多选题 第1题 2分

进程由（ ）组成

A 程序

B 数据

C TCB

D PCB

《 第2章 课后作业 》

多选题 第2题 2分

在操作系统中，进程是一个（ ）的基本单位。

A 同步互斥

B 资源分配

C 独立运行

D 调度

单选题 第3题 2分

在信号量机制中，信号量S > 0时的值表示（ ）；若S < 0，则表示（ ），此时进程应（ ）。

A 等待该资源的进程数、可用资源数目、阻塞

B 可用资源数目、等待该资源的进程数、阻塞

C 等待该资源的进程数、可用资源数目、就绪

D 可用资源数目、等待该资源的进程数、就绪

《 第2章 课后作业 》

多选题 第4题 3分

常用的进程通信方式有（ ）机制。

A 共享存储器

B 管道

C 消息

D 公用变量

《 第2章 课后作业 》 - 4/25页 -

单选题 第5题 2分

正在执行的进程等待I/O操作，其状态将由（ ）状态变为（ ）状态。

A 就绪、阻塞

B 就绪、执行

C 执行、阻塞

D 执行、就绪

单选题 第6题 1分

一次只允许一个进程访问的资源叫（ ）。

A 临界区

B 临界资源

C 信号量

D 互斥资源

多选题 第7题 2分

操作系统中，进程可以分为（ ）两类。

A 系统进程

B 内核进程

C 用户进程

D 接口进程

《 第2章 课后作业 》 - 7/25页 -

单选题 第8题 6分

试说明PCB的作用
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（）独立运行的程序，成为一个（）独立运行的基本单位，成为能与其它进程（）执行的进程。

A 不能、能、并行

B 不能、能、并发

C 能、不能、并行

D 能、不能、并发

单选题 第9题 2分

为什么说PCB是进程存在的惟一标志?
答:
OS是根据PCB对（ ）执行的（ ）进行控制和管理的。

A 并行、进程

B 并行、线程

C 并发、进程

D 并发、线程

单选题 第10题 2分

就绪状态→执行状态的原因是：

A 进程分配到CPU资源

B 时间片用完

C I/O请求

D I/O完成

《 第2章 课后作业 》

单选题 第11题 2分

执行状态→就绪状态的原因是：

A 进程分配到CPU资源

B 时间片用完

C I/O请求

D I/O完成

单选题 第12题 2分

执行状态→阻塞状态的原因是：

A 进程分配到CPU资源

B 时间片用完

C I/O请求

D I/O完成

《 第2章 课后作业 》

单选题 第13题 2分

阻塞状态→就绪状态的原因是：

A 进程分配到CPU资源

B 时间片用完

C I/O请求

D I/O完成

《 第2章 课后作业 》

多选题 第14题 8分

在进行进程切换时，所要保存的处理机状态信息有哪些?

A 进程当前暂存信息

B 下一指令地址信息

C 进程状态信息

D 过程和系统调用参数及调用地址信息

《 第2章 课后作业 》

多选题 第15题 8分

试说明引起进程创建的主要事件

A 用户登录

B 作业调度

C 提供服务

D 应用请求

《 第2章 课后作业 》 - 15/25页 -

多选题 第16题 8分

在创建一个进程时所要完成的主要工作是什么?

A 调用进程创建原语Creat()

B 申请空白PCB

C 为新进程分配资源

D 初始化进程控制块

E 将新进程插入就绪队列

《 第2章 课后作业 》 - 16/25页 -

单选题 第17题 2分

试从调度性方面，对进程和线程进行比较。
答：
在传统OS中，（）是资源拥有的基本单位，也是独立调度和分派的基本单位；在引入线程的OS中，（）是独立调度和分派的基本单位，（）只是拥有资源的基本单位，两个角色分开。

A 进程、线程、进程

B 进程、进程、线程

C 线程、线程、进程

D 线程、进程、线程

《 第2章 课后作业 》

单选题 第18题 2分

试从并发性方面，对进程和线程进行比较。
答：
在引入线程的OS中，不仅（）间可以并发执行，而且一个（）内的多（）也可以并发执行，因而比传统的OS具有更好的并发性

A 进程、线程、进程

B 进程、进程、线程

C 线程、线程、进程

D 线程、进程、线程

《 第2章 课后作业 》

单选题 第19题 2分

试从拥有资源方面，对进程和线程进行比较。
答：
在各类OS中，拥有资源的基本单位都是（），（）除了一点简单资源外，本身基本不拥有系统资源，但它可以共享所属（）的资源。

A 进程、线程、进程

B 进程、进程、线程

C 线程、线程、进程

D 线程、进程、线程

《 第2章 课后作业 》

单选题 第20题 2分

试从系统开销方面，对进程和线程进行比较。
答：
创建和撤销（），由于要分配和回收系统资源，因此系统开销远大于创建和撤销（）；（）切换要保存和设置的现场信息也较多，所以进程切换的系统开销大于（）切换。

A 线程、进程，线程、进程

B 线程、进程，进程、线程

C 进程、线程，线程、进程

D 进程、线程，进程、线程

《 第2章 课后作业 》 - 20/25页 -

单选题 第21题 4分

何谓用户级线程?
答:
用户级线程：仅存在于用户空间中的线程，（）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（）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（）内核支持。

A 无需、需要、需要

B 无需、无需、无需

C 需要、需要、需要

D 需要、无需、无需

《 第2章 课后作业 》 - 21/25页 -

单选题 第22题 4分

何谓内核支持线程?
答:
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，还是系统线程中的线程，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（），内核根据该控制块感知某线程的存在并实施控制。

A 需要、PCB

B 无需、PCB

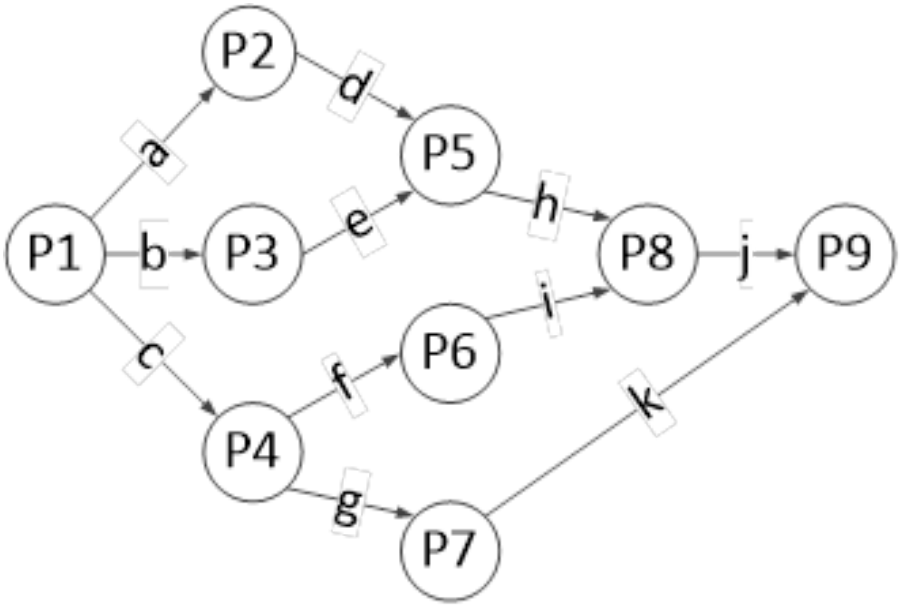
C 需要、TCB

D 无需、TCB

《 第2章 课后作业 》 - 22/25页 -

填空题 第23题 13分

完成程序，用来描述下图所示的前驱图：




```
Var a, b, c, d, e, f, g, h, I, j, k ; semaphore:=
[填空1] , [填空2] , [填空3] , 0, 0, 0, 0, 0, 0, [填
空4] , [填空5] ;
begin
parbegin
      begin P1; [填空6] (a); [填空7] (b); [填空8]
      (c);end;
      begin [填空9] (a); P2; signal(d); end;
      begin [填空10] (b); P3; signal(e); end;
      begin [填空11] (c); P4; signal(f);
      signal(g); end;
      begin wait(d); wait(e); P5; signal(h); end;
      begin wait(f); P6; signal(i); end;
      begin wait(g); P7; signal(j); end;
      begin wait(h); wait(i); P8; signal(k); end;
      begin [填空12] (j); [填空13] (k); P9; end;
parend
end
```

《 第2章 课后作业 》 - 23/25页 -

填空题 第24题 11分

在测量控制系统中的数据采集任务，把所采集的数据送至一单缓冲区；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。(方法1)

```
Var mutex, empty, full: semaphore:= [填空1] ,
[填空2] , [填空3] ;
gather:
begin
      repeat
            ......
            gather data in nextp;
             [填空4] (empty);
             [填空5] (mutex);
            buffer:=nextp;
             [填空6] (mutex);
             [填空7] (full);
      until false;
end
compute:
begin
      repeat
            ......
             [填空8] (full);
             [填空9] (mutex);
            nextc:=buffer;
             [填空10] (mutex);
             [填空11] (empty);
            compute data in nextc;
      until false;
end
```

《 第2章 课后作业 》 - 24/25页 -

填空题 第25题 6分

在测量控制系统中的数据采集任务，把所采集的数据送至一单缓冲区；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。(方法2)

```
Var empty, full: semaphore:= [填空1] , [填空2] ;
gather:
begin
      repeat
            ......
            gather data in nextp;
            [填空3] (empty);
            buffer:=nextp;
            [填空4] (full);
      until false;
end
compute:
begin
      repeat
            ......
            [填空5] (full);
            nextc:=buffer;
            [填空6] (empty);
            compute data in nextc;
      until false;
end
```