户 4) 作业类型（CPU 繁忙型、I/O 繁忙型、批量型、终端型）5) 作业状态 6) 调度信息（优先级、作业已运行）7) 资源要求 8) 进入系统时间 9) 开始处理时间 10) 作业完成时间 11) 作业退出时间 12) 资源使用情况等

4. 在作业调度中应如何确定接纳多少个作业和接纳哪些作业？

答：作业调度每次接纳进入内存的作业数，取决于多道程序度。应将哪些作业从外存调入内存，取决于采用的调度算法。最简单的是先来服务调度算法，较常用的是短作业优先调度算法和基于作业优先级的调度算法。

5. 试说明低级调度的主要功能。

答：（1）保存处理机的现场信息（2）按某种算法选取进程（3）把处理机分配给进程。

6. 在抢占调度方式中，抢占的原则是什么？

答：抢占的原则有：时间片原则、优先权原则、短作业优先权原则等。

7. 在选择调度方式和调度算法时，应遵循的准则是什么？

答：

（1）面向用户的准则：周转时间短、响应时间快、截止时间的保证、优先权准则。

（2）面向系统的准则：系统吞吐量高、处理机利用率好、各类资源的平衡利用。

8. 在批处理系统、分时系统和实时系统中，各采用哪几种进程（作业）调度算法？

答：批处理系统的调度算法：短作业优先、优先权、高响应比优先、多级反馈队列调度算法。

分时系统的调度算法：时间片轮转法。

实时系统的调度算法：最早截止时间优先即 EDF、最低松弛度优先即 LLF 算法。

9. 何谓静态和动态优先级？确定静态优先级的依据是什么？

答：静态优先级是指在创建进程时确定且在进程的整个运行期间保持不变的优先级。

动态优先级是指在创建进程时赋予的优先权，可以随进程推进或随其等待时间增加而改变的优先级，可以获得更好的调度性能。

确定进程优先级的依据：进程类型、进程对资源的需求和用户要求。

10. 试比较 FCFS 和 SPF 两种进程调度算法。

答：相同点：两种调度算法都可以用于作业调度和进程调度。

不同点：**FCFS** 调度算法每次都从后备队列中选择一个或多个最先进入该队列的作业，将它们调入内存、分配资源、创建进程、插入到就绪队列。该算法有利于长作业/进程，不利于短作业/进程。**SPF** 算法每次调度都从后备队列中选择一个或若干个估计运行时间最短的作业，调入内存中运行。该算法有利于短作业/进程，不利于长作业/进程。

11. 在时间片轮转法中，应如何确定时间片的大小？

答：时间片应略大于一次典型的交互需要的时间。一般应考虑三个因素：系统对相应时间的要求、就绪队列中进程的数目和系统的处理能力。

12. 通过一个例子来说明通常的优先级调度算法不能适用于实时系统？

答：实时系统的调度算法很多，主要是基于任务的开始截止时间和任务紧急/松弛程度的任务优先级调度算法，通常的优先级调度算法不能满足实时系统的调度实时性要求而不适用。

13. 为什么说多级反馈队列调度算法能较好地满足各方面用户的需求？

答：（1）终端型作业用户提交的作业大多属于较小的交互型作业，系统只要使这些作业在第一队列规定的时间片内完成，终端作业用户就会感到满足。

（2）短批处理作业用户，开始时像终端型作业一样，如果在第一队列中执行一个时间片段即可完成，便可获得与终端作业一样的响应时间。对于稍长作业，通常只需在第二和第三队列各执行一时间片即可完成，其周转时间仍然较短。

（3）长批处理作业，它将依次在第 1, 2, ..., n 个队列中运行，然后再按轮转方式运行，用户不必担心其作业长期得不到处理。所以，多级反馈队列调度算法能满足多用户需求。

14. 为什么在实时系统中，要求系统（尤其是 CPU）具有较强的处理能力？

答：实时系统中通常有着多个实时任务。若处理机的处理能力不够强，有可能因为处理机忙不过来而使某些实时任务得不到及时处理，导致发生难以预料的后果。

15. 按照调度方式可将实时调度算法分为哪几种？

答：可分为非抢占式和抢占式两种算法。而非抢占式算法又分为非抢占式轮转和优先调度算法；抢占式调度算法又分为基于时钟中断的抢占式优先权和立即抢占式优先权调度算法。

16. 什么是最早截止时间优先调度算法？举例说明。

答：根据任务的开始截止时间确定的任务优先级调度算法。截止时间越早则优先级越高。该算法要求在系统中保持一个实时任务就绪队列，该队列按各任务截止时间的先后排序。

举例：非抢占式调度方式用于非周期实时任务。图 3-9 是将该算法用于非抢占调度方式之例。该例中具有四个非周期任务，它们先后到达。系统首先调度任务 1 执行，在任务 1 执行期间，任务 2、3 又先后到达。由于任务 3 的开始截止时间早于任务 2，故系统在任务 1 后将

调度任务 3 执行。在此期间又到达作业 4，其开始截止时间仍是早于任务 2 的，故在任务 3 执行完后，系统又调度任务 4 执行，最后才调度任务 2 执行。

图 3-9 EDF 算法用于非抢占调度的调度方式

17. 什么是最低松弛度优先调度算法？举例说明之。

答：该算法是根据任务紧急(或松弛)的程度，来确定任务的优先级。任务的紧急程度愈高，为该任务所赋予的优先级就愈高，以使之优先执行。例如，一个任务在 200 ms 时必须完成，而它本身所需的运行时间就有 100 ms，因此，调度程序必须在 100 ms 之前调度执行，该任务的紧急程度(松弛程度)为 100 ms。又如，另一任务在 400 ms 时必须完成，它本身需要运行 150 ms，则其松弛程度为 250 ms。

18. 何谓死锁？产生死锁的原因和必要条件是什么？

答：死锁是指多个进程在运行过程中因争夺资源而造成的一种僵局，当进程处于这种僵持状态时，若无外力作用，它们都将无法再向前推进。

产生死锁的原因为竞争资源和进程间推进顺序非法。其必要条件是：互斥条件、请求和保持条件、不剥夺条件、环路等待条件。

19. 在解决死锁问题的几个方法中，哪种方法最易于实现？哪种方法使资源利用率最高？

答：解决死锁的四种方法即预防、避免、检测和解除死锁中，预防死锁最容易实现；避免死锁使资源的利用率最高。

20. 请详细说明可通过哪些途径预防死锁。

答：（1）摒弃“请求和保持”条件，就是如果系统有足够资源，便一次性把进程需要的所有资源分配给它；

（2）摒弃“不剥夺”条件，就是已经拥有资源的进程，当它提出新资源请求而不能立即满足时，必须释放它已保持的所有资源，待以后需要时再重新申请；

（3）摒弃“环路等待”条件，就是将所有资源按类型排序标号，所有进程对资源的请求必须严格按序号递增的次序提出。

21. 在银行家算法的例子中，如果 P0 发出请求向量由 Request(0,2,0)改为 Request(0,1,0)，问系统可否将资源分配给它？(此答案有点问题，需重新考虑)

答：（1）可以。银行家算法各种资源数量分别为 10、5、7，在 T0 时刻的资源分配如图所示：

（2）具体分析如下：

① $Request_0(0,1,0) \leq Need_0(7,4,3)$;

② $Request_0(0,1,0) \leq Available(2,3,0)$;

系统先假定可为 P0 分配资源，并修改 $Available_0$ ， $Allocation_0$ 和 $Need_0$ 向量，由此形成的资源变化情况如下图所示：

（3）P0 请求资源：P0 发出请求向量 $Request_0(0,1,0)$ ，系统按银行家算法进行检查：

① $Request_0(0,1,0) \leq Need_0(7,4,3)$;

② $Request_0(0,1,0) \leq Available(2,3,0)$;

③ 系统暂时先假定可为 P0 分配资源，并修改_____有关数据，如下图所示

综上所述系统可以将资源分配给它。

22. 银行家算法中出现以下资源分配，试问（1）该状态是否安全？（2）若进程 P2 提出 $Request(1,2,2,2)$ 后，系统能否将资源分配给它？

试问：（1）该状态是否安全？

（2）若进程 P2 提出请求 $Request(1,2,2,2)$ 后，系统能否将资源分配给它？(参考答案有错)

答：（1）安全，因为存在安全序列{P0,P3,P4,P1,P2}

（2）系统能分配资源，分析如下。

① $Request(1,2,2,2) \leq Need_2(2,3,5,6)$;

② $Request(1,2,2,2) \leq Available_2(1,3,5,4)$ 改成 $Available_2(1,6,2,2)$;

③系统先假定可为 P2 分配资源，并修改 $Available_2$ ， $Allocation_2$ 和 $Need_2$ 向量，

由此形成的资源变化情况如下图所示：

④ 再利用安全性算法检查此时系统是否安全。如下图

由此进行的安全性检查得知，可以找到一个安全序列{P2,P0,P1,P3,P4}。

第四章

1. 为什么要配置层次式存储器？

答：设置多个存储器可以使存储器两端的硬件能并行工作；采用多级存储系统，特别是 Cache 技术，是减轻存储器带宽对系统性能影响的最佳结构方案；在微处理机内部设置各种缓冲存储器，减轻对存储器存取的压力。增加 CPU 中寄存器数量大大缓解对存储器压力。

2. 可采用哪几种方式将程序装入内存？它们分别适用于何种场合？

答：（1）绝对装入方式，只适用于单道程序环境。

（2）可重定位装入方式，适用于多道程序环境。

（3）动态运行时装入方式，用于多道程序环境；不允许程序运行时在内存中移位置。

3. 何谓静态链接？何谓装入时动态链接和运行时的动态链接？ P120

答：静态链接是指在程序运行前，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配模块，以后不再拆开的链接方式。

装入时动态链接是指将用户源程序编译后得到的一组目标模块，在装入内存时采用边装入边链接的链接方式。

运行时动态链接是指对某些目标模块的链接，是在程序执行中需要该目标模块时，才对它进行的链接。

4. 在进行程序链接时，应完成哪些工作？

答：由链接程序 Linker 将编译后形成的一组目标模块，以及它们需要的库函数链接在一起，形成一个完整的装入模块 Load Module。主要工作是修改程序内的相对地址和修改目标程序中的外部调用标号。

5. 在动态分区分配方式中，应如何将各空闲分区链接成空闲分区链？

答：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”。

6. 为什么要引入动态重定位？如何实现？

答：在程序执行过程中，每当访问指令或数据时，将要访问的程序或数据的逻辑地址转换成物理地址，引入了动态重定位；

具体实现方法是在系统中增加一个重定位寄存器，用来装入程序在内存中的起始地址，程序执行时，真正访问的内存地址是相对地址与重定位寄存器中的地址相加之和，从而实现动态重定位。

7. 在采用首次适应算法回收内存时，可能出现哪几种情况？应怎样处理这些情况？

答：在采用首次适应算法回收内存时可能出现 4 种情况：

- （1）回收区前邻空闲区。将回收区与前邻空闲区合并，将前邻空闲区大小修改为两者之和。
- （2）回收区后邻空闲区。将两区合并，改后邻空闲区始址为回收区始址，大小为两者之和。
- （3）回收区前后均邻空闲区。将三个分区合并，修改前邻空闲区大小为三者之和。
- （4）回收区前后均不邻空闲区。为回收区设置空闲区表项，填入回收区始址和大小并插入空闲区队列。

8. 令 p 表示大小为 s 、地址为 x 的块的伙伴系统地址，试写出的通用表达式。

答：当 $s=1$ 时， $p=x$ ；当 $s=2$ 时， $p=x \oplus 1$ ；

9. 分区存储管理中常用那些分配策略？比较它们的优缺点。

答：分区存储管理中的常用分配策略：首次适应算法、循环首次适应算法、最佳适应算法、最坏适应算法。

首次适应算法优缺点：保留了高址部分的大空闲区，有利于后来的大型作业分配；低址部分不断被划分，留下许多难以利用的小空闲区，每次查找都从低址开始增加了系统开销。

循环首次适应算法优缺点：内存空闲分区分布均匀，减少了查找系统开销；缺乏大空闲分区，导致不能装入大型作业。

最佳适应算法优缺点：每次分配给文件的都是最适合该文件大小的分区，内存中留下许多难以利用的小空闲区。

最坏适应算法优缺点：剩下空闲区不太小，产生碎片几率小，对中小型文件分配分区操作有利；存储器中缺乏大空闲区，对大型文件分区分配不利。

10. 在系统中引入对换后可带来哪些好处？

答：交换技术将暂不需要的作业移到外存，让出内存空间以调入其它作业，交换到外存的作业也可以被再次调入。目的是解决内存紧张问题，带来的好处是进一步提高了内存利用率和

系统吞吐量。

11. 为实现对换，系统应具备哪几方面的功能？

答：系统应具备三方面功能：对换空间管理，进程换出，进程换入。

12. 在以进程为单位进行对换时，每次是否都将整个进程换出？为什么？

答：在以进程为单位进行对换时，并非每次都将整个进程换出。这是因为：

(1)从结构上讲，进程由程序段、数据段和进程控制块组成的，其中进程控制块总有部分或全部常驻内存，不被换出。

(2)程序段和数据段可能正被若干进程共享，此时它们也不能换出。

13. 为实现分页存储管理，需要哪些硬件的支持？

答：动态重定位技术、虚拟存储技术、多道程序设计技术。

14. 较详细的说明引入分段存储管理是为了满足用户哪几方面的需要。

答：

1) 方便编程。用户通常把自己的作业按照逻辑关系划分为若干段，每段都从 0 编址，并有自己名字和长度。因此，希望要访问的逻辑地址是由段名和段内偏移量决定。

2) 信息共享。在实现对程序和数据的共享时，是以信息逻辑单位为基础。分页系统中的页是存放信息的物理单位，无完整意义，不便于共享；段是信息的逻辑单位。为了实现段的共享，希望存储管理能与用户程序分段的组织方式相适应。

3) 信息保护。对信息的逻辑单位进行保护，分段能更有效方便地实现信息保护功能。

4) 动态增长。在实际应用中，有些段特别是数据段，在使用过程中会不断增长，事先又无法确切知道增长多少。分段存储管理方式能较好解决这个问题。

5) 动态链接。运行时先将主程序对应的目标程序装入内存并启动运行，运行过程中又需要调用某段时，才将该段调入内存链接。所以动态链接也要求以段作为管理单位。

15. 在具有快表的段页式存储管理方式中，如何实现地址变换？

答：在 CPU 给出有效地址后，由地址变换机构自动将页号 P 送入高速缓冲寄存器，并将此页号与高速缓存中的所有页号比较，若找到匹配页号，表示要访问的页表项在快表中。可直接从快表读出该页对应物理块号，送到物理地址寄存器中。如快表中没有对应页表项，则再访问内存页表，找到后，把从页表项中读出物理块号送地址寄存器；同时修改快表，将此页表项存入快表。但若寄存器已满，则 OS 必须找到合适的页表项换出。

16. 为什么说为什么说分段系统比分页系统更易于实现信息的共享和保护？

答：分页系统的每个页面是分散存储的，为了实现信息共享和保护，页面之间需要一一对应，为此需要建立大量的页表项；而分段系统的每个段都从 0 编址，并采用一段连续的地址空间，在实现共享和保护时，只需为要共享和保护的程序设置一个段表项，将其中的基址与内存地址一一对应就能够实现。

17. 分段和分页存储管理有何区别？

答：

（1）页是信息的物理单位，分页是为了实现离散分配方式，以消减内存的外部零头，提高内存利用率。段则是信息的逻辑单位，它含有一组相对完整的信息。

（2）页的大小固定且由系统决定，由系统把逻辑地址划分为页号和页内地址两部分，是由机械硬件实现的,因而在系统中只能有一种大小的的页面;而段的长度却不固定,决定于用户所编写的程序,通常由编译程序在对原程序进行编译时,根据信息的性质来划分。

（3）分页的作业地址空间是一维的,而分段作业地址空间则是二维的。

18. 试全面比较连续分配和离散分配方式.

答：

（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；

19. 虚拟存储器有哪些特征？其中最本质的特征是什么？

答：虚拟存储器有多次性、对换性、虚拟性三大特征。最本质的特征是虚拟性。

20. 实现虚拟存储器需要哪些硬件支持？

答：（1）请求分页（段）的页（段）表机制（2）缺页（段）中断机构（3）地址变换机构

21. 实现虚拟存储器需要哪几个关键技术？

答：

(1) 在分页请求系统中是在分页的基础上，增加了请求调页功能和页面置换功能所形成的页式虚拟存储系统。允许只装入少数页面的程序（及数据），便启动运行。

(2) 在请求分段系统中是在分段系统的基础上，增加了请求调段及分段置换功能后形成的段式虚拟存储系统。允许只装入少数段（而非所有段）的用户程序和数据，即可启动运行。

22. 在请求分页系统中，页表应包括哪些数据项？每项的作用是什么？

答：页表应包括：页号、物理块号、状态位 **P**、访问字段 **A**、修改位 **M** 和外存地址。

其中状态位 **P** 指示该页是否调入内存，供程序访问时参考；访问字段 **A** 用于记录本页在一段时间内被访问的次数，或最近已有多长时间未被访问，提供给置换算法选择换出页面时参考；修改位 **M** 表示该页在调入内存后是否被修改过；外存地址用于指出该页在外存上的地址，通常是物理块号，供调入该页时使用。

23. 在请求分页系统中，应从何处将所需页面调入内存？

答：请求分页系统中的缺页从何处调入内存分三种情况：

(1) 系统拥有足够对换区空间时，可以全部从对换区调入所需页面，提高调页速度。在进程运行前将与该进程有关的文件从文件区拷贝到对换区。

(2) 系统缺少足够对换区空间时，不被修改的文件直接从文件区调入；当换出这些页面时，未被修改的不必换出，再调入时，仍从文件区直接调入。对于可能修改的，在换出时便调到对换区，以后需要时再从对换区调入。

(3) UNIX 方式。未运行页面从文件区调入。曾经运行过但被换出页面，下次从对换区调入。UNIX 系统允许页面共享，某进程请求的页面有可能已调入内存，直接使用不再调入。

24. 在请求分页系统中，常采用哪几种页面置换算法？

答：采用的页面置换算法有：最佳置换算法和先进先出置换算法，最近最久未使用（LRU）置换算法，Clock 置换算法，最少使用置换算法，页面缓冲算法等。

25. 在请求分页系统中，通常采用哪种页面分配方式？为什么？

答：固定分配方式是基于进程的类型(交互型)或根据程序员、系统管理员的建议，为每个进程分配固定页数的内存空间，整个运行期间不再改变；采用可变分配方式有全局置换和局部置换两种，前者易于实现，后者效率高。

26. 在一个请求分页系统中，采用 LRU 页面置换算法时，假如一个作业的页面走向为 4, 3, 2, 1, 4, 3, 5, 4, 3, 2, 1, 5，当分配给该作业的物理块数 M 分别为 3 和 4 时，试计算访问过程中所发生的缺页次数和缺页率？比较所得结果？(参考答案有错)

答：当分配给该作业的物理块数 **M** 为 3 时，缺页 7 次，缺页率: $7/12=0.583$ ；

当分配给该作业的物理块数 M 为 4 时，缺页 4 次，缺页率: $4/12=0.333$.

-----以上解答出错。正确解法见下面：

答：当分配给该作业的物理块数 M 为 3 时，缺页 9 次，缺页率: $9/12=3/4$ ；

当分配给该作业的物理块数 M 为 4 时，缺页 10 次，缺页率: $10/12=5/6$.

27. 实现 LRU 算法所需的硬件支持是什么？

答：需要寄存器和栈等硬件支持。寄存器用于记录某进程在内存中各页的使用情况，栈用于保存当前使用的各个页面的页面号。

28. 试说明改进型 Clock 置换算法的基本原理.

答：因为修改过的页面在换出时付出的开销比未被修改过的页面大，在改进型 Clock 算法中，既考虑页面的使用情况，还要增加置换代价的因素；在选择页面作为淘汰页面时，把同时满足未使用过和未被修改作为首选淘汰页面。

29. 说明请求分段系统中的缺页中断处理过程。

答：请求分段系统中的缺页中断处理过程描述如下：

- (1) 根据当前执行指令中的逻辑地址查页表，判断该页是否在主存储器中
- (2) 该页标志为“0”形成缺页中断，中断装置通过交换 PSW 让操作系统的中断处理程序占用处理器。
- (3) 操作系统处理缺页中断处理的办法是查主存分配表找一个空闲的主存块，查页表找出该页在磁盘上位置，启动磁盘读出该页信息。
- (4) 把从磁盘上读出的信息装入找到的主存块中。
- (5) 当页面住处被装入主存后，应修改页表中对应的表目，填上该页所占用主存块把标志置为“1”，表示该页已在主存储器中
- (6) 由于产生缺页中断时的那条指令并没执行完，所以在把页面装入之后应重新执行被中断指令。

请求分段系统中的缺页中断处理过程如下图所示：

30. 如何实现分段共享？

答：在每个进程的段表中，用相应的表项指向共享段在内存中起始地址；配置相应的数据结构作为共享段表，在段表项中设置共享进程计数 **Count**，每调用一次该共享段，**Count** 值增 1，每当进程释放一个共享段时，**Count** 减 1，若减为 0，则系统回收该共享段的物理内存，取消在共享段表中该段对应的表项；共享段应给不同的进程以不同的存取权限；不同的进程可以使用不同的段号去共享该段。

第五章

1. 试说明设备控制器的组成。

答：由设备控制器与处理机的接口，设备控制器与设备的接口与 I/O 逻辑组成。

2. 为了实现 CPU 与设备控制器间的通信，设备控制器应具备哪些功能？

答：接收和识别命令；数据交换；标识和报告设备状态；地址识别；数据缓冲；差错控制。

3. 什么是字节多路通道？什么是数组选择通道和数组多路通道？

答：（1）字节多路通道。按字节交叉方式工作的通道。通常含有许多非分配型子通道，数量从几十到数百个，每个子通道连接一台 I/O 设备，控制其 I/O 操作。子通道按时间片轮转方式共享主通道。

（2）数组选择通道。按数组方式传送数据，传输速率很高，每次只允许一个设备数据。

（3）数组多路通道。将数组选择通道传输速率高和字节多路通道的各子通道分时并行操作的优点结合而成。含有多个非分配型子通道，具有很高的数据传输率和通道利用率。

4. 如何解决因通道不足而产生的瓶颈问题？

答：解决问题的有效方法是增加设备到主机间的通路而不增加通道，把一个设备连到多个控制器上，控制器又连到多个通道上，这种多通路方式解决了“瓶颈”问题，提高了系统可靠性，个别通道或控制器的故障不会使设备和存储器之间没有通路。

5. 试对 VESA 及 PCI 两种总线进行比较。

答：VESA 总线的设计思想是以低价占领市场。总线带宽 32 位，最高传输速率 132Mb/s。

广泛用于 486 微机。缺点是能连接的设备数仅为 2~4 台，控制器中无缓冲，难于适应处理器速度的提高，不支持 Pentium 机。

PCI 总线在 CPU 和外设间插入了复杂的管理层，协调数据传输和提供一致接口。管理层中配有数据缓冲，放大了线路的驱动能力，最多支持 10 种外设，支持高时钟频率的 CPU 运行，最大传输速率 132Mb/s。可连接 ISA、EISA 等传统总线，又支持 Pentium 的 64 位系统，是基于奔腾等新一代微处理器而发展的总线。

6. 试说明推动 I/O 控制发展的主要因素是什么？

答：推动 I/O 控制发展的主要动力在于尽量减少主机对 I/O 控制的干预，把主机从繁杂的 I/O 控制事务中解脱出来，用更多的时间和精力去完成其数据处理任务。同时，中断机制在计算机系统引入、DMA 控制器的出现和通道研制的成功使 I/O 控制的发展具备了技术支持和成为可能。

7. 有哪几种 I/O 控制方式？各适用于何种场合？

答：共有四种 I/O 控制方式。

(1)程序 I/O 方式：早期计算机无中断机构，处理机对 I/O 设备的控制采用程序 I/O 方式或称忙等的方式。

(2)中断驱动 I/O 控制方式：适用于有中断机构的计算机系统中。

(3)直接存储器访问（DMA）I/O 控制方式：适用于具有 DMA 控制器的计算机系统中。

(4)I/O 通道控制方式：具有通道程序的计算机系统中。

8. 试说明 DMA 的工作流程。

答：以从磁盘读入数据为例，说明 DMA 的工作流程。当 CPU 要从磁盘读入数据块时，先向磁盘控制器发送一条读命令。该命令被送到命令寄存器 CR 中。同时还发送本次要读入数据的内存起始目标地址，送入内存地址寄存器 MAR；本次要读数据的字节数送入数据计数器 DC，将磁盘中的源地址直接送 DMA 控制器的 I/O 控制逻辑上。然后启动 DMA 控制器传送数据，以后 CPU 便处理其它任务。整个数据传送过程由 DMA 控制器控制。下图为 DMA 方式的工作流程图。

9. 引入缓冲的主要原因是什么？

答：引入缓冲的主要原因是：

- (1)缓和 CPU 与 I/O 设备间速度不匹配的矛盾
- (2)减少对 CPU 的中断频率，放宽对中断响应时间的限制
- (3)提高 CPU 与 I/O 设备之间的并行性

10. 在单缓冲情况下，为什么系统对一块数据的处理时间为 $\max(C, T) + M$ ？

答：在块设备输入时，先从磁盘把一块数据输入到缓冲区，耗时为 T；然后由操作系统将缓冲区数据送给用户区，耗时 M；接下来由 CPU 对块数据进行计算，耗时 C。在单缓冲情况下，磁盘把数据输入到缓冲区的操作和 CPU 对数据的计算过程可以并行展开，所以系统对每一整块数据的处理时间为 $\max(C, T) + M$ 。

11. 为什么在双缓冲情况下，系统对一块数据的处理时间为 $\max(T, C)$ ？

答：写入者花费时间 T 将数据写满一个缓冲区后再写另一个缓冲区；读出者花费时间 M 将一个缓冲区数据送到用户区后再传送另一个缓冲区数据，计算者读出用户区数据进行处理。

由于将数据从缓冲区传到用户区操作必须与读用户区数据处理串行进行,而且可以与从外存传送数据填满缓冲区的操作并行。因此耗时大约为 $\max(C+M,T)$ 。考虑 M 是内存数据块的移动耗时非常短暂可以省略,因此近似地认为系统对一块数据处理时间为 $\max(C,T)$ 。

12. 试绘图说明把多缓冲用于输出时的情况。

答: 多缓冲用于输出的示意图如下:

13. 试说明收容输入工作缓冲区和提取输出工作缓冲区的工作情况。

答:

①收容输入工作缓冲区的工作情况为: 在输入进程需要输入数据时, 调用 `GetBuf(EmptyQueue)`过程, 从 `EmptyQueue` 队列的队首摘下一个空缓冲区, 作为收容输入工作缓冲区 `Hin`。然后把数据输入其中, 装满后再调用 `PutBuf(InputQueue,Hin)`过程, 将该缓冲区挂在输入队列 `InputQueue` 的队尾。

②提取输出工作缓冲区的工作情况为: 当要输出数据时, 调用 `GetBuf(OutputQueue)`过程, 从输出队列的队首取得一装满输出数据的缓冲区作为提取输出工作缓冲区 `Sout`。在数据提取完后, 再调用 `PutBuf(EmptyQueue,Sout)`过程, 将该缓冲区挂到空缓冲队列 `EmptyQueue` 的队尾。

14. 何谓安全分配方式和不安全分配方式?

答:

① 安全分配方式是指每当进程发出 I/O 请求后, 便进入阻塞状态, 直到其 I/O 操作完成时才被唤醒。在采用这种分配策略时, 一旦进程已获得某种设备资源后便阻塞, 使它不可能再请求任何资源, 而在它运行时又不保持任何资源。这种分配方式已经摒弃了造成死锁的“请求和保持”条件, 分配是安全的。缺点是进程进展缓慢, CPU 与 I/O 设备串行工作。

②不安全分配方式是指进程发出 I/O 请求后仍继续执行, 需要时又可发出第二个 I/O 请求、第三个 I/O 请求。仅当进程请求的设备已被另一个进程占有时, 进程才进入阻塞状态。优点是一个进程可同时操作多个设备, 进程推进迅速。缺点是分配不安全, 可能具有“请求和保持”条件, 可能造成死锁。因此, 在设备分配程序中需增加一个功能, 用于对本次的设备分配是否会发生死锁进行安全性计算, 仅当计算结果表明分配安全的情况下才进行分配。

15. 为何要引入设备独立性? 如何实现设备独立性?

答: 现代操作系统为了提高系统的可适应性和可扩展性, 都实现了设备独立性或设备无关性。基本含义是应用程序独立于具体使用的物理设备, 应用程序以逻辑设备名请求使用某类设

备。实现了设备独立性功能可带来两方面的好处：（1）设备分配时的灵活性；（2）易于实现 I/O 重定向。

为了实现设备的独立性，应引入逻辑设备和物理设备概念。在应用程序中，使用逻辑设备名请求使用某类设备；系统执行时是使用物理设备名。鉴于驱动程序是与硬件或设备紧密相关的软件，必须在驱动程序之上设置一层设备独立性软件，执行所有设备的公有操作、完成逻辑设备名到物理设备名的转换（为此应设置一张逻辑设备表）并向用户层（或文件层）软件提供统一接口，从而实现设备的独立性。

16. 在考虑到设备的独立性时，应如何分配独占设备？

答：在考虑到设备的独立性时，应按如下步骤来分配独占设备：

- (1) 进程以逻辑设备名提出 I/O 请求。
- (2) 根据逻辑设备表获得 I/O 请求的逻辑设备对应物理设备在系统设备表中的指针。
- (3) 检索系统设备表，找到属于请求类型、空闲可用且分配安全设备的设备控制表，将对应设备分配给请求进程；未找到则等待等待唤醒和分配。
- (4) 到设备控制表中找出与其相连接的控制器的控制器控制表，根据状态字段判断是否忙碌，忙则等待；否则将该控制器分配给进程。
- (5) 到该控制器的控制器控制表中找出与其相连接的通道的通道控制表，判断通道是否忙碌，忙则等待；否则将该通道分配给进程。
- (6) 只有在设备、控制器和通道三者都分配成功时，这次的设备分配才算成功，然后便可启动设备进行数据传送。

17. 何谓设备虚拟？实现设备虚拟时所依赖的关键技术是什么？

答：设备虚拟是指把独占设备经过某种技术处理改造成虚拟设备。

可虚拟设备是指一台物理设备在采用虚拟技术后，可变成多台逻辑上的虚拟设备，则可虚拟设备是可共享的设备，将它同时分配给多个进程使用，并对这些访问该物理设备的先后次序进行控制。

18. 试说明 SPOOLing 系统的组成。

答：SPOOLing 系统由输入井和输出井、输入缓冲区和输出缓冲区、输入进程 SPi 和输出进程 SPo 三部分组成。

19. 在实现后台打印时，SPOOLing 系统应为请求 I/O 的进程提供哪些服务？

答：在实现后台打印时，SPOOLing 系统应为请求 I/O 的进程提供以下服务：

- （1）由输出进程在输出井中申请一空闲盘块区，并将要打印的数据送入其中；
- （2）输出进程为用户进程申请空白用户打印表，填入打印要求，将该表挂到请求打印队列。

(3) 一旦打印机空闲, 输出进程便从请求打印队列的队首取出一张请求打印表, 根据表中要求将要打印的数据从输出井传送到内存缓冲区, 再由打印机进行打印。

20. 试说明设备驱动程序具有哪些特点。

答: 设备驱动程序具有如下特点:

- (1) 是请求 I/O 进程与设备控制器间的一个通信程序;
- (2) 驱动程序与 I/O 设备的特性紧密相关;
- (3) 驱动程序与 I/O 控制方式紧密相关;
- (4) 驱动程序与硬件紧密相关, 部分程序用汇编语言书写, 基本部分往往固化在 ROM 中。

21. 试说明设备驱动程序应具有哪些功能?

答: 设备驱动程序的主要功能包括:

- (1) 将接收到的抽象要求转为具体要求;
- (2) 检查用户 I/O 请求合法性, 了解 I/O 设备状态, 传递有关参数, 设置设备工作方式;
- (3) 发出 I/O 命令, 启动分配到的 I/O 设备, 完成指定 I/O 操作;
- (4) 及时响应由控制器或通道发来的中断请求, 根据中断类型调用相应中断处理程序处理;
- (5) 对于有通道的计算机, 驱动程序还应该根据用户 I/O 请求自动构成通道程序。

22. 设备中断处理程序通常需完成哪些工作?

答: 设备中断处理程序通常需完成如下工作:

- (1) 唤醒被阻塞的驱动程序进程;
- (2) 保护被中断进程的 CPU 环境;
- (3) 分析中断原因、转入相应的设备中断处理程序;
- (4) 进行中断处理;
- (5) 恢复被中断进程。

23. 磁盘访问时间由哪几部分组成? 每部分时间应如何计算?

答: 磁盘访问时间由寻道时间 T_s 、旋转延迟时间 T_r 、传输时间 T_t 三部分组成。

- (1) T_s 是启动磁臂时间 s 与磁头移动 n 条磁道的时间和, 即 $T_s = m \times n + s$ 。

(2) T_r 是指定扇区移动到磁头下面所经历的时间。硬盘 15000r/min 时 T_r 为 2ms;软盘 300 或 600r/min 时 T_r 为 50~100ms。

(3) T_t 是指数据从磁盘读出或向磁盘写入经历的时间。 T_t 的大小与每次读/写的字节数 b 和旋转速度有关: $T_t = b/rN$ 。

24.目前常用的磁盘调度算法有哪几种?每种算法优先考虑的问题是什么?

答: 目前常用的磁盘调度算法有先来先服务、最短寻道时间优先及扫描等算法。

(1) 先来先服务算法优先考虑进程请求访问磁盘的先后次序;

(2) 最短寻道时间优先算法优先考虑要求访问的磁道与当前磁头所在磁道距离是否最近;

(3) 扫描算法考虑欲访问的磁道与当前磁道间的距离,更优先考虑磁头当前的移动方向。

25.为什么要引入磁盘高速缓冲?何谓磁盘高速缓冲?

答: 目前磁盘的 I/O 速度远低于内存的访问速度,通常低上 4-6 个数量级。因此,磁盘 I/O 已成为计算机系统的瓶颈。为提高磁盘 I/O 的速度,便引入了磁盘高速缓冲。

磁盘高速缓冲是指利用内存中的存储空间,暂存从磁盘中读出的一系列盘块中的信息。

26.在设计磁盘高速缓冲时,如何实现数据交付?

答: 数据交付是指将磁盘高速缓存中的数据传给请求进程。当进程请求访问某个盘块中的数据时,由核心先查看磁盘高速缓冲,看其中是否存在所需盘块数据的拷贝。若有便直接从中提取数据交付给请求进程,避免了访盘操作,本次访问速度提高 4-6 个数量级;否则先从磁盘中将要访问的数据读入并交付给请求者进程,同时送高速缓存以便下次直接读取。

27.何谓提前读、延迟写和虚拟盘?

答: 提前读是指在读当前盘块的同时,将下一个可能要访问的盘块数据读入缓冲区,以便需要时直接从缓冲区中读取,无需启动磁盘。

延迟写是指在写盘块时,将对应缓冲区中的立即写数据暂时不立即写以备不久之后再被访问,只将它置上“延迟写”标志并挂到空闲缓冲队列的末尾。当移到空闲缓冲队首并被分配出去时,才写缓冲区中的数据。只要延迟写块仍在空闲缓冲队列中,任何要求访问都可直接从其中读出数据或将数据写入其中,而不必去访问磁盘。

虚拟盘又称 RAM 盘,是利用内存空间仿真磁盘。其设备驱动程序可以接受所有标准的磁盘操作,但这些操作不是在磁盘上而是在内存中,因此速度更快。

28.廉价磁盘冗余阵列是如何提高对磁盘的访问速度和可靠性的?

答: 廉价磁盘冗余阵列 RAID 是利用一台磁盘阵列控制器,统一管理和控制一组(几台到几十台)磁盘驱动器,组成高度可靠快速大容量的磁盘系统。

操作系统将 RAID 中的一组物理磁盘驱动器看作一个单个的逻辑磁盘驱动器。用户数据和系统数据可分布在阵列的所有磁盘中，并采取并行传输方式，大大减少数据传输时间和提高了可靠性。

第六章

1. 何谓数据项、记录 and 文件？

答：①数据项分为基本数据项和组合数据项。基本数据项描述一个对象某种属性的字符集，具有数据名、数据类型及数据值三个特性。组合数据项由若干数据项构成。

②记录是一组相关数据项的集合，用于描述一个对象某方面的属性。

③文件是具有文件名的一组相关信息的集合。

2. 文件系统的模型可分为三层，试说明其每一层所包含的基本内容。

答：第一层：对象及其属性说明（文件、目录、硬盘或磁带存储空间）；

第二层：对对象操纵和管理的软件集合（I/O 控制层即设备驱动程序、基本文件系统即物理 I/O 层、基本 I/O 管理程序或文件组织模块层、逻辑文件系统层）

第三层：文件系统接口（命令接口/图形化用户接口与程序接口）。

3. 试说明用户可以对文件施加的主要操作有哪些？

答：用户通过文件系统提供的系统调用对文件实施操作。

（1）基本文件操作：创建、删除、读、写、截断、设置读/写位置等；

（2）文件打开和关闭操作：第一步通过检索文件目录找到指定文件属性及其在外存上位置；第二步对文件实施读写等相应操作。

（3）其他文件操作：一是文件属性操作；二是目录操作；三是文件共享与文件系统操作的系统调用实现等。

4. 何谓逻辑文件？何谓物理文件？

答：逻辑文件是物理文件中存储的数据的一种视图方式，不包含具体数据，仅包含物理文件中数据的索引。物理文件又称文件存储结构，是指文件在外存上的存储组织形式。

5. 如何提高对变长记录顺序文件的检索速度？

答：基本方法是为变长记录顺序文件建立一张索引表，以主文件中每条记录的长度及指向对应记录的指针（即该记录在逻辑地址空间的首址）作为相应表项的内容。由于索引表本身是一个定长记录的顺序文件，若将其按记录键排序，则实现了对主文件方便快捷的直接存取。如果文件较大，应通过建立分组多级索引以进一步提高检索效率。

6. 试说明对索引文件和索引顺序文件的检索方法。

答：①索引文件的检索，首先根据用户（程序）提供的关键字，利用折半查找法检索索引表，找到相应表项；再利用给出的指向记录指针值，访问对应记录。

②索引顺序文件的检索，首先利用用户（程序）提供的关键字及查找方法，检索索引表，找到该记录在记录组中的第一条记录表项，得到第一个记录在主文件中的位置；再利用顺序查找法查找主文件，找到所要求的记录。

7. 试从检索速度和存储费用两方面对索引文件和索引顺序文件进行比较。

答：索引文件的主文件每条记录配置一个索引项，存储开销 N ，检索到具有指定关键字的记录，平均查找 $N/2$ 条记录。对于索引顺序文件，每个记录分组配置一个索引项，存储开销为 N ，检索到具有指定关键字的记录，平均需要查找 $N/2$ 次。

8. 试说明顺序文件的结构及其优点。

答：第一种是串结构：各记录之间的顺序与关键字无关。第二种是顺序结构：指文件中的所有记录按关键字（词）排列。可以按关键词长短排序或英文字母顺序排序。

顺序文件的最佳应用场合是对诸记录进行批量存取时，存取效率最高；只有顺序文件才能存储在磁带上并有效工作。

9. 在链接式文件中常用哪种链接方式？为什么？

答：链接方式分为隐式链接和显式链接两种形式。隐式链接是在文件目录的每个目录项中，都含有指向链接文件第一个盘块和最后一个盘块的指针。显式链接则把用于链接文件各物理块的指针，显式地存放在内存的一张链接表中。

10. 在 MS-DOS 中有两个文件 A 和 B，A 占用 11、12、16 和 14 四个盘块；B 占用 13、18 和 20 三个盘块。试画出在文件 A 和 B 中各盘块间的链接情况及 FAT 的情况。

答：如下图所示。

11. NTFS 文件系统对文件采用什么样的物理结构？

答：在 NTFS 文件系统中，以簇作为磁盘空间分配和回收的基本单位。一个文件占若干个簇，一个簇只属于一个文件。

12. 假定一个文件系统的组织方式与 MS-DOS 相似，在 FAT 中可有 64K 个指针，磁盘的盘块大小为 512B，试问该文件系统能否指引一个 512MB 的磁盘？

解：512MB/512B=1M 个盘块，而每个盘块都应有一个指针来指示，所以应该有 1M 个指针，因此若有 64K 指针不能指引一个 512MB 的磁盘。

13. 为了快速访问，又易于更新，当数据为以下形式时，应选用何种文件组织方式。

(1)不经常更新，经常随机访问；(2)经常更新，经常按一定顺序访问；(3)经常更新，经常随机访问；

答：以上三种宜分别采用(1)顺序结构(2)索引顺序结构(3)索引结构的组织方式。

14. 在 UNIX 中, 如果一个盘块的大小为 1KB, 每个盘块号占 4 个字节, 即每块可放 256 个地址。请转换下列文件的字节偏移量为物理地址。

(1)9999; (2)18000; (3)420000

答: 首先将逻辑文件的字节偏移量转换为逻辑块号和块内偏移量, 就是将[字节偏移量]/[盘块大小], 商为逻辑块号, 余数是块内偏移量。在 FCB 中, 第 0-9 个地址为直接地址, 第 10 个为一次间接地址, 第 11 个地址为二次间接地址, 第 12 个地址为三次间接地址。

再将文件的逻辑块号转换为物理块号。使用多重索引结构, 在索引节点中根据逻辑块号通过直接索引或间接索引找到对应的物理块号。

(1) $9999/1024=9$ 余 783, 则逻辑块号为 9, 直接索引第 9 个地址得到物理块号, 块内偏移地址为 783。

(2) $18000/1024=17$ 余 592, 则逻辑块号为 10 $<17<10+256$, 通过一次间接索引在第 10 个地址可得到物理块号, 块内偏移地址为 592。

(3) $420000/1024=410$ 余 160, 则逻辑块号为 10+256 <410 , 通过二次间接索引在第 11 个地址可得到一次间址, 再由此得到二次间址, 再找到物理块号, 其块内偏移地址 160。

15. 什么是索引文件? 为什么要引入多级索引?

答: 索引文件是指当记录为可变长度时, 通常为之建立一张索引表, 并为每个记录设置一个表项构成的文件。通常将索引非顺序文件简称为索引文件。索引是为了是用户的访问速度更快, 多级索引结构可以有效的管理索引文件, 可根据用户的访问情况多级处理。

16. 试说明 UNIX 系统中所采用的混合索引分配方式。

答: 混合索引分配方式是指将多种索引分配方式结合而成的分配方式。常见的是采用直接地址和一级索引联合的分配方式, 或两级索引分配方式, 甚至三级索引分配方式。在 UNIX System V 和 BSD UNIX 的索引结点中, 都设置了 13 个地址项, 即 iaddr(0)~iaddr(12), 把所有地址项分成直接地址和间接地址。

17. 对目录管理的主要要求是什么?

答: 实现按名存取、提高检索目录的速度、文件共享、允许文件重名。

18. 采用单级目录能否满足对目录管理的主要要求? 为什么?

答: 不能。单级目录在整个文件系统中只建立一张目录表, 每个文件占一个目录项, 其中含文件名、文件扩展名、文件长度、文件类型、文件物理地址、状态位等其它文件属性。

单级只能实现目录管理的基本功能, 不能满足查找速度、允许重名和文件共享的要求。

19. 目前广泛应用的目录结构有哪些? 它有什么优点?

答: 现代操作系统都采用多级目录结构。基本特点是查询速度快、层次结构清晰、文件管理和保护易于实现。

20. Hash 检索法有何优点? 又有何局限性?

答：Hash 检索法是系统把用户提供的文件名变换为文件目录的索引值，再利用该值查找目录，有效提高目录的检索速度，但 Hash 检索法局限于非通配符文件名。

21. 在 Hash 检索法中，如何解决“冲突”问题？

答：在 Hash 法查找目录时，如果目录表中相应目录项为空，表示系统中无指定文件。如果文件名与指定文件名匹配，表示找到了目标文件，也就找到了文件的物理地址。如果目录表中找到的相应文件名不匹配，则发生了冲突，需要 Hash 转换形成新的索引值，返回第一步重新查找。

22. 试说明在树型目录结构中线性检索法的检索过程，并给出相应的流程图。

答：在树型结构的目录中，当有两个或多个用户共享一个子目录或文件时，将共享文件或字母路连接到两个或多个用户目录中，方便找到该文件。此时目录结构不再是树形结构，而是有个有向非循环图 DGA。

23. 有一计算机系统利用图 6-33 所示的位示图来管理空闲盘块。盘块的大小为 1KB，现要为某文件分配 5 个盘块，试说明盘块的具体分配过程。

答：分配 5 个盘块的过程如下：

(1) 顺序扫描位示图，从中找到第一个值为 0 的二进制位，得到行号 $i=3$ ，列号 $j=3$ 。

(2) 将找到的二进制位转换成对应盘块号。盘块号为： $b = (3-1) * 16 + 3 = 35$ ；

(3) 修改位示图，令 $map[3, 3]=1$ ，并将该盘块分配出去。

类似地，可使用相同的方法找到第二个值为 0 的二进制位，得到行号 $i=4$ ，列号 $j=7$ ，其对应的盘块号为 55，令 $map[i, j]=1$ ，并将该盘块分配出去。

24. 某操作系统磁盘文件空间共 500 块，若用字长为 32 位的位示图管理磁盘空间，试问：

(1) 位示图需要多少字？

(2) 第 i 字第 j 位对应的块号是多少？

(3) 给出申请/归还一块的工作流程。

答：(1) 位示图需要的字数计算： $INT(500/32) = 16$ 个字。

(2) 块号 $b = (i-1) * 32 + j$

(3) 申请的过程：顺序扫描位示图、找到空闲块并分配、修改位示图 $map[i, j]=1$ 。

归还的过程：找到回收盘块在位示图中的行和列，修改位示图 $map[i, j]=0$ 。

25. 对空闲磁盘空间的管理常采用哪几种分配方式？在 UNIX 系统中采用何种分配方式？

答：空闲表法、空闲链表法、位示图法、成组链接法。UNIX 系统采用的是成组链接法

26. 基于索引节点的文件共享方式有何优点？

答：优点是建立新的共享链接时，不改变文件拥有者关系，仅把索引结点共享计数器加 1，系统可获悉了由多少个目录项指向该文件。缺点是拥有者不能删除自己的文件，否则会出错。

27. 基于符号链的文件共享方式有何优点？

答：能够通过网络链接世界上任何地方的计算机中的文件。

28. 在一级系统容错技术中，包括哪些容错措施？什么是写后读校验？

答：一级系统容错技术包括双份目录、双份文件分配表及写后读校验等容错措施。

写后读校验是每次从内存缓冲区向磁盘写入一个数据块后，又立即从磁盘上读出该数据块，并送至另一缓冲区中，再将该缓冲区内容与内存缓冲区中在写后仍保留的数据进行比较。若两者一致，才认为写入成功，继续写下一个盘块。否则重写。若重写后仍不一致，则认为盘块缺陷，便将应写入该盘块的数据，写入到热修复重定向区中。

29. 在第二级系统容错技术中，包括哪些容错措施？画图说明之。

答：第二级容错技术包括磁盘镜像和磁盘双工两种容错措施。图示如下：

30. 何谓事务？如何保证事务的原子性？

答：事务是用于访问和修改各种数据项的一个程序单位。

要保证事务的原子性必须要求一个事务在对一批数据执行修改操作时，要么全部完成，用修改后的数据代替原来数据，要么一个也不改，保持原来数据的一致性。

31. 引入检查点的目的是什么？引入检查点后又如何进行恢复处理？

答：引入检查点的目的是使对事务记录表中事务记录的清理工作经常化。

恢复处理由恢复例程来实现。首先查找事务记录表，确定在最近检查点以前开始执行的最后的事务 T_i 。找到 T_i 后再返回搜索事务记录表，找到第一个检查点记录，从该检查点开始，返回搜索各个事务记录，利用 redo 和 undo 过程对他们进行相应的处理。

32. 为何引入共享锁？如何用互斥锁或共享锁来实现事务的顺序性？

答：引入共享锁是为了提高运行效率。在给对象设置了互斥锁和共享锁的情况下，如果事务 T_i 要对 Q 执行读操作，只需获得 Q 的共享锁。如果对象 Q 已被互斥锁锁住，则 T_i 必须等待；否则便获得共享锁对 Q 执行读操作。如果 T_i 要对 Q 执行写操作，则 T_i 还要获得 Q 的互斥锁。若失败则等待；成功则获得互斥锁并对 Q 执行写操作。

33. 当系统中有重复文件时，如何保证他们的一致性？

答：可以采用两种方法：一是对所有的重复文件进行同样的修改，二是用新修改的文件替换所有的重复文件。

34. 如何检索盘块号的一致性？检查时可能出现哪几种情况？

答：为了保证盘块号的一致性，先将计数器表中的所有表项初始化为 0，用 N 个空闲盘块号计数器组成的第一组计数器对从空闲盘块表中读出的盘块号计数，用 N 个数据盘块号计数器组成的第二组计数器对从文件分配表中读出的已分配给文件使用的盘快号计数。如果两组计数中的对应数据互补则数据一致，反之则发生错误。

检查时可能出现的情况：

- （1）两组计数器中盘块 K 的计数值均为 0，应在空闲盘块表中增加盘块号 K；
- （2）空闲盘块号计数器中盘块 K 的计数值为 2，应删除一个空闲盘块号 K；
- （3）空闲盘块号计数器中盘块号 K 的计数值为 0，而数据盘块号计数器中盘块号 K 的计数值大于 1，则错误严重，存在数据丢失等事件，必须立即报告系统加以处理。

第七章

1. 操作系统包括哪几种类型的用户接口？它们分别适用于哪种情况？

答：操作系统包括四种类型的用户接口：命令接口（分为联机与脱机命令接口）、程序接口、图形化用户接口和网络用户接口。

命令接口和图形化用户接口支持用户直接通过终端来使用计算机系统，程序接口提供给用户在编制程序时使用，网络用户接口是面向网络应用的接口。

2. 联机命令接口由哪些部分构成？

答：联机命令接口由一组联机命令、终端处理程序和命令解释程序构成。

3. 联机命令通常包含哪些类型？每种类型又包含哪些主要命令？

答：联机命令通常包含如下类型：

- （1）系统访问类，主要是注册命令 login、password；
- （2）磁盘操作类，包括磁盘格式化 format、软盘复制 diskcopy、软盘比较 diskcomp 及备份 backup 等命令；
- （3）文件操作类，包括文件显示 type、文件拷贝 copy、文件比较 comp、文件重命名 rename、文件删除 erase 等命令；

(4) 目录操作类, 包括子目录建立 **mkdir**、目录显示 **dir**、子目录删除 **rmdir**、目录结构显示 **tree**、当前目录改变 **chdir** 等命令;

(5) 其它命令, 包括输入输出重定向>、<,管道联接|、过滤命令/,批命令**.bat** 等。

4. 什么是输入输出重定向? 试举例说明。

答: 命令的输入通常取自标准输入设备键盘, 命令输出则送往标准输出设备显示终端。如果在命令中设置输出定向“>”, 其后接文件或设备名, 则命令的结果输出将送到指定文件或设备上; 如果使用输入重定向“<”, 则不从键盘而是从重定向符右边的指定文件或设备上取得输入信息。这就是输入输出的重定向。

5. 何谓管道联接? 试举例说明。

答: 管道联接是指把第一个命令的输出作为第二个命令的输入, 或把第二个命令的输出作为第三条命令的输入, 以此类推, 由两条以上的命令可形成一条管道。在 **MS-DOS** 和 **UNIX** 中, 都用“|”作为管道符号。其一般格式为: **command1 |command2 | ...|commandn**。

6. 终端设备处理程序的主要作用是什么? 它具有哪些功能?

答: 主要用于实现人机交互, 它具有以下功能:

(1) 接收用户从终端上键入的字符; (2) 字符缓冲, 用于暂存接收的字符; (3) 回送显示; (4) 屏幕编辑; (5) 特殊字符处理。

7. 命令解释程序的主要作用是什么?

答: 主要作用是: 在屏幕上给出提示符, 请用户输入命令, 读入并识别命令, 转至相应命令处理程序入口地址, 把控制权交给该处理程序执行, 最后将处理结果或出错信息送屏幕显示。

8. 试说明 **MS-DOS** 的命令处理程序 **COMMAND.COM** 的工作流程。

答: **COMMAND.COM** 的工作流程如下:

(1) 系统通电或复位后, 初始化程序对整个系统完成初始化工作, 自动执行 **Autoexec.bat** 文件, 然后把控制权交给暂存部分, 给出提示符等待接收用户键入命令;

(2) 暂存部分读入键盘缓冲区中的命令, 判别文件名、扩展名及驱动器名是否正确, 若有错则出错返回, 正确则查找和识别命令;

(3) 若为内部命令, 暂存部分找到后便从对应表项中获得命令处理程序的入口地址并执行; 若为外部指令, 则建立命令行, 执行系统调用 **exec** 装入其命令处理程序, 得到对应基地址并执行; 若键入命令非法, 则出错返回;

(4) 命令完成后, 控制权重新交给暂存部分给出提示符等待接收用户命令, 转 (2)。

9. 为了将已存文件改名, 应用什么 **UNIX** 命令?

答: 对已存文件改名的命令为 **mv**, 其格式为: **mv oldname newname**。

10. 要想将工作目录移到目录树的某指定点上，应利用什么命令？

答：用命令“cd ..”或“cd .”将当前目录上移或转移，直到工作目录移到目录树的指定点。

11.如果希望把 file1 的内容附加到原有的文件 file2 的末尾，应用什么指令？

答：\$catfile1>>file4

12.试比较 mail 和 write 命令的作用有何不同？

答：mail 命令作为 UNIX 的多用户之间非交互式通信的工具。write 命令是用户与当前系统中的其他用户直接进行联机通讯的工具。

13. 试比较一般的过程调用和系统调用？

答：系统调用本质上是过程调用的一种特殊形式，与一般过程调用有差别：

（1）运行状态不同。一般过程调用的调用过程和被调用过程均为用户程序，或者均为系统程序，运行在同一系统状态（用户态或系统态）；系统调用的调用过程是用户态下的用户程序，被调用过程是系统态下的系统程序。

（2）软中断进入机制。一般的过程调用可直接由调用过程转向被调用过程；而系统调用不允许由调用过程直接转向被调用过程，一般通过软中断机制，先进入操作系统内核，经内核分析后，才能转向相应命令处理程序。

（3）返回及重新调度。一般过程调用在被调用结束后，返回调用点继续执行；系统调用被调用完后，要对系统中所有运行进程重新调度。只有当调用进程仍具有最高优先权才返回调用过程继续执行。

（4）嵌套调用。一般过程和系统调用都允许嵌套调用，注意系统过程嵌套而非用户过程。

14. 什么是系统调用？它都有哪些类型？

答：系统调用是指在操作系统内核设置的一组用于实现各种系统功能的子程序或过程，并提供给用户程序调用。主要类型包括：

（1）进程控制类。用于进程创建、终止、等待、替换、进程数据段大小改变及进程标识符或指定进程属性获得等；

（2）文件操纵类。用于文件创建、打开、关闭、读/写及文件读写指针移动和属性修改，目录创建及索引结点建立等；

（3）进程通信类，用于实现通信机制如消息传递、共享存储区及信息量集机制等；

（4）信息维护类，用于实现日期、时间及系统相关信息设置和获得。

15. 如何设置系统调用所需的参数？

答：设置系统调用参数包括两种方式：

（1）直接将参数送入相应的寄存器中。问题是寄存器有限，限制了设置参数的数目。

(2) 参数表方式。将系统调用所需参数，放入参数表中，再将该表指针放在寄存器中。

16. 试说明系统调用的处理步骤。

答：(1) 设置系统调用号和参数。

(2) 对系统调用命令的一般性处理。如保护 CPU 现场，PSW、PC、系统调用号、用户栈指针及通用寄存器入栈，保存用户定义参数等。UNIX 中执行 CHMK 命令将参数表中的参数传到 User 结构的 U.U-arg() 中；MS-DOS 执行 INT21 软中断。

(3) 根据系统调用入口表及具体系统调用命令转至对应命令处理程序具体处理。

17. 为什么在访问文件前，要用 open 系统调用先打开该文件？

答：系统将在用户与该文件之间建立一条快捷通路。在文件被打开后，系统将给用户返回一个该文件的句柄或描述符。

18. 在 UNIX 系统中是否设置了专门用来删除文件的系统调用？为什么？

答：没有。当用户不再使用此文件时，利用系统调用 unlink 断开连接，做 i.link 减 1 操作。

当 i.link 减 1 结果为 0，表示此文件不再被用户需要，该文件将从文件系统中删除。

19. 在 IPC 软件包中包含哪几种通信机制？在每种通信机制中设置了那些系统调用？

答：IPC 中提供了三种通信机制：

(1) 消息机制。存在 msgget, msgsend, msgrcv 系统调用。

(2) 共享存储器机制。存在 shmget, shmid, shmdt 系统调用。

(3) 信号量机制。无系统调用。

20. trap.S 是什么程序？它完成哪些主要功能？

答：UNIX 系统 V 内核中的 trap.S 文件，是中断和陷入总控程序。用于中断和陷入的一般性处理，由汇编语言编写。trap.S 中包含了绝大部分的中断和陷入向量的入口地址，每当系统发生中断和陷入时，通常都是直接进入 trap.S 程序。

21. 在 UNIX 系统内，被保护的 CPU 环境中包含哪些数据项？

答：当用户程序处在用户态且执行 CHMK (CHange Mode toKernel) 之前，应在用户空间提供系统调用所需参数表，并将参数表地址送入 R0 寄存器。执行 CHMK 命令后，处理机转为核心态，由硬件自动将处理机状态长字(PSL)、PC 和代码操作数(code)压入用户核心栈，从中断和陷入向量表中取出 trap.S 入口地址，转入 trap.S 中执行。

trap.S 程序执行后，将陷入类型 type 和用户栈指针 usp 压入用户核心栈，将被中断进程的 CPU 环境中的一系列寄存器如 R0~R11 的部分或全部压栈。哪些寄存器内容压栈，取决于特定寄存器的屏蔽码，屏蔽码每一位都与 R0~R11 中的寄存器相对应。当某一位置 1 时，表示对应寄存器内容压栈。

22. trap.C 是什么程序？它将完成哪些处理？

答：trap.C 程序是处理各种陷入情况的 C 语言文件，共处理 12 种陷入后的公共问题。包括：确定系统调用号、实现参数传递、转入相应系统调用处理子程序。在由系统调用子程序返回到 trap.C 后，重新计算进程的优先级，对收到的信号进行处理等。

23. 为方便转入系统调用处理程序，在 UNIX 系统中配置了什么样的数据结构？

答：在 UNIX 系统中配置了系统调用定义表 `sysent[]`，在表的每个结构中包含三个元素，一是相应系统调用所需参数的个数；二是系统调用经寄存器传送的参数个数；三是相应系统调用处理子程序的入口地址。系统中设置了该表之后，可根据系统调用号 `i` 从系统调用定义表中找出相应表目，按表目中的入口地址转入相应处理子程序，完成系统调用的特定功能。执行完后，返回中断和陷入总控程序中的 trap.C 程序中，返回断点前的公共处理部分。

第八章网络操作系统

第八章网络操作系统

第八章

1. 按网络拓扑结构可以把计算机网络分为哪几类？试画出它们的网络拓扑图。

答：计算机网络可分为星形、环形、总线形、树形和网状形网络。它们的网络拓扑图如下：

2. 试说明分组交换网的组成。

答：由分组交换机、网路管理中心、远程集中器、分组装拆设备以及传输设备等组成。

3. 何谓帧交换方式及信元交换方式？

答：帧交换方式是在传统分组交换的基础上发展起来的，传输基本单位是帧，长度可变，采用“存储转发”方式，即帧交换机每接到一个新帧时，都将该帧送帧缓冲区排队，按照该帧中的目标地址，将该帧转发给相应路径的下一个帧交换机。

信元交换方式是改进了的帧中继交换方式。当源帧交换机收到用户设备发来的帧，便分割为多个定长信元，在整个帧中继器网络中传输和交换时，都以信元为基本单位，到达目标帧交换机后，被重组为帧。

4. 局域网可分为基本型和快速型两大类，每一类中包括哪几种局域网？

答：基本型局域网有：（1）以太网（2）令牌环网

快速局域网有：（1）FDDI 光纤环网（2）快速以太网 100 BASE-T。

5. 为实现同构 LAN 网络互连，应采用什么样的网络互连设备？应具有哪些功能？

答：同构 LAN 网络互连设备与功能：

(1) 网桥。功能：帧的发送和接受、缓冲处理、协议转换。

(2) 路由器。功能：拆包和打包、路由选择、协议转换、分段和重组

6. 为实现异构型网络互连，应采用什么样的网络互联设备？它又应具有哪些功能？

答：采用网关。实现异构 LAN 互连、LAN 与 WAN 互连、WAN 互连、LAN 与主机互连。

7. 网络层向传输层提供了哪两类数据传输服务？试对它们做简要的说明。

答：(1) 数据包服务。发方网络层从传输层接收报文，为它配上完整的目标地址，作为独立信息单位传送出去。数据包每经过一个中继节点都根据当时当地情况，按一定算法选择一条最佳传输路径转发出去。采用数据包服务的收、发双发无需建立连接。

(2) 虚电路服务。通信前由源主机发送呼叫报文分组，包含源和目标主机的全网地址。

目标主机同意通信，便由网络层在双方间建立一条虚电路。在以后通信中只需填上虚电路的逻辑信道号；通信结束拆除该虚电路。

8. 传输层所起的桥梁作用具体表现在哪几方面？

答：(1) 传输出错率和建立连接的失败率。(2) 数据传输速率、吞吐量和传输时延。

(3) 分段和组段功能。

9. TCP/IP 模型中包含了哪几个层次？简要说明每个层次的主要功能。

答：TCP/IP 模型中包含 4 个层次。

(1) 应用层。对应于 OSI 高层，为用户提供需要的服务。如 FTP、Telnet、DNS 等。

(2) 传输层。对应于 OSI 传输层，为应用层实体提供端到端的通信功能。定义了面向连接的 TCP 和无连接的用户数据报协议 UDP 这两个主要协议。

(3) 网络互联层。对应于 OSI 网络层，解决主机到主机的通信问题。有网际协议 IP、地址解析协议 ARP、互联网组管理协议 IGMP 和互联网控制报文协议 ICMP 四个主要协议。

(4) 网络访问层。对应 OSI 的物理层和数据链路层。

10. 网络互联层 IP 协议的主要作用是什么？为什么在有了 IP 协议之后还要配置 TCP 协议？

答：(1) IP 协议主要用于异构网络间的相互连接和路由选择。IP 提供的是不可靠、面向无

连接的数据报传递服务。

(2) TCP 协议提供面向连接、可靠的端端通信机制。TCP 比 IP 可以确保数据传输的可靠性，即使网络层出错，TCP 仍能正确控制建立连接、数据传输和连接释放。

11.试说明在介质访问控制 MAC 子层中, IEEE 802.2、IEEE802.3、IEEE 802.3u、IEEE 802.2z、IEEE 802.5、IEEE802.6 都是些什么标准?

答: IEEE 802.2 是逻辑链路控制的标准。 IEEE 802.3 是以太网的标准。

IEEE 802.3u 是以太网的标准。 IEEE 802.2z 是以太网的标准。

IEEE 802.5 是令牌环的标准。 IEEE 802.6 是城域网的标准。

12.何谓网络体系结构? OSI/RM 由哪几部分组成?

答: 网络体系结构是指通信系统的整体设计, 为网络硬件、软件、协议、存取控制和拓扑提供标准。OSI/RM 从低到高分七层: 物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

13.什么是网络协议? 扼要说明其所含的三要素。

答: 网络协议是为计算机网络中进行数据交换而建立的规则、标准或约定的集合。

计算机网络协议主要由语义、语法和交换规则三部分即协议三要素组成。

语义: 规定通信双方彼此讲什么, 确定协议元素类型, 如规定通信双方发什么控制信息, 执行的动作和返回的应答。

语法: 规定通信双方彼此如何讲, 确定协议元素格式, 如数据和控制信息的格式。

交换规则: 规定信息交流的次序。

14.ISO 将 OSI/RM 分成几层? 各层的主要用途是什么?

答: OSI/RM 分 7 层。各层的主要用途是:

物理层: 规定了网络设备间物理连接标准, 在网络设备间透明地传输比特流。

数据链路层: 提供相邻节点间可靠的数据传输功能。

网络层: 在通信子网中进行路由选择和通信控制。

传输层: 提供两个端系统间的可靠通信。

会话层: 建立和控制两个应用实体间的会话过程。

表示层: 提供统一的网络数据表示。

应用层: 提供网络用户间的分布式应用环境(普通用户)和应用开发环境(网络程序员)。

15.客户/服务器模式得以广泛流行的主要因素是什么?

答: (1) 模块化与应用的分布特性 (2) 充分利用资源, 提高网络效率

(3) 便于系统维护, 扩充性强 (4) 并发特性

16.试说明客户与服务器之间的交互情况。

答：C/S 模式的两层结构系统是：第一层在客户机系统上结合表示与业务逻辑；第二层通过网络结合了数据库服务器。C/S 模式主要由客户应用程序、服务器管理程序和中间件三部分组成。

17.两层 C/S 模式有哪些局限性？如何解决？

答：（1）不能适应应用不断增多的情况。

（2）需要在客户机与服务器上安装特定的网络软件，实现 C/S 间的互用性。

（3）客户机直接与服务器交互。

解决办法：设法使 C 与提供数据等服务的 S 无关，在 C/S 间增设中间实体。

18.为什么在大型信息系统和 Internet 环境下，应采用三层客户/服务器模式？

答：因为 Internet 发展极为迅速，三层客户/服务器模式更适合发展。把客户机作为 Web 浏览器，从而形成了 Web 浏览器、Web 服务器和数据库服务器三层的 C/S 模式。

19.试比较两层和三层的 C/S 模式。

答：三层与两层模式相比的优点：（1）增加了系统的灵活性和可扩展性。

（2）简化了客户机，降低了系统费用。（3）使客户机安装、配置和维护更为方便。

三层的缺点：（1）软件开发难度大，开发周期长。（2）访问效率低。

20.现代计算机网络有哪些主要功能。

答：计算机网络的主要功能是数据通信和资源共享、系统容错、网络管理、应用互操作功能。

21.试说明在层次式结构的网络中进行数据通信时，信息的流动过程。

答：请求信息从客户机到应用服务器，再到数据服务器，然后数据服务器根据要求向应用服务器传送信息，再由应用服务器找到客户机。

22.为实现数据通信，计算机网络应有哪些具体功能？

答：连接的建立和拆除、报文的分解和组装、传输控制、流量控制、差错检测与纠正。

23.试说明当前实现文件和数据共享的两种主要方式。

答：以虚拟软盘方式和以文件服务方式实现的数据共享方式。

24.网络管理的主要目标是什么？

答：A. 增强网络的可用性。 B. 提高网络运行质量。 C. 提高网络资源利用率。

D. 保障网络的安全性 E. 提高网络和社会经济效益。

25.网络管理包括哪几方面的具体功能？

答：配置管理、故障管理、性能管理、安全管理、计费管理。

26.何谓信息“互通性”和信息“互用性”？

答：信息的互通性是指在不同网络结点间实现通信。目前主要利用 TCP/IP 实现信息互通。

信息的互用性是指在不同网络中的站点间实现信息的互用，即一个网络中的用户能访问另一个网络文件系统或数据库系统中的文件或数据。

27.何谓电子邮件？它可分为哪几种类型？

答：电子邮件 E-mail,标志@，又称电子信箱、电子邮政，是用电子手段提供信息交换的通信方式。电子邮件服务器分为两种类型，MIME 协议和 SMTP 协议。现代 E-mail 中可包含多种不同类型的文件，如文本、图像、音频和视频信息等。

28.文件传输的复杂性表现在哪几方面？如何解决？

答：异构网络下的文件传输，需要在 Internet 中建立了统一的文件传输协议 FTP。

（1）内部用户 FTP。只允许在文件服务器上拥有账户的用户使用 FTP 服务。

（2）匿名 FTP。在 Internet 上实现资源共享的重要手段，允许非注册用户拷贝文件。

29.试比较电子邮件服务和文件传输服务。

答：电子邮件服务借助于 E-mail 设施与世界上所有国家和地区的网络用户通信。

文件传输服务是在 Internet 中建立统一的文件传输协议 FTP，实现用户在不同主机间的文件拷贝功能。

30.网络环境下的目录服务有何特点？

答：规模小的局域网不需要提供目录服务，对于大型企业网必须对网络管理员和用户提供目录服务，发挥网络的应有作用。目录服务还应能对每台物理设备提供的网络服务进行管理。对服务器提供的网络服务可以是文件/打印服务、数据库服务等。

31. 目录服务包括哪些主要功能？

答：（1）用户管理。保证核准用户能方便地访问各种网络服务，禁止非法用户访问。

（2）分区和复制。将庞大目录库分成若干个分区，并分别复制到多台服务器，使每个分区被复制的位置尽量靠近最常使用这些对象的用户，有的目录服务还允许一台服务器上存放多个不同分区的拷贝。

（3）创建扩充和继承功能。创建是在目录中创建新的对象，并设置属性。扩充指对原

有目录服务功能的扩充。继承是指目录对象继承其他对象的属性和权力的能力。

(4) 多平台支持功能。由于目录服务存在着管理对象的差异, 要求具有跨越平台能力。

32. Internet 具有哪些特征?

答: (1) 广域性 (2) 广泛性 (3) 高速性 (4) 综合性

33.何谓 WWW? 它与一般的信息检索工具有何不同?

答: WWW (Word Wide Web) 称为万维网或 Web, 是当前最为流行的信息服务类型。

它与一般信息检索工具不同表现在: 一般检索工具每次只能从一台主机上查找需要的文件, 且文件数据类型单一; 而 Web 检索可以一次从多台主机中查找需要的数据, 允许类型各异, 并将这些数据形成一份文件。

34.何谓 BBS? 它何以会受到广大网络用户的欢迎?

答: BBS (Bulletin Board System) 即电子公告板。BBS 用户已经扩展到各行各业, BBS 可以交换各种文件。通过 BBS 系统可随时取得国际最新软件及信息, 可以和别人讨论计算机软件、硬件、Internet、多媒体、程序设计以及医学等各种有趣话题, 可以利用 BBS 刊登征友、廉价转让及公司产品等启事。只要拥有 1 台计算机和上网设备就能立刻进入“超时代”BBS 领域, 享用它无比的威力! 因此 BBS 受到了广大网络用户的欢迎。

35.什么是域名服务? Internet 的域名是由几段构成的?

答: 域名是 Internet 网络上的一个服务器或一个网络系统的名字。域名的形式是以若干个英文字母和数字组成, 由 "." 分隔成几部分, 如 cctv.com 就是一个域名。

一个完整的域名由两个或两个以上词段组成, 部分之间用英文句号 "." 分隔, 最后一个 "." 的右边部分称为顶级域名 (TLD) 或一级域名, 最后一个 "." 的左边部分称为二级域名 (SLD), 二级域名的左边部分称为三级域名, 以此类推, 每一级的域名控制它下一级域名的分配。

36.什么是域名解析? 最基本的一种域名解析方式是如何实现的?

答: 域名解析是将域名重新转换为对应 IP 地址的过程。一个域名只对应一个 IP 地址, 多个域名可以同时解析到一个 IP 地址。域名解析需要由专门的域名解析服务器 DNS 完成。

域名解析的过程: 当应用过程需要将一个主机域名映射为 IP 地址时, 就调用域名解析函数将待转换的域名放在 DNS 请求中, 以 UDP 报文方式发给本地域名服务器。查到域名后将对应 IP 地址放在应答报文中返回。若域名服务器不能回答该请求, 则此域名服务器向根域名服务器发出请求解析, 找到下面的所有二级域名服务器, 以此类推, 直到查询到所请求的

域名并赋 IP 值返回。

37.为能支持 Internet 所提供的服务，在操作系统中应配置哪些软件？

答：应配置 WEB 浏览器，如 IE、firefox、Chrome 等，特殊的服务可以根据需要安装对应的软件。

38.何谓浏览器/服务器模式？浏览器和服务器的基本功能是什么？

答：浏览器/服务器模式即 B/S 结构或 Browser/Server 结构。只安装维护一个服务器 Server，客户端采用浏览器 Browse 软件。利用成熟的 WWW 技术，结合多种 Script 语言 (VBScript、JavaScript...)和 ActiveX 技术，是全新的软件系统构造技术。

在 B / S 体系结构系统中，浏览器向分布在网络上的许多服务器发出请求，服务器对浏览器的请求进行处理，将用户所需信息返回到浏览器。而数据请求、加工、结果返回及动态网页生成、数据库访问和应用程序执行等工作全部由 Web Server 完成。随着 Windows 将浏览器技术植入操作系统内部，这种结构已成为当今应用软件的首选体系结构。

B/S 结构的主要特点是分布性广、维护方便、开发简单、共享性强、总体成本低。但数据安全性、服务器要求高、数据传输慢、软件个性化特点明显降低，难以实现传统模式下的特殊功能要求。

浏览器是指可以显示网页服务器或者文件系统的 HTML 文件内容，并让用户与这些文件交互的一种软件。服务器是网络上为客户端计算机提供各种服务的高可用性计算机。

第九章系统安全性

第九章

1. 系统安全的复杂性表现在哪几个方面？

答：（1）多面性：大规模系统存在多个风险点，每点都包括物理、逻辑及管理三方面安全。

（2）动态性：信息技术不断发展和攻击手段层出不穷，系统安全问题呈现出动态性。

（3）层次性：系统安全涉及多方面且相当复杂的问题，需要采用系统工程方法解决。

（4）适度性：根据实际需要提供适度安全目标加以实现。

2. 对系统安全性的威胁有哪几种类型？

答：假冒身份、数据截获、拒绝服务、修改伪造信息、否认操作、中断传输、通信量分析。

3. 攻击者可通过哪些途径对软件和数据进行威胁？

答：数据截获，修改信息，伪造信息，中断传输

4. 可信计算机系统评价标准将计算机系统的安全度分为哪几个等级？

答：该标准将计算机系统安全程度分为 8 级，有 D1(安全度最低级)、C1(自由安全保护级)、C2(受控存取控制级)、B1、B2、B3、A1、A2。

5. 何谓对称加密算法和非对称加密算法？

答：对称加密也叫私钥加密，是指加密和解密使用相同密钥的加密算法。加密密钥能够从解密密钥中推算出来，解密密钥也能够从加密密钥中推算出来。在大多数对称算法中，加密和解密的密钥相同，也称为保密密钥算法或单密钥算法。

非对称加密算法需要两个密钥：公钥（publickey）和私钥（privatekey）。公钥与私钥配对，如果用公钥加密数据，只有用对应的私钥才能解密。

6. 什么是易位法和置换算法？试举例说明置换算法。

答：易位法是指按一定规则，重新安排明文中的比特或字符顺序形成密文，而字符本身保持不变。置换法是按照一定规则，用一个字符去置换另一个字符形成密文。

如：How are you? 的每个字符用后面的字母代替就是 lpx bsf zpv?

7. 试说明 DES 加密的处理过程。

答：分为四个阶段：

第一阶段，将明文分出 64 位明文段，并做初始易位得到 X_0 ，左移 32 位，记为 L_0 ，右移 32 位，记为 R_0 。

第二阶段，对 X_0 进行 16 次迭代，每一次用 56 位加密密钥 K_i 。

第三阶段，把经过 16 次迭代处理的结果的左 32 位与右 32 位互换位置。

第四阶段，进行初始易位的逆变换。

8. 试说明非对称加密的主要特点。

答：非对称加密算法复杂、安全性依赖于算法与密钥，加密解密速度慢。对称密码体制只有密钥，安全性就是密钥的安全性，而非对称加密有公钥和私钥，安全性更强。

9. 试说明保密数据签名的加密和解密方式。

答：（1）发送者 A 可用自己的私用密钥 K_{da} 对明文 P 进行加密，得到密文 $DK_{da}(P)$ 。

（2）A 再用 B 的公钥 K_{eb} 对 $DK_{da}(P)$ 加密，得到 $EK_{eb}(DK_{da}(P))$ 后送 B。

（3）B 收到后，先用私钥 K_{db} 解密，得到 $DK_{da}(EK_{eb}(DK_{da}(P))) = DK_{da}(P)$ 。

（4）B 再用 A 的公钥 K_{ea} 对 $DK_{da}(P)$ 解密，得到 $EK_{eb}(DK_{da}(P)) = P$ 。

10. 数字证书的作用是什么？用一例来说明数字证书的申请、发放和使用过程。

答：数字证书又称公钥证书，用于证明通信请求者的身份。

数字证书的申请、发放和使用过程如下：

- (1) 用户 A 先向 CA 申请数字证书，A 应提供身份证明和希望使用的公钥 A。
- (2) CA 收到 A 发来的申请报告后，若接受申请，便发给 A 一份数字证书，其中包括公钥 A 和 CA 发证者的签名等信息，并对所有信息利用 CA 私钥加密（即对 CA 进行数字签名）。
- (3) 用户 A 在向 B 发送信息时，由 A 用私钥对报文加密（数字签名），连同证书发给 B。
- (4) 为能对收到的数字证书解密，用户 B 须向 CA 申请获得 CA 的公钥 B。CA 收到用户 B 的申请后，可决定将公钥 B 发给用户 B。
- (5) 用户 B 利用 CA 公钥 B 对数字证书解密，确认该数字证书系原件，并从数字证书中获得公钥 A，并且确认该公钥 A 系用户 A 的密钥。
- (6) 用户 B 再利用公钥 A 对用户 A 发来的加密报文解密，得到发来报文的真实明文。

11. 何谓链路加密？其主要特点是什么？

答：链路加密是对网络相邻节点间的通信线路上传输的数据的加密过程。特点是：

- （1）相邻节点间的物理信道上传输的报文是密文，在所有中间节点上的报文则是明文。
- （2）对不同的链路分别采用不同的加密密钥。

12. 何谓端-端加密？其主要特点是什么？

答：端-端加密是在源主机或前端机 FEP 高层（从传输层到应用层）对传输数据进行的加密。

特点：（1）整个网络传输过程中的报文正文都是密文，信息到达目标主机后才译成明文。

- （2）不能对报头中的控制信息加密，否则中间结点无法得知目标地址和控制信息。

13. 可利用哪几种方式来确定用户身份的真实性？

答：（1）口令密码组合；（2）物理标志（3）生物标志（4）公开密钥

14. 在基于口令机制的身份认证技术中，通常应满足哪些要求？

答：口令长度适中、自动断开连接、隐蔽回送显示、记录和报告。

15. 基于物理标志的认证技术又可细分为哪几种？

答：主要有基于磁卡或 IC 卡的两种认证技术。

16. 智能卡可分为哪几种类型？这些是否都可用于基于用户持有物的认证技术中？

答：智能卡分为存储器卡、微处理器卡和密码卡等类型。

存储器卡没有安全功能，不能用于基于用户持有物的认证；微处理器卡和密码卡采用了加密措施，可以用于基于用户持有物的认证。

17.被选用的人的生理标志应具有哪几个条件？请列举几种常用的生理标志。

答：被选用的生理标志应具有三个基本条件，即足够的可变性、稳定性好、不易伪装。

常用的生理标志是指纹、视网膜组织、声音、手指长度等。

18. 对生物识别系统的要求有哪些？一个生物识别系统通常是有哪几部分组成的？

答：对生物识别系统的要求有：性能满足要求（抗欺骗和防伪防攻击）、能被用户接受、系统成本适当。

一个生物识别系统通常由注册和识别两部分组成。注册部分配有一张用户注册表，识别部分要对用户进行身份认证和生物特征识别。

19. 试详细说明 SSL 所提供的安全服务。

答：SSL 称为安全套接层协议，用于提供 Internet 上的信息保密，身份认证服务，目前 SSL 已成为利用公开密钥进行身份认证的工业标准。

SSL 提供的安全服务有：申请数字证书（服务器申请数字证书、客户申请数字证书）和 SSL 握手协议（身份认证、协商加密算法和协商加密密钥）。

20.什么是保护域？进程与保护域之间存在着什么动态联系？

答：保护域是进程对一组对象访问权的集合，规定了进程能访问对象和执行的操作系统。

进程与保护域之间的动态联系是指进程的可用资源集在其生命周期中是变化的；进程运行在不同的阶段可以根据需要从一个保护域切换到另一个保护域。

21.试举例说明具有域切换权的访问控制矩阵。

答：在访问矩阵中增加几个对象，分别作为访问矩阵中的几个域，当且仅当 $switch$ 包含在 $access(i,j)$ 时，才允许进程从域 i 切换到域 j 。例如在下图中，域 $D1$ 和 $D2$ 对应的项目中有 S ，故允许域 $D1$ 中的进程切换到域 $D2$ 中，在域 $D2$ 和 $D3$ 中也有 S ，表示 $D2$ 域中进行的进程

可切换到域 $D3$ 中，但不允许该进程再从域 $D3$ 返回到域 $D1$ 。

22.如何利用拷贝权来扩散某种访问权？

答：如果域 i 具有对象 j 的某访问权 $access(i,j)$ 的拷贝权，则运行在域 i 的进程可将其访问权 $access(i,j)$ 扩展到访问矩阵同一列中的其它域，即为运行在其它域的进程也赋予关于同一对象的同样访问（ $access(i,j)$ ）。

23.如何利用拥有权来增删某种访问权？

答：如果域 i 具有关于对象 j 的所有权，则运行在域 i 的进程可以增删在 j 列的任何项中的任何访问权。或该进程可以增删在任何其它域中运行的进程关于对象 j 的任何访问权。

24.增加控制权的主要目的是什么？试举例说明控制权的应用。

答：控制权用于改变某个域中运行进程关于不同对象的访问权。若某域访问权 $access(i,j)$ 中含有控制权 C ，则运行在 D_i 域中的进程能改变运行在 Q_j 域中的任何进程关于任何对象的任何访问权。

25. 什么是访问控制表？什么是访问权限表？

答：访问控制表是指对访问矩阵按列划分，为每列建立一张访问控制表 **ACL**，由有序对（域，权集）组成，用来保证系统安全性的一种手段。

访问权限表是指对访问矩阵按行划分，由每行构成一张访问权限表。

26. 系统如何利用访问控制表和访问权限表来实现对文件的保护？

答：当进程第一次试图访问一个对象时，必须先检查访问控制表，查看是否有权访问该对象。如果无则拒绝访问，并构成一个例外异常事件；否则便允许访问，并为之建立访问权限，以便快速验证其访问的合法性。当进程不再访问该对象时便撤销该访问权限。

27. 什么是病毒？它有什么样的危害？

答：病毒是编制或者在计算机程序中插入的破坏计算机功能或数据，影响计算机系统使用并且能够自我复制的一组计算机指令或程序代码。

计算机病毒的危害：占用系统空间、占用处理机时间、破坏文件、使机器运行异常。

28. 计算机病毒的特征是什么？它与一般的程序有何区别？

答：计算机病毒的特征是寄生性、传染性、隐蔽性和破坏性。

它与一般程序的区别是：病毒程序通常不是独立的程序，具有自我复制和迅速传播的传染性，想方设法隐藏自身，存在的基本目标就是破坏性。

29.什么是文件型病毒？试说明文件型病毒对文件的感染方式。

答：文件型病毒是指采用寄生方式附着在正常程序里，病毒发作时原来程序仍能正常运行，

以致用户不能及时发现而长期潜伏下来的病毒。

文件型病毒对文件的感染方式是主动攻击和执行时感染的方式。

30.病毒设计者采取了哪几种隐藏方式来让病毒逃避检测？

答：(1)隐藏于目录和注册表空间。 (2)隐藏于程序的页内零头里。

(3)更改用于磁盘分配的数据结构。 (4)更改坏扇区列表。

31.用户可采用哪些方法来预防病毒？

答：(1)定期在外存备份重要软件和数据 (2)使用安全性高的操作系统

(3)使用正版软件 (4)使用高性能反病毒软件

(5) 不轻易打开来历不明的电子邮件 (6)定期检查外存并清除病毒

32.试说明基于病毒数据库的病毒检测方法。

答：(1)建立病毒数据库 (2)扫描硬盘上的可执行文件

第十章

1. UNIX 系统具有哪些特征？

答：开放性、多用户多任务环境、功能强大高效、丰富网络功能、支持多处理器。

2. 试说明 UNIX 系统的内核结构。

答：UNIX 内核结构分四层：最底层是硬件，次底层是 OS 核心，第二层是 OS 与用户接口

shell 及编译程序等，最高层是应用程序。

3. UNIX 系统中的 PCB 含哪几部分？用图说明各部分之间的关系。

答：UNIX 系统中的 PCB 含进程表项、U 区、系统区表、进程区表。

4. 进程映像含哪几部分？其中系统级上、下文动态部分的作用是什么？

答：进程映像含用户上下文、寄存器上下文、系统级上下文。

系统级上下文动态部分的作用是当因中断或系统调用进入核心状态时，核心把一个寄存器上下文压入核心栈，退出系统调用时，核心又弹出寄存器上下文，在上下文切换时，核心将压入老进程的上下文，弹出新进程的上下文。

5. 在 UNIX 系统中用于进程控制的主要系统调用有哪些？它们各自的主要功能是什么？

答：用于进程控制的主要系统调用有：

- (1) **fork** 系统调用：用于创建新进程
- (2) **exit** 系统调用：实现进程自我终止
- (3) **exec** 系统调用：改变进程原有代码
- (4) **wait** 系统调用：将调用进程挂起并等待子进程终止

6. 为创建一个新进程，须做哪些工作？

答：为新进程分配一个进程表项和进程标志符；检查同时运行的进程数目；拷贝进程表项中的数据；子进程继承父进程的所有文件；为子进程创建进程上下文；子进程执行。

7. 为何要采取进程自我终止方式？如何实现 **exit**？

答：为了及时回收进程占用的资源，在进程任务完成后应尽快撤销。Unix 内核用 **exit** 实现进程的自我终止。父进程在创建子进程时，应在子进程末尾安排 **exit** 使子进程能自我终止。实现 **exit** 的具体操作是：关闭软中断、回收资源、写记账信息和置进程为僵死状态。

8. 在 UNIX 系统中采用了何种调度算法？如何确定进程的优先数？

答：UNIX 系统采用动态优先数轮转的进程调度算法。优先数确定公式：

优先数 = (最近使用 CPU 的时间/2) + 基本用户优先数

9. 在进入 **sleep** 过程后，内核应做哪些处理？

答：进入 **sleep** 过程后，核心首先保存进入睡眠时的处理机运行级，提高处理机的运行优先级屏蔽所有中断，将该进程置为睡眠状态，将睡眠地址保存在进程表项中，将该进程放入睡眠队列。如果进程的睡眠不可中断，在进程上下文切换后，进程便安稳睡眠。当进程被唤醒并被调度执行，将恢复处理机的运行级为进入睡眠时的值，此时允许中断处理机。

10. 试说明信号与中断两种机制间的异同处。

答：不同点：中断有优先级，而信号没有，所有信号皆平等；信号处理程序在用户态运行，而中断处理程序是在核心态运行；还有中断响应及时，而信号响应通常都是延时的。

相同点：都采用异步通信方式；当检测出信号或中断请求时都暂停正在执行的程序而转去执行相应的处理程序；都在处理完毕返回到原来断点；对信号或中断都可进行屏蔽。

11. 扼要说明信号机制中信号的发送和对信号的处理功能。

答：信号发送是指由发送进程把信号送到目标进程的 **proc** 结构中信号域的某一位上。

对信号的处理功能：首先利用系统调用 **signal (sig, func)** 预置对信号的处理方式，

func=1 时屏蔽该类信号；func=0 时，进程收到信号后终止自己；func 为非 0 非 1 时，func 值作为信号处理程序的指针，系统从核心态转为用户态并执行相应的处理程序，处理完毕再返回用户程序的断点处。

12. 什么是管道？无名管道和有名管道的主要差别是什么？

答：管道是指能连接写进程和读进程，并允许它们以生产者消费者方式进行通信的一个共享文件或 pipe 文件。无名管道是个临时文件，是利用系统调用 pipe() 建立起来的无路径名文件，只有调用 pipe 的进程及其子孙进程才能识别此文件描述符而利用该文件（管道）进行通信；有名管道是利用 mknod 系统调用建立的、可以在文件系统中长期存在的有路径名文件，其它进程可以知道其存在，并利用该路径名访问的文件。

13. 在读、写管道时，应遵循哪些规则？

答：（1）对 pipe 文件大小的限制

（2）进程互斥

（3）进程写管道时在管道空间上满足生产者操作规则

（4）进程读管道时在管道空间上满足消费者操作规则

14. 在消息机制中有哪些系统调用？说明它们的用途。

答：在消息机制中的系统调用是 msgctl()、msgsnd()、msgrcv()。

msgctl() 系统调用用于指定的消息队列进行操纵。

msgsnd() 系统调用来发送消息。

msgrcv() 系统调用从指定消息队列中读取一个消息。

15. 在共享存储机制中有哪些系统调用？扼要说明它们的用途

答：在共享存储机制中的系统调用有 shmget()、shmctl()、shmat()。

shmget() 用于建立一块共享存储区，提供该区名字 key 和共享存储区长度 size 等参数。

shmctl() 系统调用用于查询共享存储区的状态信息。

shmat() 系统调用用于将该共享存储区附接到用户给定的某个进程虚地址 shmaddr 上，并指定该存储区的访问属性是只读还是可读可写。

16. 核心在执行 shmget 系统调用时需完成哪些工作？

答：(1) 检查共享存储区表，若找到 key 表项，表明该区已建立，返回共享区描述符 shmid；

(2) 若未找到指定 key 表项，而 flag 标志为 IPC_CREAT 且参数 size 值在系统限制值

内，则分配一系统空闲区作为共享区的页表区，分配相应的内存块，将这些块号填入页表中；

(3)核心在共享区_____x 1yl/____储区和系统区表中，为新建立的共享区分配一空表项，并填上存储区

的关键字及大小、共享区页表始址，指向系统区表项指针等，最后返回共享区描述符 shmid。

17. 在信号量机制中有哪些系统调用？说明它们的用途。

答：在信号量机制中的系统调用是 `semget()` 和 `semop()`。`semget()` 用于用户建立信号量集。
`semop()` 用来对信号量集进行操作。

18. 核心是如何对信号量进行操作纵的？

答：核心根据 `sem_op` 改变信号量的值，分 3 种情况：

若 `sem_op` 值为正，则将其值加到信号量值上，相当于 V 操作；若 `sem_op` 值为负，相当于 P 操作，若信号量值大于操作值的绝对值，则核心将一个负整数加到信号量值上，否则核心将已操作的信号量恢复到系统调用开始时的值；若 `(sem_flg & IPC_NOWAIT)` 为真，便立即返回，否则让进程睡眠等待。

19. 为实现请求调页管理，在 UNIX 系统中配置了那些数据结构？

答：UNIX 系统 V 将进程的每个区分为若干个虚页，这些虚页可以分配到不邻接的页框中，为此设置了一张页表。其中每个表项中，记录了每个虚页和页框的对照关系。

20. 当访问的缺页是在可执行文件上或在对换设备上时，应如何将它们调入内存？

答：(1) 缺页在可执行文件上。如果欲访问虚页对应磁盘块描述表项类型是 `file`，表示该缺

页尚未运行，其拷贝在可执行文件中，核心应将该页调入内存。调入过程是：根据对应系统区表项中的索引结点指针，找到该文件的索引节点，把该页的逻辑块号作为偏移量，查找索引结点中的磁盘块号表，找到磁盘块号，将该页调入内存。

(2) 缺页在对换设备上。核心先为缺页分配一内存页，修改该页表项，指向内存页，并将页面数据表项放入相应散列队列中，把该页从对换设备上调入内存，当 I/O 操作完成时，核心把请求调入该页的进程唤醒。

21. 在将一页换出时，可分成哪几种情况？应如何处理这些情况？

答：分三种情况：(1) 若在对换设备上有被换出页的拷贝，内容未改，则核心只将该页页表项中的有效位清零，将引用计数减 1，将该页框数据表项放入空闲链表中。

(2) 若在对换设备上没有换出页的拷贝，则将该页写到对换设备上。先将所有要换出页链

入到待换出页面链上。当链上页面数达到规定值时才将这些页面写到对换区中。

(3) 在对换设备上有换出页副本，但页内容已修改，核心应释放该页在对换设备上原占有的空间，再重新将该页拷贝到对换设备上，使拷贝内容最新。

22. 如何对字符缓冲区进行分配与回收？

答：在字符设备进行 I/O 时，内核利用 `getc` 过程从空闲字符缓冲区队列中取得空闲缓冲区，若队列空，则无缓冲区可分配，返回；否则从队首取得一个空闲缓冲区，把该缓冲区指针 `bp` 返给调用者。采取互斥访问措施，在过程开始处将处理机优先级提升为 6，在取得空闲缓冲区之后再恢复处理机的优先级。

当不再需要缓冲区时，调用 `putc` 过程释放缓冲区。输入参数是指向已不再需要的缓冲区指针 `bp`，把该缓冲区送回到空闲缓冲区队列的队首指针 `cfreelist` 指向的头部。此时若有申

请空闲缓冲区而阻塞的进程，则唤醒它。对空闲缓冲区队列的访问应互斥进行。

23. 试说明盘块缓冲区的组成和盘块缓冲池的构成。

答：UNIX 系统的每一个盘块缓冲区均有两部分：一是用于存放数据的数据缓冲区；另一部分是缓冲控制块，用于存放对应缓冲区的管理信息。

盘块缓冲池结构：(1) 空闲链表 (2) 散列队列。

24. `getblk()`和 `getblk(dev, blkno)`进程的主要区别是什么？

答：`getblk()`用于从空闲缓冲区队列中获得任意空闲缓冲区。`getblk(dev, blkno)`用于为指定设备 `dev` 和盘块号为 `blkno` 的盘块申请一个缓冲区。仅当要把数据写入特定盘块的内容不在缓冲区时，才调用 `getblk` 过程分配一个空闲缓冲区。

25. 试说明 `gdopen`、`gdstart`、`gdstartegy` 和 `gdintr` 过程的主要功能。

答：`gdopen` 用于打开磁盘驱动器，输入参数是设备号，无输出参数。

`gdstart` 用于装配磁盘控制器中的各个寄存器，然后启动磁盘控制器。

`gdstartegy` 把指定缓冲首部排在磁盘控制器 I/O 队列末尾，并启动磁盘控制器。

`gdintr` 用于磁盘 I/O 传送完成并发出中断请求时的磁盘中断处理过程。

26. 在 UNIX 系统中设置了哪些读和写过程？两者的主要区别是什么？

答：读过程有一般读过程 `bread` 和提前读过程 `breada`。

写过程有一般写过程 `bwrite`、异步写过程 `bawrite` 和延迟写过程 `bdwrite`。

27. 试说明 UNIX 文件系统的特点？

答：A. 文件系统的组织是分级树形结构形式 B. 文件的物理结构为混合索引式文件结构

C. 采用成组链接法管理空闲盘块。D. 引入了索引结点的文件检索技术。

28. 在 UNIX 系统中的文件物理结构采用了何种形式？试举例说明。

答：UNIX 文件物理结构采用混合索引式文件结构。

在查找文件时，只要找到了文件的索引结点，用直接或间接寻址方式获得该文件的盘块。

29. 在 UNIX 系统中如何将文件的逻辑块号转换为物理盘块号？

答：寻址方式不同，转换方法也不同。

(1) 直接寻址，仅当文件的逻辑块号不大于 10 时采用。如访问对象是字节偏移量 9999 处的数据。则 $9999/1024=9$ 余 783，则文件逻辑块号 9，直接索引地址项 `i-addr(9)` 中的

块号，其块内偏移地址为 783 字节处就是文件的 9999 字节处。

(2) 一次间址，仅当文件的逻辑块号大于 10 而不大于 $10+256$ 时采用。如访问对象是字节偏移量 18000 处的数据。则 $18000/1024=17$ 余 592，则逻辑块号为 $10<17<10+256$ ，需要通过一次间接索引方式。先从 `i-addr(10)` 中得到一次间址盘块号，再将逻辑盘块号减 10，根据一次间址中的逻辑块号得到间址块号地址项的下标，再从中得到最终的物理盘块号。这里的逻辑盘块号 17，从 `i-addr(10)` 中得到的块号设为 428，则 $17-10=7$ 就是一次间址号，其对应的盘块号就是要找的物理盘块号，块内偏移地址为 592 处就是文件的 18000 字节处。

(3) 多次间址，仅当文件的逻辑块号大于 266 而不大于 64266 时采用。如访问对象是字节偏移量 420000 处的数据。则逻辑块号为 $266<410<64266$ ，通过二次间址索引。在 `i-addr(11)` 中得到一次间址盘块号，再将逻辑盘块号减 266，根据一次间址中的逻辑块号得到间址块号地址项的下标，再从中得到二次间址，再找到对应的物理块号，块内偏移地址 160 处就是文件的 420000 字节处。

30. 如何对磁盘索引节点进行分配与回收？

答：分配过程 `ialloc` 是：先检查超级块是否上锁，检索 `i` 结点栈空否，从空闲 `i` 结点编号栈中分配一个 `i` 结点并初始化，填写有关文件属性，分配内存 `i` 结点，将磁盘 `i` 结点总数减 1，并置超级块修改标志后返回。

回收过程 `ifree` 是：先检查超级块上锁否；检查 `i` 结点编号栈满否；若 `i` 结点编号栈未满，便使回收的 `i` 结点的编号进栈，并使当前空闲 `i` 结点数加 1；置超级块修改标志后返回。

31. 何时需要构造目录项？核心需完成哪些工作？

答：当用户（进程）要创建新文件时，内核便应在其父目录文件中构造一个目录项；当某进程需要共享另一用户的某文件时内核也将为共享该文件的用户建立一个目录项。由系统调用 `creat` 过程完成目录项的构造。

32. 何时需删除一个目录项？核心须完成哪些工作？

答：对于某用户独享的文件，当该用户不需要它时，应将它删除而腾出存储空间。核心须完成的工作是利用 `unlink` 断开连接，当 `nlink` 值为 0 时，系统将自动删除该文件。