

一、概念题（每空 0.5 分，共 34 分）

1. 现代操作系统的两个最基本的特征是（ ）和（ ）。
2. 操作系统是计算机系统中的一个（ ），它管理和控制计算机系统中的（ ）。
3. 允许多个用户以交互方式使用计算机的操作系统称为（ ），允许多个用户将多个作业提交给计算机集中处理的操作系统称为（ ），计算机系统能及时处理过程控制数据并做出响应的操作系统称为（ ）。
4. 用户与操作系统之间的接口主要分为（ ）和（ ）两类。
5. 作业的输入方式有五种，如联机输入方式、（ ），（ ），（ ）和（ ）。
6. 进程控制块的初始化工作包括（ ），（ ）和（ ）。
7. 程序并发执行与顺序执行时相比产生了一些新特性，分别是：（ ），（ ）和（ ）。
8. 进程是一个程序对某个数据集的（ ）。
9. 如果系统有 N 个进程，则在等待队列中进程的个数最多可为（ ）个。
10. 在操作系统中，不可中断执行的操作称为（ ）。
11. 如果信号量的当前值为-4，则表示系统中在该信号量上有（ ）个等待进程。
12. 在操作系统中引入线程概念的主要目的是（ ）。
13. 在单用户单任务环境下，用户独占全机，此时机内资源的状态，只能由运行程序的操作加以改变，此时的程序执行具有（ ）性和（ ）性。
14. 并发程序之间的相互制约，是由于它们（ ）和（ ）而产生的，因而导致程序在并发执行时，具有（ ）特征。
15. 在多用户环境下，由多个程序共享一台计算机，机内资源的状态将由多个程序来改变，因此使程序失去了在顺序执行时具有的（ ）和（ ）特性。
16. 进程最基本的特征是（ ），因为进程的实质是程序的一次执行过程，而且该特征还表现在进程由（ ）而产生，由（ ）而执行，由（ ）而消亡，即进程具有一定的生命期。
17. 在操作系统中，进程是一个（ ）的基本单位，也是一个（ ）和（ ）的基本单位。
18. 当前进程若因时间片用完而被暂停执行时，该进程应转变为（ ）状态；若因发生某事件而不被继续运行时，该进程应转变为（ ）状态。处于就绪状态的进程被调度应转变为（ ）状态。
19. 用户为阻止进程继续运行，应利用（ ）原语；若进程正在执行，应转变为（ ）状态。不久，若用户要恢复其运行，应利用（ ）原语，此时进程应转变为（ ）状态。
20. 每执行一次 V 操作，表示（ ）；若 $S.value \leq 0$ ，则表示（ ），此时应（ ）。
21. 在利用信号量实现进程互斥时，应将（ ）置于（ ）和（ ）之间。
22. 在每个进程中访问（ ）的那段代码称为临界区。为实现对它的共享，应保证进程（ ）

进入自己的临界区,为此在每个进程中的临界区前面应设置(),在临界区之后应设置()。

23. 进程通信的类型有()、()和()三类。

为实现消息缓冲通信,在PCB中应增加()、()和()三个数据项。

24. 系统产生死锁是指(),产生死锁的基本原因是()和()。

25. 死锁是指在系统中的多个()无限期地等待永远不会发生的条件。

26. 死锁产生的必要条件有四个,即()、()、()、()。

27. 对待死锁,一般应考虑死锁的预防、避免、检测和解除四个问题。典型的银行家算法是属于(),破坏循环等待条件是(),而剥夺资源是()的基本方法。

28. 线程是();线程基本上不拥有资源,并且可以并发操作。

二、选择题(每空 0.5 分,共 15 分)

1. 操作系统负责管理计算机系统的(),其中包括处理机、内存、外围设备和文件。

A、程序; B、文件; C、资源; D、进程

2. 操作系统是一种()。

A. 应用软件 B. 系统软件 C. 通用软件 D. 工具软件

3. 没有下列设备计算机无法工作():

A、硬盘; B、软盘; C、内存; D、打印机

4. 在单一处理器上,将执行时间有重叠的几个程序称为()

A. 顺序程序 B. 多道程序 C. 并发程序 D. 并行程序

5. 在下列系统中,()是实时系统。

A. 计算机激光照排系统 B. 航空订票系统 C. 办公自动化系统 D. 计算机辅助设计系统

6. 引入多道程序的目的在于()。

A. 充分利用 CPU, 减少 CPU 等待时间 B. 提高实时响应速度

C. 有利于代码共享, 减少主、辅存信息交换量 D. 充分利用存储器

7. 已经获得除()以外的所有运行所需资源的进程处于就绪状态

A. 存储器 B. 打印机 C. CPU D. 磁盘空间

8. 进程调度的关键问题: 一是选择合理的(), 二是恰当地进行代码转换

A. 时间片间隔 B. 调度算法 C. CPU 速度 D. 内存空间

9. 在多进程的系统中, 为了保证公共变量的完整性, 各进程应互斥进入临界区, 所谓临界区是指():

A、一个缓冲区 B、一段数据区 C、同步机制 D、一段程序

10. 一个进程是():

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

11. 操作系统中, 当(), 进程从执行状态转变为就绪状态。

A. 进程被进程调度程序选中, B. 时间片完

C. 等待某一事件 D. 等待的时间发生

12. 采用资源剥夺法可解除死锁, 还可以采用()方法解除死锁

A. 执行并行操作 B. 撤消进程 C. 拒绝分配新资源 D. 修改信号量

13、产生死锁的四个必要条件是：互斥、() 循环等待和不剥夺。

A. 请求与阻塞 B. 请求与保持 C. 请求与释放 D. 释放与阻塞

14、发生死锁的四个必要条件有四个，要防止死锁的发生，可以破坏这四个必要条件，但是破坏() 是不太实际。

A. 互斥 B. 不可抢占 C. 部分分配 D. 循环等待

15、资源的按序分配策略可以破坏()

A. 互斥使用资源 B. 占有且等待资源 C. 非抢占资源 D. 循环等待资源

16、在() 的情况下，系统出现死锁。

A. 计算机系统发生了重大故障 B. 有多个封锁的进程同时存在

C. 若干进程因竞争资源而无休止地相互等待他方释放已占有的资源

D. 资源数大大小于进程数或进程同时申请的资源数大大超过资源总数

17、银行家算法是一种() 算法。

A. 死锁解除 B. 死锁避免 C. 死锁预防 D. 死锁检测

18、某系统中有 3 个并发进程，都需要同类资源 4 个，试问该系统不会产生死锁的最少资源数是()。

A. 9 B. 10 C. 11 D. 12

19、下面关于线程的说法，错误的是()

A. 同一进程中的线程可以共享该进程的内存空间

B. 线程是调度和执行单位

C. 不同的线程可执行相同的程序

D. 线程是资源分配单位

20、下面关于进程和线程的说法，正确的是()

A. 线程是进程中可独立执行的子任务，一个进程可以包含一个或多个线程，一个线程可以属于一个或多个进程

B. 多线程技术具有明显的优越性，如速度快、通信简便、设备并行性高等

C. 由于线程不作为资源分配单位，线程之间可以无约束地并行执行

D. 线程又称为轻型进程，因为线程都比进程小

三、简答题（每题 2 分,共 36 分）

1、简述操作系统的五大管理功能。

2、什么是批处理系统？为什么要引入批处理系统？

3、什么叫多道程序？试述多道程序涉及技术的基本思想及特征，为什么对作业进行多道批处理可以提高系统效率？

4、何为分时系统？简述其特点。

5、分时系统和实时系统有何不同？

6、实现多道程序解决哪些问题？

7、分布式操作系统应具有哪些功能？

8、什么是作业、作业步和作业流？作业管理包括哪些内容？

9、在多道操作系统控制下，一个作业反复执行多次，它的运行时间都相同吗？为什么？

10、作业调度的功能是什么？作业调度算法应考虑的主要因素是什么？

- 11、有哪些基本的作业调度算法？在什么情况下调用作业调度程序。
- 12、人机交互界面是什么？
- 13、计算机人机交互界面的三代发展特点是什么？
- 14、什么是进程？作业与程序和进程有何区别？
- 15、说明进程的结构、特征和基本状态。
- 16、什么是临界资源？什么是临界区？
- 17、消息缓冲通信技术是一种高级通信机制，
 - (1) 试叙述高级通信机制与低级通信机制 P, V 元语操作的主要区别。
 - (2) 给出消息缓冲机制的基本工作原理
 - (3) 消息缓冲通信机制中提供发送原语 SEND (RECEIVE, A)，调用参数 A 表示发送消息的内存区首地址，试设计相应的数据结构，并用 PV 原语操作实现 SEND 原语。
- 18、试说明进程互斥、同步和通信三者之间的关系？

四、综合题（共20分）

1、A、B两个程序，程序A按顺序使用CPU 10s，使用设备甲5s，使用CPU 5s，使用设备乙5s，最后使用CPU 10s；程序B按顺序使用设备甲10s，使用CPU 10s，使用设备乙5s，使用CPU 5s，使用设备 10s。试问：

- a. 在顺序环境下，CPU的利用率是多少？
- b. 在多道程序环境下，CPU的利用率是多少？

2、若程序 P_a 和 P_b 单独执行时分别用 T_a 和 T_b ， $T_a=1$ 小时， $T_b=1.5$ 小时，其中CPU工作时间分别为 $t_a=18$ 分钟， $t_b=27$ 分钟，如果采用多道程序设计方法，让 P_a 和 P_b 并行工作，假定CPU利用率达到50%，另外15分钟系统开销，请问系统效率提高多少？

3、下列程序执行时，系统的输出可能是什么？

```
{
    a=55;
    pid=fork();
    if (pid==0){
        sleep(5);
        a=99;
        sleep(5);
        printf("child leaving\n");
        exit(0);
    }
    else
    {
        sleep(7);
        printf("a=%d\n",a);
        wait(0);
        printf("parent child exited\n");
    }
}
```

4、说明在 Linux 系统中下述程序的输出结果。

```

1  #include "csapp.h"
2
3  int main()
4  {
5      pid_t pid;
6      int x = 1;
7
8      pid = Fork();
9      if (pid == 0) { /* child */
10         printf("child : x=%d\n", ++x);
11         exit(0);
12     }
13
14     /* parent */
15     printf("parent: x=%d\n", --x);
16     exit(0);
17 }

```

- 5、某车站售票厅，任何时刻最多可容纳 20 名购票者进入，当售票厅中少于 20 名购票者时，则厅外的购票者可立即进入，否则需在外面等待。若把一个购票者看作一个进程，请用 PV 操作管理这些并发进程时，应怎样定义信号量，写出信号量的初值以及信号量各种取值的含义。
- 6、有一个阅览室，共有 100 个座位，读者进入时必须先在一张登记表上登记，该表为每一座位列一表目，包括座号和读者姓名等，读者离开时要消掉登记的信息，试问：

(1) 为描述读者的动作，应编写几个程序，设置几个进程？

(2) 试用PV操作描述读者进程之间的同步关系。

- 7、用信号量解决“独木桥”问题：同一方向的行人可连续过桥，当某一方向有人过桥时，另一个方向的行人必须等待；当某一方向无人过桥时，另一方向的行人可以过桥。

- 8、设系统中有3种类型的资源（A，B，C）和5个进程P1、P2、P3、P4、P5，A资源的数量为17，B资源的数量为5，C资源的数量为20。在 T_0 时刻系统状态见下表（ T_0 时刻系统状态表）所示。系统采用银行家算法实施死锁避免策略。

T_0 时刻系统状态表

	最大资源需求量			已分配资源数量		
	A	B	C	A	B	C
P1	5	5	9	2	1	2
P2	5	3	6	4	0	2
P3	4	0	11	4	0	5
P4	4	2	5	2	0	4
P5	4	2	4	3	1	4

- (1) T_0 时刻是否为安全状态？若是，请给出安全序列。
 - (2) 在 T_0 时刻若进程P2 请求资源 (0, 3, 4)，是否能实施资源分配？为什么？
 - (3) 在(2)的基础上，若进程 P4 请求资源 (2, 0, 1)，是否能实施资源分配？为什么？
- 在(3)的基础上，若进程P1请求资源 (0, 2, 0)，是否能实施资源分配？为什么？

参考答案

一、概念题 (每空0.5分, 共34分)

1. (程序的并发执行) (资源共享)。
2. (管理者), (各种硬件和软件资源)。
3. (分时系统), (批处理系统), (实时系统)。
4. (命令接口) 和 (程序接口)。
5. (脱机输入方式) (直接耦合方式) (SPOOLING 系统) 和 (网络输入方式)。
6. (标识信息), (处理机状态信息) 和 (处理机控制信息)。
7. (间断性), (失去封闭性) 和 (不可再现性)。
8. (运行过程)。
9. (N-1)。
- 10 (原语操作)。
- 11 (4)。
- 12 (使得多个程序更好的并发执行同时有尽量减少系统的开销, 有效的改善多处理机的性能)。
- 13 (封闭性) (可再现性)
- 14 (相互合作) (共享资源) (间断性)
- 15 (封闭性) (可再现性)
- 16 (动态性) (创建) (调度) (撤销)
- 17 (资源分配) (独立运行) (调度)
- 18 (就绪) (阻塞) (执行)
- 19 (挂起) (静止阻塞) (激活) (活动就绪)
- 20 (释放一个单位资源)
(仍有请求该资源的进程被阻塞)
(唤醒等待该资源的队首进程, 并将之插入就绪队列)
- 21 (临界区) (进入区) (推出区)
- 22 (临界资源) (互斥地) (P 操作) (V 操作)
- 23 (共享存储区通讯) (消息通讯) (管道通讯)
- 24 (若干进程等待被其他进程占用而又不能被释放的资源) (资源不足) (进程在使用资源的时候推进速度和顺序不当)
- 25 (进程)
- 26 (互斥)、(不剥夺)、(请求与保持) (循环等待)
- 27 (死锁的避免) (死锁预防) (死锁的解除)
- 28 (利用 CPU 的最小单位, 调度和分配的基本单位, 是花费最小的开销实体)

二、选择题 (每题0.5分, 共10分)

1. C 2. B 3. C 4. C 5. B 6. A 7. C 8. B 9. D 10. C
11. B 12. B 13. B 14. A 15. D 16. C 17. B 18. B 19. D 20. B

三、简答题 (每题2分, 共36分)

1. 解答:

作业管理: 包括任务管理、界面管理、人机交互、图形界面、语音控制和虚拟现实等。

文件管理: 又称为信息管理。

存储管理: 实质是对存储“空间”的管理, 主要指对内存的管理。

设备管理：实质是对硬件设备的管理，其中包括对输入输出设备的分配、启动、完成和回收。

进程管理：又称外理机管理，实质上是对处理机执行“时间”的管理，即如何将 CPU 真正合理地分配给每个任务

2. 解答：批处理系统指用户的作业成批的处理，作业建立、过渡、完成都自动由系统成批完成。因为 58~64 年，晶体管时代，计算机速度、容量、外设品种和数量等方面和第一代计算机相比都有了很大发展，计算机速度有几十倍、上百倍的提高，故使手工操作的慢速度和计算机运算的高速度之间形成一对矛盾。只有设法去掉人工干预，实现作业自动过渡，这样就出现了成批处理。

3. 解答：多道程序设计技术是在计算机内存中同时存放几道相互独立的程序，使它们在管理程序控制下，相互穿插还差运行。当某道程序因某种原因不能继续运行下去时候，管理程序就将另一道程序投入运行，这样使几道程序在系统内并行工作，可使中央处理机及外设尽量处于忙碌状态，从而大大提高计算机使用效率。在批处理系统中采用多道程序设计技术形成多道批处理系统，多个作业成批送入计算机，由作业调度程序自动选择作业运行，这样提高了系统效率。

4. 解答：分时系统采用时间片轮转办法，使一台计算机同时为多个终端服务。特点：多路调制性：多个联机用户可同时使用一台计算机。独占性：用户感觉不到其他人在和他一起共享计算机及资源。交互性：用户可与计算机会话，提出要求、命令。

5. 解答：分时系统控制的主动权在计算机，计算机按一定时间间隔，以固定时间片或不固定时间片去轮流完成多个提交的任务，只是在用户反应相对较慢时，不感到机器“走开”。而实时系统控制的主动权在用户，用户规定什么时间要计算机干什么，计算机不能“走开”。

分时系统通用性强，交互性强，及时响应性要求一般（通常数量级为秒）；实时系统往往是专用的，系统与应用很难分离，常常紧密结合在一起，实时系统并不强调资源利用率，而更关心及时响应性（通常数量级为毫秒或微秒）、可靠性等。

6. 解答：首先包括分时使用硬件的硬件设计技术：CPU 和通道分时使用内存、只读存储器和数据通道等；通道与通道分时使用 CPU、内存、通道的公用控制部分等；同一通道中的 I/O 又分时使用内存、通道等。其次包括共享硬件和软件资源的软件设计技术：包括引入“进程”“线程”等技术。

7. 解答：分布式操作系统是一个完整的一体化的系统且又具有分布处理能力，运行在不具有共享内存的多台计算机上，但在用户眼里却象是一台计算机。它的系统特征：①需要一个全局的文件系统；②所有 CPU 上运行同样的内核，统一的管理和控制机构，进行优化的协调工作；③要有一个单一的、全局的进程通信机制。提供选择优化本地和远地的资源利用；④有全局的保护机制等。它具有多机合作（自动的任务分配和协调系统可取得短的响应时间、高的吞吐量。）和坚强性（一台计算机出现故障，不影响整个系统工作）。

8. 解答：用户角度看，作业就是让计算机做的一件事，完成的一项任务。系统角度看，作业是比程序更广的概念、数据和作业说明书三部分组成。作业步：要求计算机系统做的一项相对独立的工作，是顺序执行的工作单元。作业流是作业步的控制流程。

9. 解答：不一定相同，因为在多道操作系统的控制下，同一个作业在反复执行时，会被其他进程在优先满足调度算法的条件下打断，将其执行中转周期拉长。

10. 解答：①采用作业控制块（JCB）表格，记录系统中各作业工作状况；②根据选定的调度算法，从后备作业中选出一部分（多道情况）或一个作业投入运行；③为被选中的作业做好运行前的准备工作，包括选择系统进行相应的“进程”执行单元以及为这些“进程”分配系统资源，首先判断用户的资源要求是否能够满足；④作业运行结束后的善后处理工作，例如，回收资源和记帐等工作。作业调度算法应考虑的主要因素是：①应与系统的整体设计目标一致。例如，批处理系统系统应注意系统效率的发挥，其调度算法应尽量增加系统的平均吞吐量；分时系统的调度系统应提供好的响应时间。②考虑系统中各种资源的负载均匀。③保证作业的执行，

不要让用户等待时间过长。④对一些专用资源的使用特性的考虑。

11. 解答：先来先服务、最短作业优先法、最高响应比优先法、定时轮转法、优先数法等。当完成作业从后备状态到运行状态和从运行状态到完成状态的转变时要使用作业调度算法。

12. 解答：人机交互界面也可称为用户界面，是计算机系统的一个重要组成部分，人机通信及计算机应用都要通过人机交互界面控制使用计算机。它包括：命令行界面和系统调用、图形界面、虚拟现实的界面。

13. 解答：一维命令行、二维图形界面和三维虚拟现实。

14. 解答：进程是一个具有一定独立功能的程序关于某个数据集合的一次运行活动。它是操作系统动态执行的基本单元，在传统的操作系统中，进程既是基本的分配单元，也是基本的执行单元。（在Windows NT等采用微内核结构的现代操作系统中，进程的功能发生了变化：它只是资源分配的单位，而不再是调度运行的单位，其调度运行的基本单位是线程。

作业是面向用户为主的，在非多道程序处理中，作业和进程是一一对应的，在任何时间只允许一个作业进入系统。在多道程序处理系统，进程是比作业小的概念，作业可分为多个进程获得CPU 服务。对处理机的分配在逻辑上是分两级进行的，第一级是宏观调度，也称作业调度，第二级是微观调度，也成进程调度。作业是一个比程序更广的概念，它由程序、数据和说明书三部分组成。

15.解答：结构：PCB (进程控制块)+程序+数据集合。

特征：动态性、并发性、独立性、制约性、结构性。基本状态：就绪态、执行态、等待态。

16. 解答：一次仅允许一个进程使用的资源称为临界资源。

每个进程中访问临界资源的那段程序称为临界区（临界资源是一次仅允许一个进程使用的共享资源）。每次只准许一个进程进入临界区，进入后不允许其他进程进入。

17. 解答：

（1）高级通信机制中程序员直接利用系统提供的一组通信命令来实现通信，send (p2, m1) 表示将信息 m1 发送给接收进程 p2，receive (p1, m2) 表示接收由进程 p1 发来的消息 m1

低级通信机制中相互通信的进程共享某些数据共享存储区，进程之间能够通过他们进行通信，传输的数据量较小

（2）消息缓冲机制主要利用了消息缓冲区这一数据结构，首先由发送进程将已经在自己内存空间发送区的数据用发送原语复制到消息缓冲区中去，同时将自己的首指针挂在接收进程的PCB 中，再由接收进程用接收原语操做从缓冲区中复制到自己的数据区，完成从发送进程到接收进程的数据传送。

（3）消息缓冲通信机制中提供发送原语 SEND (RECEIVE, A)，调用参数 A 表示发送消息的内存区首地址，试设计相应的数据结构，并用 PV 原语操作实现 SEND 原语。

消息缓冲区结构描述如下：

```
type message buffer=record
```

```
    sendid:发送进程标识
```

```
    length: 消息长度
```

```
    text:  消息正文
```

```
end
```

在接收进程的 PCB 中增加数据项：

```
    first:消息队列首指针
```

```
    only:消息队列互斥信号量
```

```
    resours:消息队列资源信号量
```

send 的 pv 原语操作为：

设发送进程为 x，申请的缓冲区为 y，接收进程为 z

```
procedure send(receiver, x)
```

```
begin
```

```
    getbuf (x. length, y)
```

申请消息缓冲区

```

x.sendid=y.sendid;          将要发送的消息送到消息缓冲区
x.length=y.length;
x.text=y.text;
getid(PCB set,receiver.z)   获得接收进程 Z 的内部标识
P(z.only);                  申请互斥资源
insert(z.first,y);          将缓冲区插入接收进程消息队列
V(z.only);                   释放互斥资源
V(z.resours);                使消息队列资源信号量+1

```

end

18.解答：进程的同步与互斥是指进程在推进时的相互制约关系。在多道程序系统中，由于资源共享与进程合作，这种进程间的制约称为可能。为了保证进程的正确运行以及相互合作的进程之间交换信息，需要进程之间的通信。

进程之间的制约关系体现为：进程的同步和互斥。

进程同步：它主要源于进程合作，是进程间共同完成一项任务时直接发生相互作用的关系。为进程之间的直接制约关系。在多道环境下，这种进程间在执行次序上的协调是必不可少的。

进程互斥：它主要源于资源共享，是进程之间的间接制约关系。在多道系统中，每次只允许一个进程访问的资源称为临界资源，进程互斥就是保证每次只有一个进程使用临界资源。进程通信是指进程间的信息交换。PV操作作为进程的同步与互斥工具因信息交换量少，效率太低，称为低级通信。而高级通信则以较高的效率传送大批数据。

四、综合题（共20分）

1. 解答：

a. 程序A和程序B顺序执行时，程序A执行完毕，程序B才开始执行。两个程序共耗时80s，其中占用CPU时间为40s，因此顺序执行时CPU的利用率为50%。

b. 在多道程序环境下，两个程序并发执行，执行情况如下表所示：

CPU	程序A	程序B	程序A	程序B	空闲	程序A
程序A	CPU	设备甲	等待	CPU	设备乙	CPU
程序B	设备甲	CPU	设备乙	CPU	等待	设备乙

两个程序共耗时45s，其中占用CPU时间为40s，故CPU的利用率为40/45=88.89%。

2. 解答：

在多道系统下 P_a 和 P_b 共用的执行时间为 $(18+27)/50\%=90$ （分钟），系统效率的提高为：

$$[(60+90)-(90+15)]/(60+90)=45/150=30\%。$$

3. 解答：

由于同时存在了两个进程：调用进程（父进程）和子进程，该一个程序会被两个进程分别执行（执行两次，当然相应地，fork()函数也会返回两次，其中在子进程中返回0），而且由于操作系统的进程调度方法未知，所以在不同操作系统上执行可能会出现以下四种情况：

①先父进程，后子进程而后结束；执行结果为：

```

55
parent child exited
child leaving

```

②先子进程，然后父进程而后结束；执行结果为：

```

child leaving
55

```

```

parent child exited

```

③先父进程，在执行到wait(0)时进程调度发生，从而转去执行子进程，结束后继续执行完父进程，执行结果为：

55

child leaving
parent child exited

④子进程创建彻底失败，只执行父进程，而后结束；执行结果为：

55

parent child exited

4. 解答：

parent: x=0
child: x=2

5. 解答：定义一信号量 S ，初始值为 20。意义： $S > 0$ S 的值表示可继续进入售票厅的人数； $S = 0$ 表示售票厅中已有 20 名顾客（购票者）； $S < 0$ $|S|$ 的值为等待进入售票厅的人数。

6. 解答：

读者的动作有两个，一是填表进入阅览室，这时要考虑阅览室里是否有座位；一是读者阅读完毕，离开阅览室，这时的操作要考虑阅览室里是否有读者。读者在阅览室读书时，由于没有引起资源的变动，不算动作变化。算法的信号量有三个： $seats$ ——表示阅览室是否有座位（初值为100，代表阅览室的空座位数）； $readers$ ——表示阅览室内的读者数，初值为0；用于互斥的 $mutex$ ，初值为1。

读者进入阅览室的动作描述

```
getin:
while(TRUE){
P(seats); /*没有座位则离开*/
P(mutex) /*进入临界区*/
填写登记表;
进入阅览室读书;
V(mutex) /*离开临界区*/
V(readers)
}
读者离开阅览室的动作描述
getout:
while(TRUE){
P(readers) /*阅览室是否有人读书*/
P(mutex) /*进入临界区*/
消掉登记;
离开阅览室;
V(mutex) /*离开临界区*/
V(seats) /*释放一个座位资源*/
}
```

7、解答：将独木桥的两个方向标记为 A 和 B，用变量 $countA$ 和 $countB$ 表示 A 和 B 方向上已经在独木桥上的行人数目，初值为 0；再设置三个互斥变量，初值都为 1： SA 实现对 $countA$ 互斥修改， SB 实现对 $countB$ 变量的互斥修改， $mutex$ 用来实现两个方向的行人对独木桥的互斥使用。

以 A 方向行人的动作为例，来描述行为如下：

```
P(SA);
If(countA=0) then P(mutex);
countA=countA+1;
```

```

S(SA);
通过独木桥
P(SA);
    countA=countA+1;
    If(countA=0) then V(mutex);
S(SA);

```

8、解答：当前的系统状态描述为：

$$C = \begin{bmatrix} 5 & 5 & 9 \\ 5 & 3 & 6 \\ 4 & 0 & 11 \\ 4 & 2 & 5 \\ 4 & 2 & 4 \end{bmatrix} \quad A = \begin{bmatrix} 2 & 1 & 2 \\ 4 & 0 & 2 \\ 4 & 0 & 5 \\ 2 & 0 & 4 \\ 3 & 1 & 4 \end{bmatrix} \quad C - A = \begin{bmatrix} 3 & 4 & 7 \\ 1 & 3 & 4 \\ 0 & 0 & 6 \\ 2 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$R = (17 \ 5 \ 20)$$

$$V = (2 \ 3 \ 3)$$

(1)

在 T0 时刻，由于 V (2, 3, 3) 大于等于 (C-A) 中 P5 所在行的向量 (1, 1, 0)，因此 V 能满足 P5 的运行，在 P5 运行后，系统的状态为：

$$A = \begin{bmatrix} 2 & 1 & 2 \\ 4 & 0 & 2 \\ 4 & 0 & 5 \\ 2 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix} \quad C - A = \begin{bmatrix} 3 & 4 & 7 \\ 1 & 3 & 4 \\ 0 & 0 & 6 \\ 2 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad V' = (5 \ 4 \ 7)$$

同样的，在 P5 运行后，V' (5, 4, 7) 也大于等于 C-A 中 P4 所在的行 (2, 2, 1)，则能满足 P4 的运行。P4 运行后，系统的状态为：

$$A = \begin{bmatrix} 2 & 1 & 2 \\ 4 & 0 & 2 \\ 4 & 0 & 5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad C - A = \begin{bmatrix} 3 & 4 & 7 \\ 1 & 3 & 4 \\ 0 & 0 & 6 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad V' = (7 \ 4 \ 11)$$

按照上述同样的方法，P4 运行后，P3, P2, P1 也能按顺序运行。(备注：考试时需要都写出来)。

因此，在 T0 时刻，存在安全序列：P5、P4、P3、P2、P1。

T0 时刻是安全的。

(2)

P2 申请资源 (0, 3, 4)，但在 C-A 中，P2 所在行向量是 (1, 3, 4)。对于资源 R1，P2 的申请超过它所预定的需求。因此，该申请不给予分配。

(3)

A) P4 申请 (2, 0, 1) 不超过 C-A 中 P4 所在行的向量 (2, 2, 1)。

B) V (2, 3, 3) 大于等于 P4 的申请 (2, 0, 1)

C) 对 P4 的申请 (2, 0, 1) 进行预分配，预分配后，系统的状态为：

$$A = \begin{bmatrix} 2 & 1 & 2 \\ 4 & 0 & 2 \\ 4 & 0 & 5 \\ 4 & 0 & 5 \\ 3 & 1 & 4 \end{bmatrix} \quad C - A = \begin{bmatrix} 3 & 4 & 7 \\ 1 & 3 & 4 \\ 0 & 0 & 6 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad V = (0 \ 3 \ 2)$$

可用资源 $V(0, 3, 2)$ 大于等于 $C-A$ 中 $P4$ 所在的行 $(0, 2, 0)$ ，因此可以满足 $P4$ 的运行。 $P4$ 运行后，系统的状态为：

$$A = \begin{bmatrix} 2 & 1 & 2 \\ 4 & 0 & 2 \\ 4 & 0 & 5 \\ 0 & 0 & 0 \\ 3 & 1 & 4 \end{bmatrix} \quad C - A = \begin{bmatrix} 3 & 4 & 7 \\ 1 & 3 & 4 \\ 0 & 0 & 6 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad V' = (4 \ 3 \ 7)$$

同样的方法（考试时需要列出），可计算出存在安全序列： $P4, P5, P3, P2, P1$ 。
因此，预分配后系统的状态是安全状态。

对于， $P4$ 请求资源 $(2, 0, 1)$ ，给予分配，分配后的系统新状态为：

$$C = \begin{bmatrix} 5 & 5 & 9 \\ 5 & 3 & 6 \\ 4 & 0 & 11 \\ 4 & 2 & 5 \\ 4 & 2 & 4 \end{bmatrix} \quad A = \begin{bmatrix} 2 & 1 & 2 \\ 4 & 0 & 2 \\ 4 & 0 & 5 \\ 4 & 0 & 5 \\ 3 & 1 & 4 \end{bmatrix} \quad C - A = \begin{bmatrix} 3 & 4 & 7 \\ 1 & 3 & 4 \\ 0 & 0 & 6 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

$$R = (17 \ 5 \ 20)$$

$$V = (0 \ 3 \ 2)$$

(4)

进程 $P1$ 请求资源 $(0, 2, 0)$

A) $P1$ 申请 $(0, 2, 0)$ 不超过 $C-A$ 中 $P1$ 所在行的向量 $(3, 4, 7)$ 。

B) $V(0, 3, 2)$ 大于等于 $P1$ 的申请 $(0, 2, 0)$

C) 对 $P1$ 的申请 $(0, 2, 0)$ 进行预分配，预分配后，系统的状态为：

$$A = \begin{bmatrix} 2 & 3 & 2 \\ 4 & 0 & 2 \\ 4 & 0 & 5 \\ 4 & 0 & 5 \\ 3 & 1 & 4 \end{bmatrix} \quad C - A = \begin{bmatrix} 3 & 2 & 7 \\ 1 & 3 & 4 \\ 0 & 0 & 6 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad V = (0 \ 1 \ 2)$$

$V(0, 2, 1)$ 不大于等于 $P1$ 到 $P5$ 任一进程在 $C-A$ 中的向量，因此系统进行预分配后处于不安全状态。

对于 $P1$ 申请资源 $(0, 2, 0)$ ，不给予分配。

■