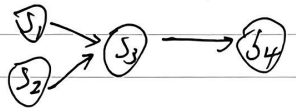
操作系统 第二章

1. 什么是前趋图？请画出下列语句的前趋图

S_1: a=x+y　S_2: b=z+1　S_3: c=a-b　S_4: w=c+1

前趋图：有向无环图，用于描述进程间执行的前后关系



2. 什么是进程？OS为什么要引入进程？它会产生什么样的影响？

进程是一段可并发执行的具有独立功能的程序，是关于某一个数据集的一次执行过程

在OS中引入进程是为了实现多个程序的并发执行。传统程序与其他程序并发执行时，执行结果不可再现，因此传统程序不能与其它程序并发执行。

建立进程所带来的好处是多个程序能并发执行，这极大地提高了资源利用率和系统吞吐量但同时也付出了内存的空间开销，调用进程的时间开销等代价。

3. 进程最基本的状态有哪些？哪些事件可能会引起不同状态的转换

①运行态 ②就绪态 ③等待态

运行态→等待态：等待使用资源或某事件的发生

等待态→就绪态：资源得到满足或某事件已经发生

就绪态→运行态：CPU空闲时调度选中一个就绪进程需要其运行

运行态→就绪态：运行时间片到达或出现有更高优先级进程

6. 请给出PCB的主要内容。描述当前进程~~的运行~~状态发生转换时，OS需要使用/修改PCB的哪些内容（就绪→运行，运行→就绪）

PCB的内容可分为调度信息和现场信息两部分，调度信息供进程调度时使用，现场信息用于保留进程发生状态转换时所要保留的CPU现场信息。

① 就绪→运行是将PCB由当前"就绪态"改为"运行态"，修改PCB队列指针，将PCB从就绪队列中移出，利用PCB中的CPU现场信息，~~以保证该进程~~恢复CPU现场并投入运行

② 运行→就绪是将CPU的当前状态保存到PCB中，将进程状态由"运行"改为就绪

No

DATE

11. 为什么要在OS中引入线程

在OS中引入线程是为了减少进行并发执行时所付出的时空开销，使OS有更好的并发性提高CPU利用率，能使系统适应新的对称多处理机环境，充分发挥其性能

21. 试从调度、并发、拥有资源和系统开销四方面比较传统进程和线程

	调度	并发	拥有资源	系统开销
线程	线程是OS中调度和分派的基本单位具有唯一的标识符和TCB具有独立的虚地址空间以传统进程为单位进行任务调动时，系统必须切换地址空间，且切换时间长	同一进程的的线程可在个/的处理机上并发执行多任务OS会限制用户，通常不的多	同一进程的有线程共享但每个拥有进程的状态和资源可以访问相同数据，通信方便	小，同一进程的的线程共享同一地址空间，线程快速切换对的个传统进的进行管理时，系统开销大
传统进程			传统进程是系统中资源分配和保护的基本单位也是系统调度的独立单位，每个传统进程都的以各自为独的进程在CPU运行	

No.
DATE / /

操作系统 第四章

1. 什么是临界资源？什么是临界区？
一次仅允许一个进程使用的资源称为临界资源。
进程中访问临界资源的那段代码称为临界区

2. 同步机制应遵循的准则有哪些？
(1) 空闲让进。当无进程处于临界区时，表明临界资源处于空闲状态，应允许一个请求进入临界区的进程立即进入临界区，以有效地利用临界资源。
(2) 住则等待。当已有进程进临界区时，表明临界资源正在被访问，因而其他试图进入临界区的进程必须等待，以保证对临界资源的互斥访问
(3) 有限等待。对要求访问临界资源的进程，应保证其能在有限时间内进入临界区，以免陷入死等"状态
(4) 让权等待。当进程不能进入临界区时，应立即释放处理机，以免进程陷入"忙等"

14. 有m个进程共享同一临界资源，若使用信号量机制实现对某个临界资源的互斥访问，请问信号量的变化范围
初值/max为1，每次-1，最小为1-m
范围[1-m,1]

20. 桌上有一个能盛得下5个水果的空盘子。爸爸不停地向盘中放苹果或橘子，儿子云停从盘子中取橘子，女儿云停从盘子中取苹果。3人不能同时向(从)盘子中放(取)水果
试用信号量来实现爸爸、儿子和女儿这3个"循环进程"之间的同步

```
semaphore empty=5, orange=0, apple=0, mutex=1
Dad(){                  Son(){                  Daughter(){
  while(1){               while(1){               while(1){
    P(empty);               P(orange);              P(apple);
    P(mutex);<put(fruit);   P(mutex);               P(mutex);
    V(mutex);               put(fruit);             put(fruit);
    if(fruit==orange)       V(mutex);               V(mutex);
      V(orange);            V(empty);               V(empty);
    else V(apple)           enjoy(orange);          enjoy(apple);
  }                       }                       }
}                       }                       }.
```