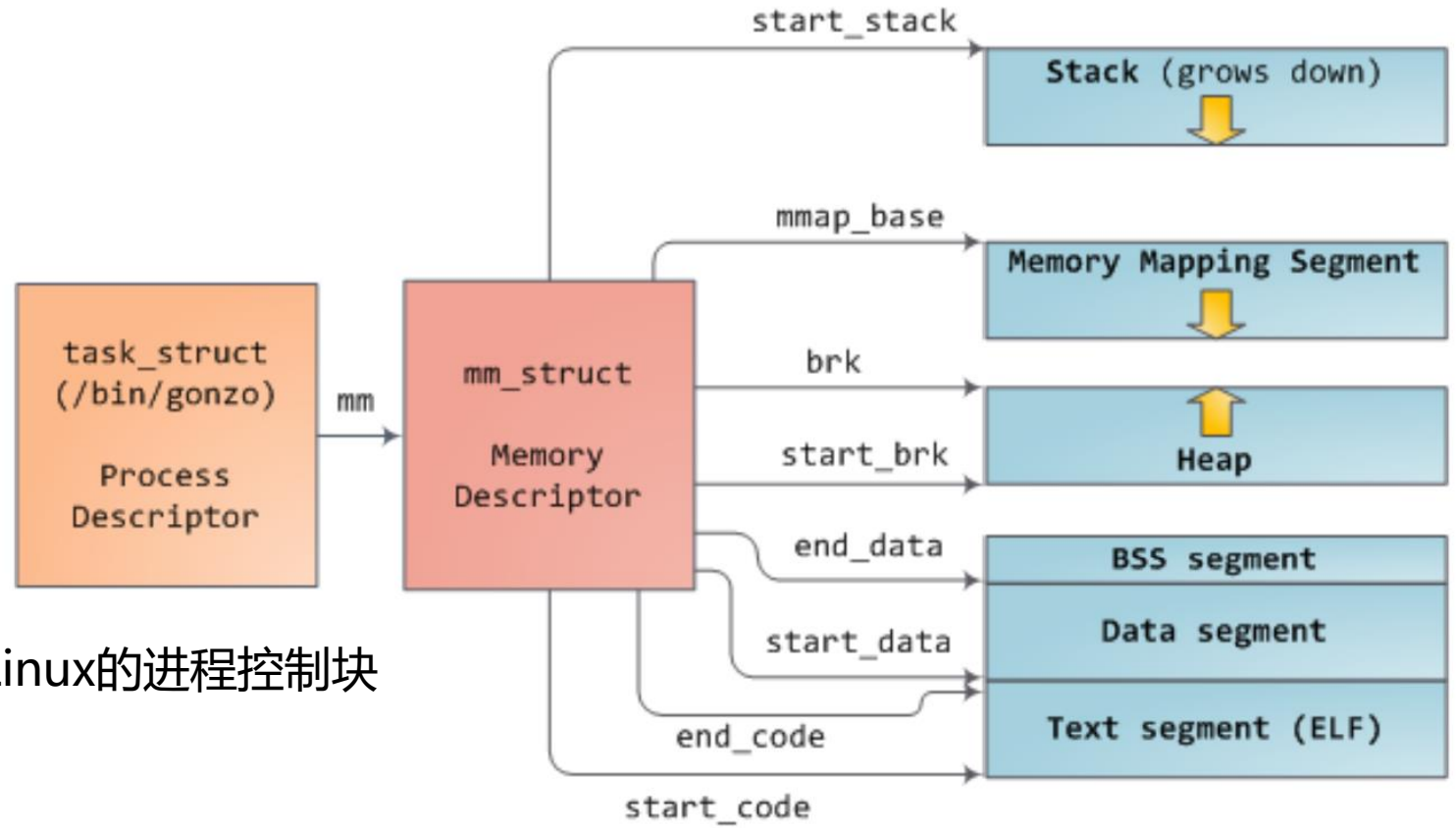
操作系统

L17 进程和内存习题讲解

胡燕
大连理工大学 软件学院

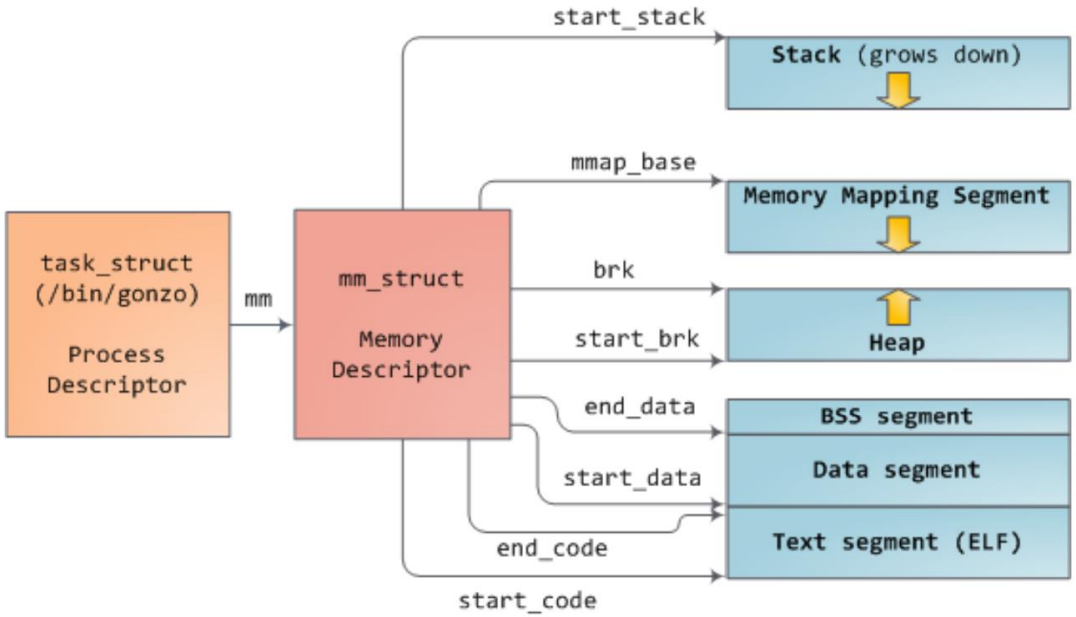
进程概念

进程 Process

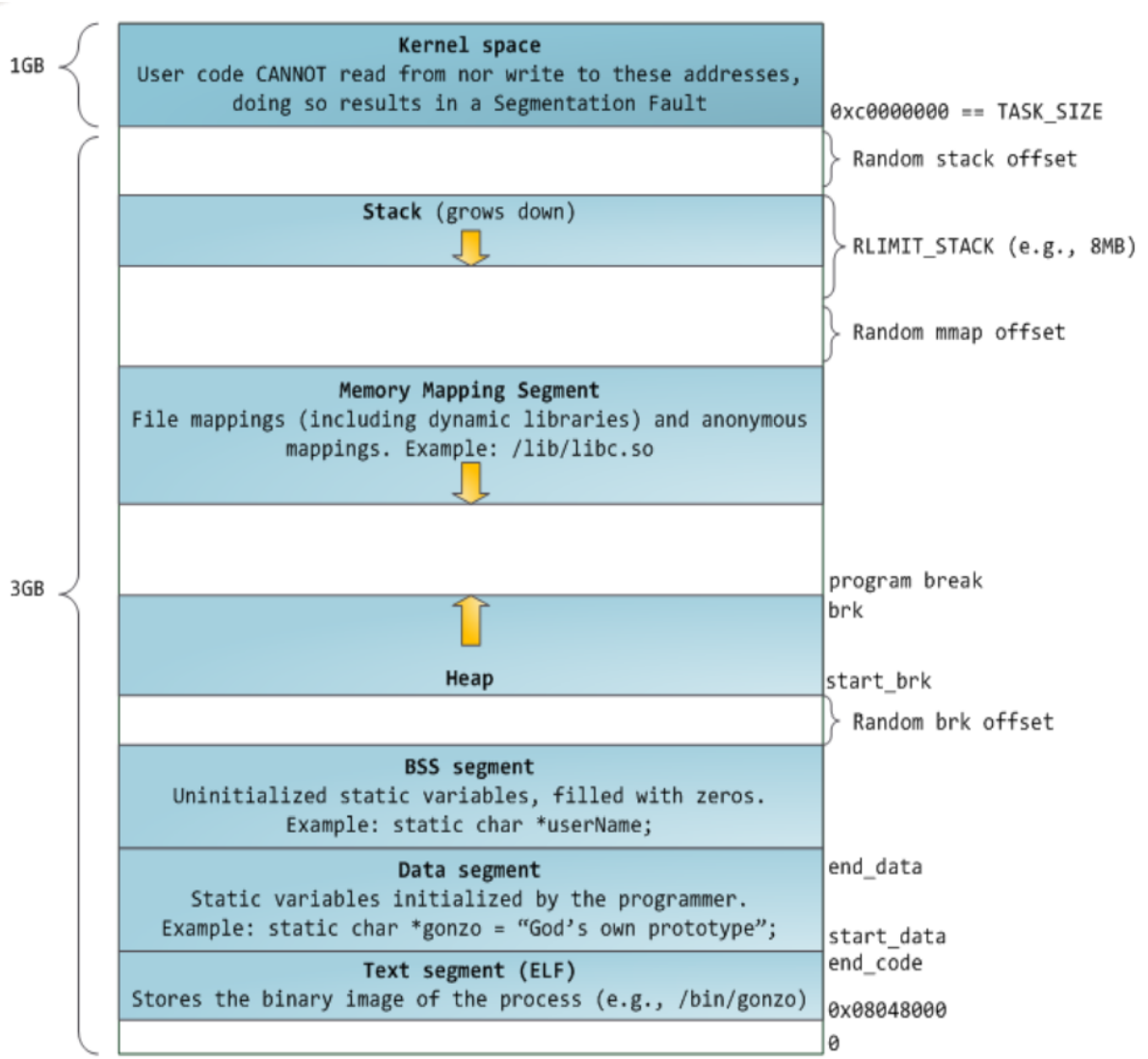


Linux的进程控制块

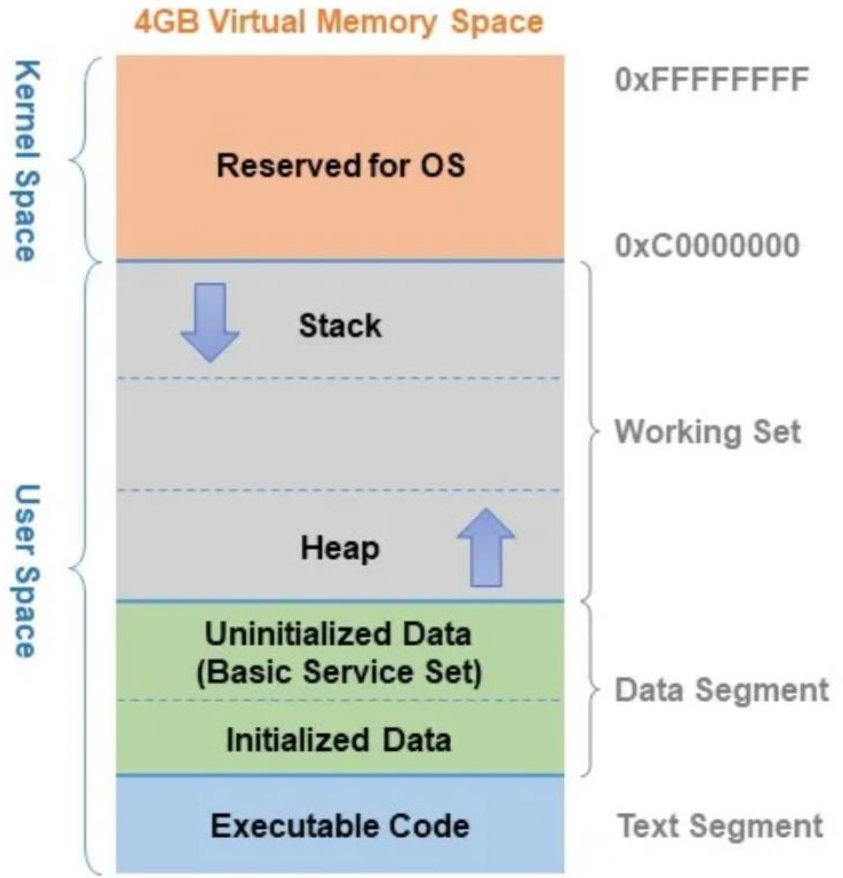
进程概念



进程地址空间映像



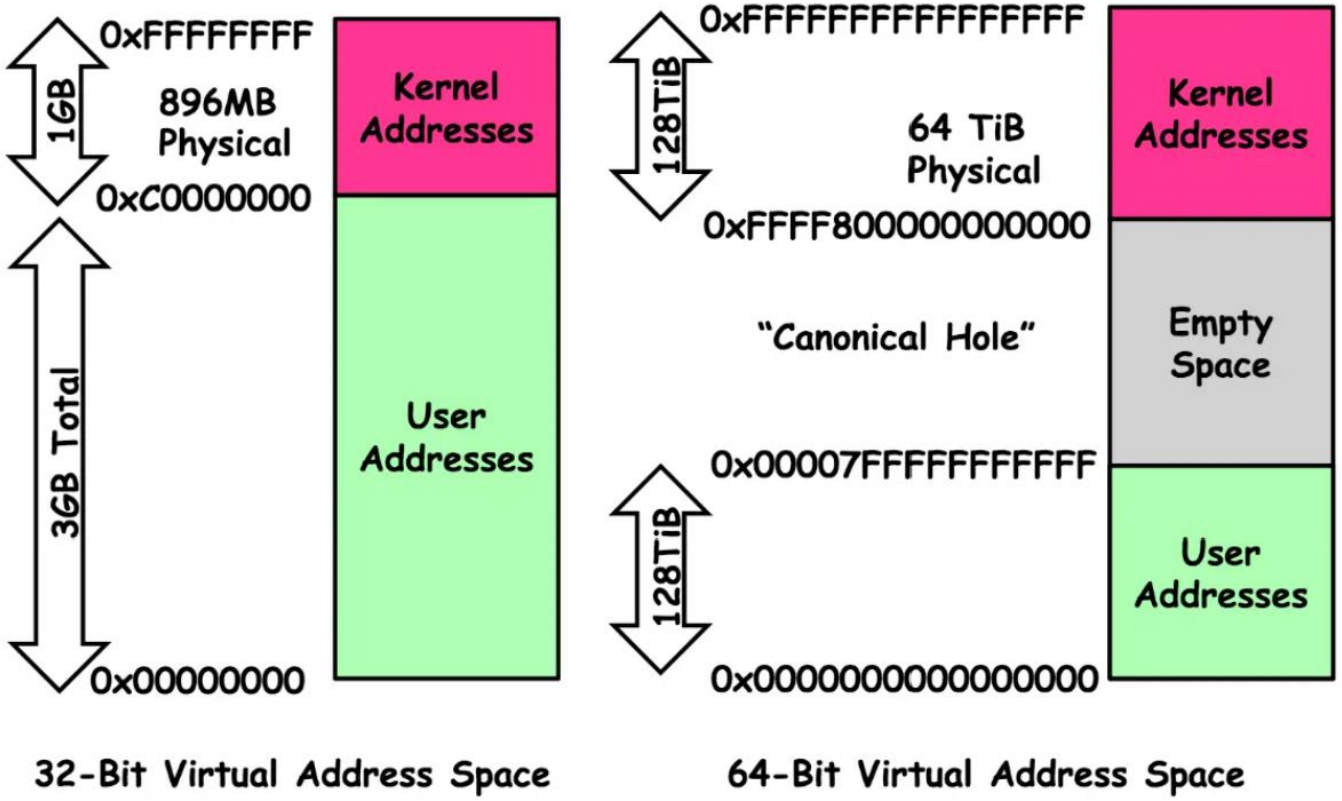
进程概念




```c
#include <stdio.h>

char gMessage[] = "Initialized global variable stored in Initialized Data Segment";
const char constMessage[] = "Constant global variable stored in Initialized Data Segment";
int gSessionCouter; /* Global variable stored in uninitialized BSS */

int main()
{
    static int sCounter = 0;  /* static variable stored in Initialized Data Segment */
    char *pBuffer = (char*)malloc(1024 * sizeof(char)); /* Memory allocated in Heap Segment */
    int sum = add(1,2);
    printf("%d", sum);
    return 0;
}

/*
** Variables (result, a, and b) and the return address
** of the "add" function are stored in the Stack.
*/
int add(int a, int b)
{
  int result;
  result = a + b;
  return result;
}
```

https://wentzwu.com/2020/04/11/cissp-practice-questions-20200412/

进程概念


0xFFFFFFFF
1GB
896MB
Physical
0xC0000000
3GB Total
0x00000000
Kernel
Addresses
User
Addresses
32-Bit Virtual Address Space
0xFFFFFFFFFFFFFFFF
128TiB
64 TiB
Physical
0xFFFF800000000000
"Canonical Hole"
0x00007FFFFFFFFFFF
128TiB
0x0000000000000000
Kernel
Addresses
Empty
Space
User
Addresses
64-Bit Virtual Address Space

进程概念

进程：程序在给定数据集上的一次执行

程序的执行

顺序执行：顺序性 —— 间断性；封闭性 —— 失去封闭性；可再现性 —— 不可再现性；并发执行

P_1:

```
if(x>=100){
    x-=100;
}
```

P_2:

```
if(x>=100){
    x-=100;
}
```

x是进程P1和P2间的共享变量

进程P1
if(x>=100);
进程P1
x-=100;
进程P2
If(x>=100)
进程P2
x-=100;

进程P1
if(x>=100);
进程P2
If(x>=100)
进程P1
x-=100;
进程P2
x-=100;

进程概念

进程特征

- 动态 → 程序在给定输入下的一次执行
- 并发 → 多个进程交替执行
- 独立 → 每个进程有自己的私有进程地址空间
- 异步 → 进程执行有自己的步调
- 结构性 → 进程有自己的结构

单选题 1分

设置

并发进程指的是（ ）。

A 可并行执行的进程

B 可先后执行的进程

C 可同时执行的进程

D 不可中断的进程

提交

练习

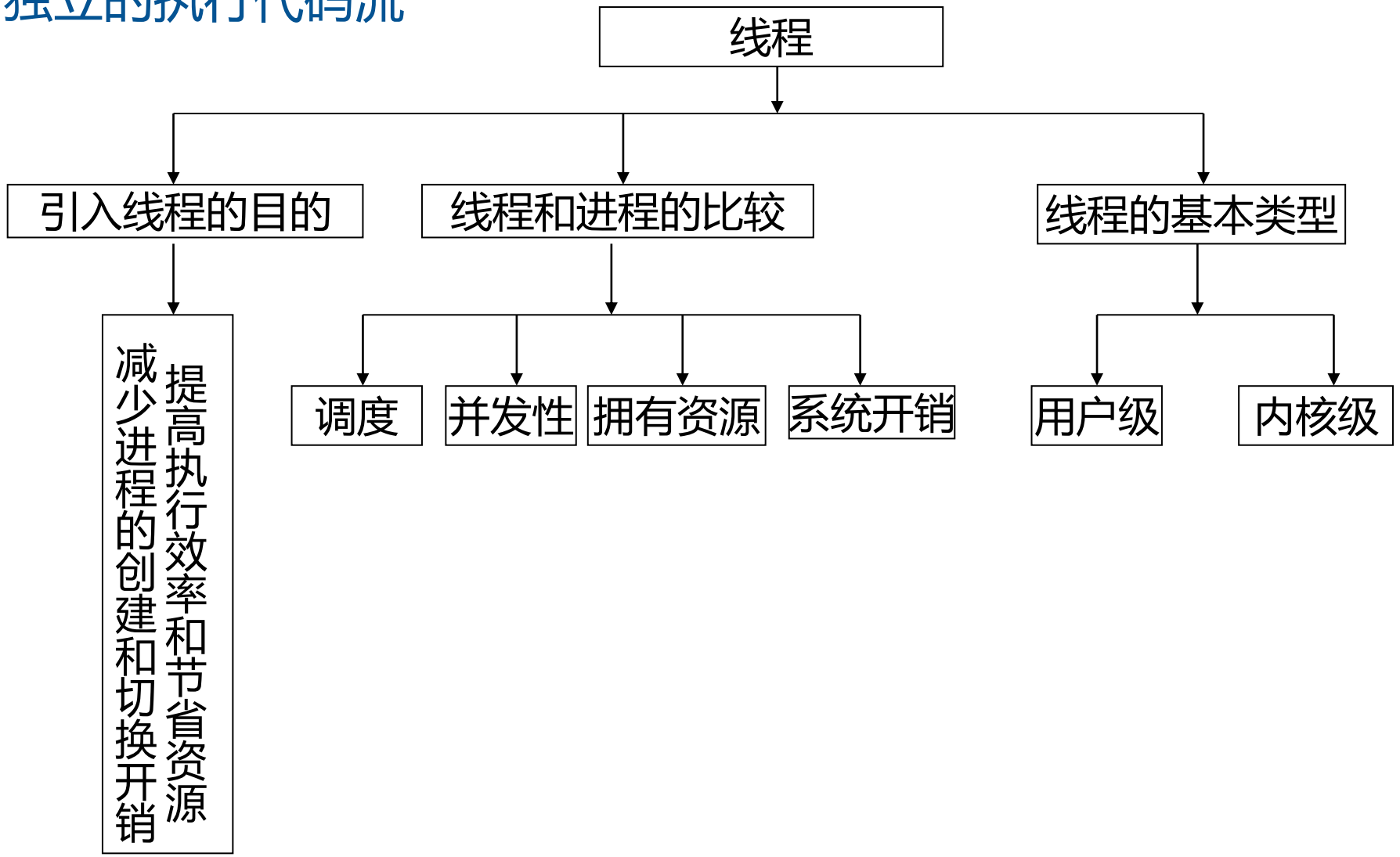
- 试说明进程在3个基本状态之间转换的典型原因。
  - 就绪到执行：CPU调度；
  - 执行到就绪：当前执行的进程时间片用完，或是被更高优先级的进程抢占了CPU。
  - 执行到阻塞：等待某种事件的发生（例如，进行I/O，需要等待I/O完成）。
  - 阻塞到就绪：进程所等待的事件发生。

练习

- 当发生状态转移时，操作系统完成哪些操作?
  - 就绪到执行：为进程分配CPU，并把进程切换到CPU上执行。
  - 执行到就绪：把进程从CPU撤下来，加入到就绪队列。
  - 执行到阻塞：把进程从CPU撤下来，状态改为阻塞，并加入到阻塞队列，等待重新调度。
  - 阻塞到就绪：把进程状态改为就绪，并把其PCB结构从阻塞队列中撤下来，加入就绪队列；最后返回原进程继续执行或进行进程调度

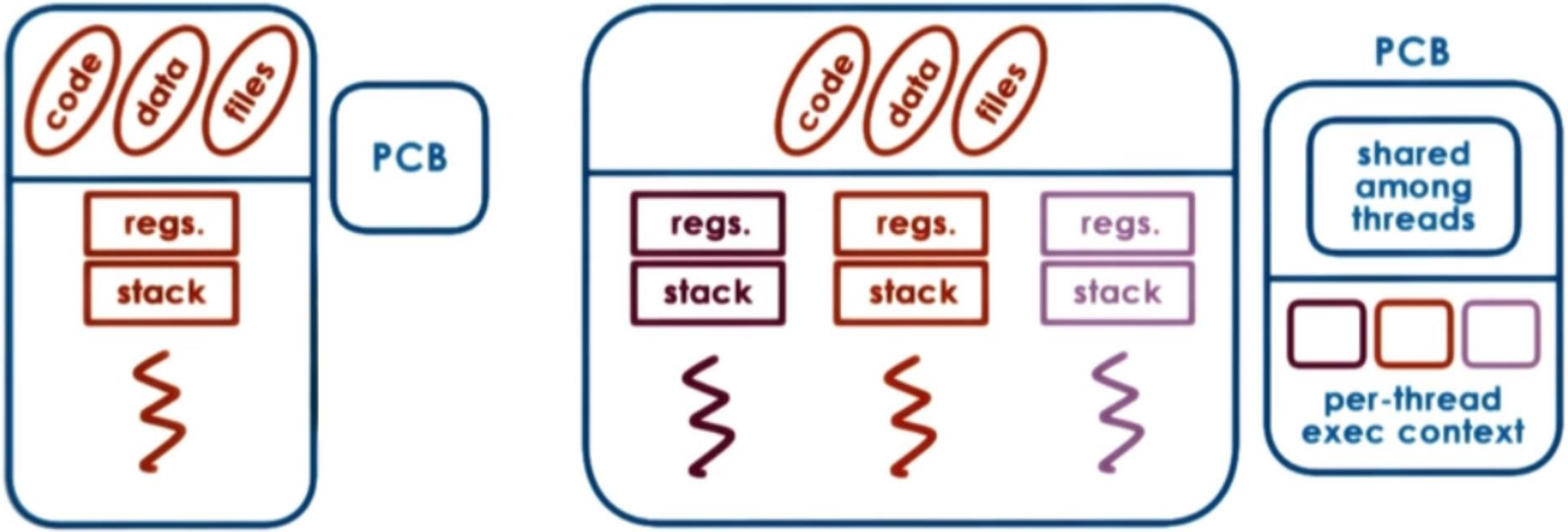
线程概念

线程：独立的执行代码流



线程概念

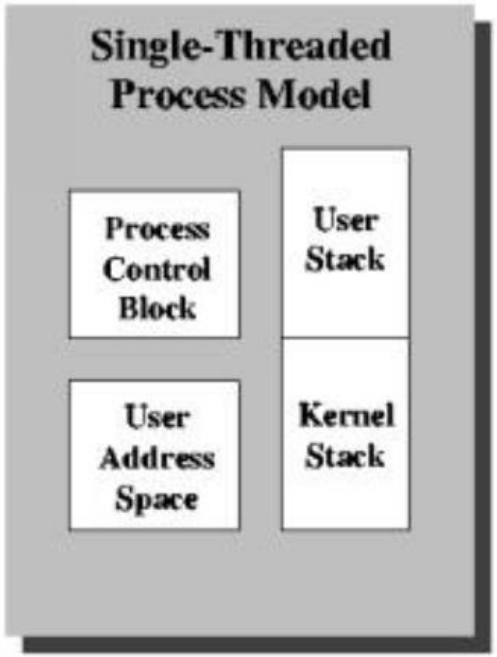
进程与线程的关系

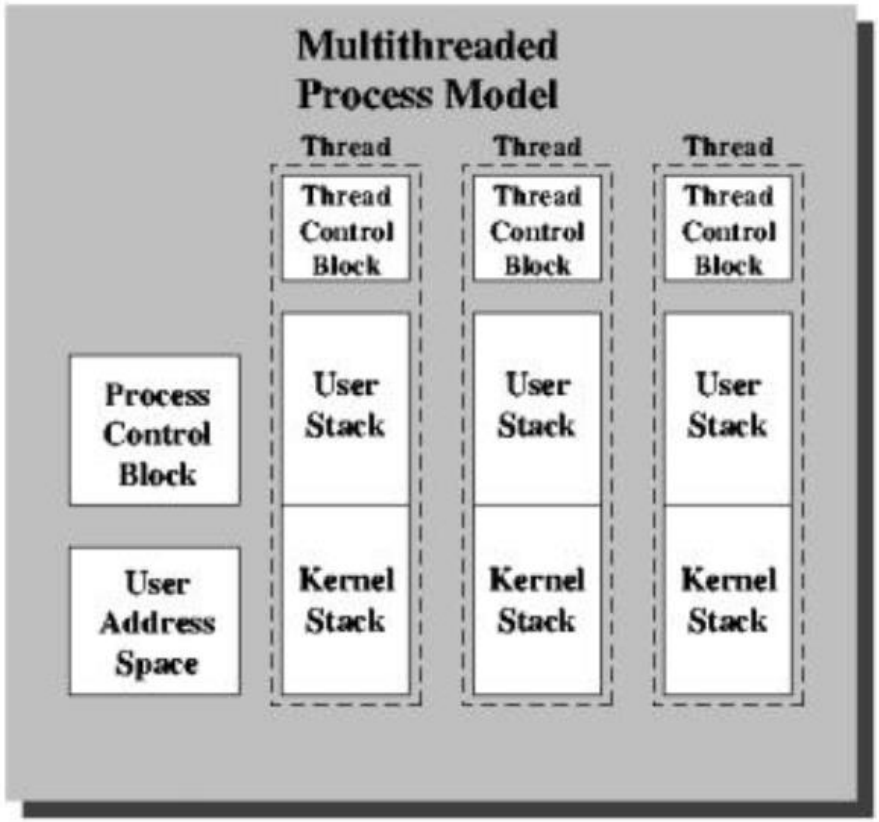


https://applied-programming.github.io/Operating-Systems-Notes/3-Threads-and-Concurrency/

线程概念

进程与线程的关系





https://www.cs.fsu.edu/~baker/opsys/notes/threads.html

线程概念

练习 程序、进程和线程的区别?

- **程序**
  静态概念。
- **进程**
  动态概念（有生命周期）；操作系统内进行资源分配的基本单位。
- **线程**
  动态概念（有生命周期）；调度与分派的基本单位。

线程概念

练习 用户级线程与内核级线程有何区别?

➢ 线程调度与切换时间

用户级线程的切换通常发生在用户态的同一进程的多个线程之间，无须通过中断进入OS的内核，且切换规则也简单，因此其切换速度很快。

而核心级线程的切换时间相对比较慢。

➢ 阻塞型系统调用的影响

用户级线程调用系统调用时，内核不知道用户级线程的存在，会被当作整个进程行为，使进程等待并调度另一个进程执行，在内核完成系统调用而返回时，进程才能继续执行。

而核心级线程则以线程为单位进行调度，当线程调用系统调用时，内核将其视为线程的行为，因此阻塞该线程，可以调度该进程中的其他线程执行

➢ 线程执行时间

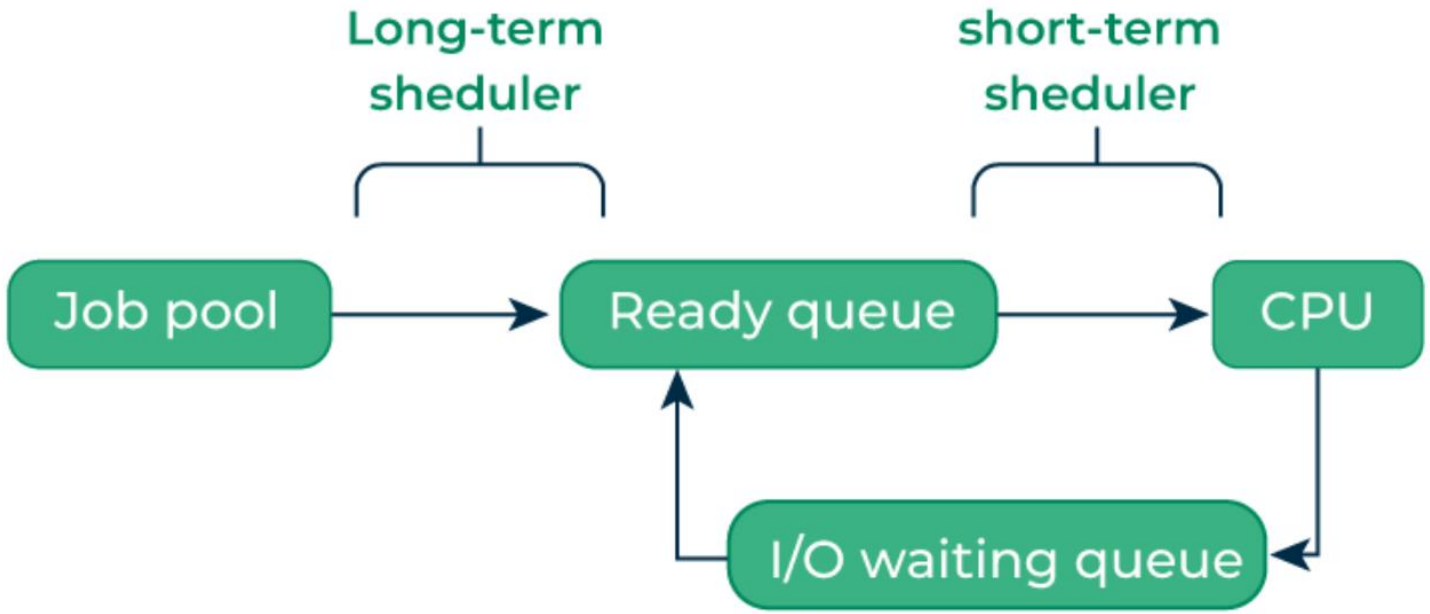
如果用户设置了用户级线程，系统调度是以进程为单位进行，但随着进程中线程数目的增加，每个线程得到的执行时间就少。而如果设置的是核心级线程，则调度以线程为单位，因此可以获得更多的执行时间

线程概念

练习 进程和线程的主要区别?

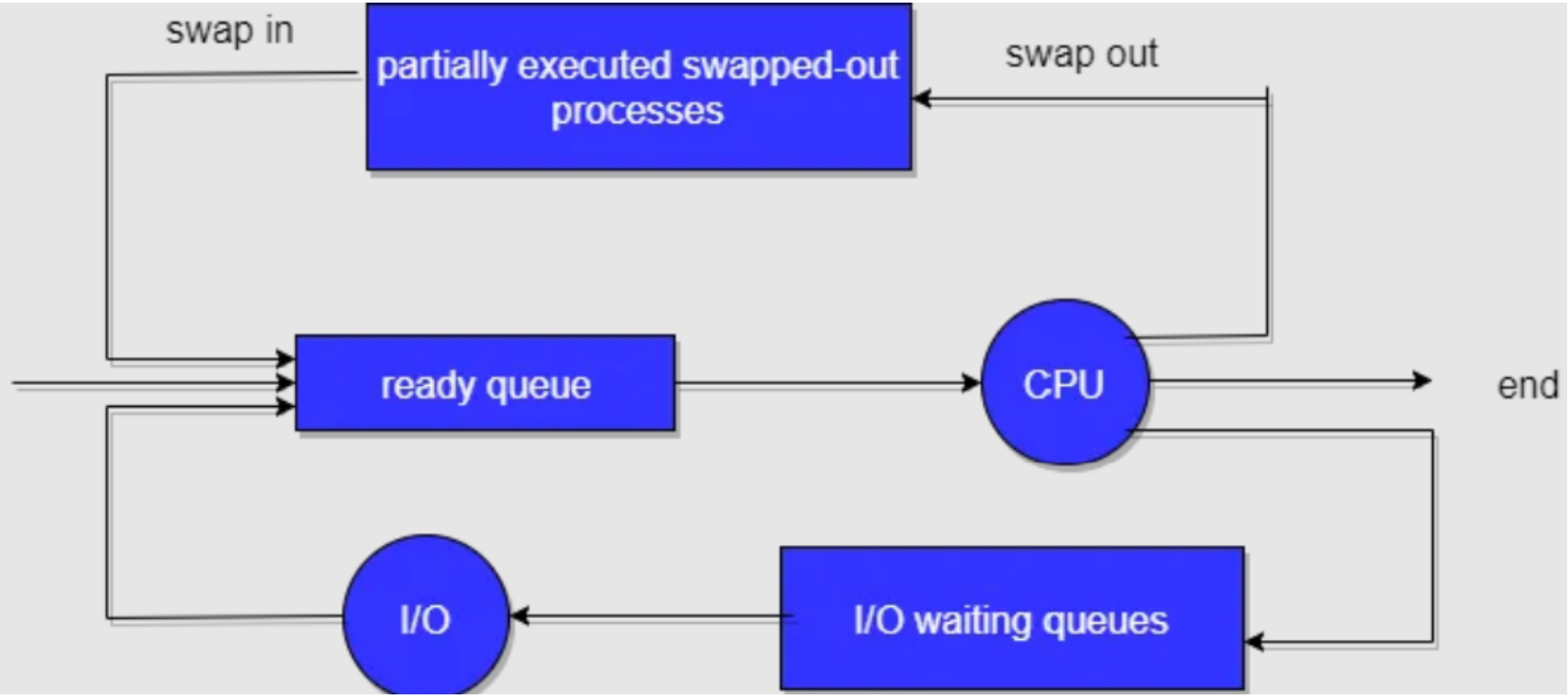
- **调度**
  线程作为调度的基本单位。
- **并发性**
  线程能够支持更细粒度的并发。
- **拥有资源**
  进程拥有资源。
- **系统开销**
  线程切换开销要小于进程切换开销。

长程调度与短程调度



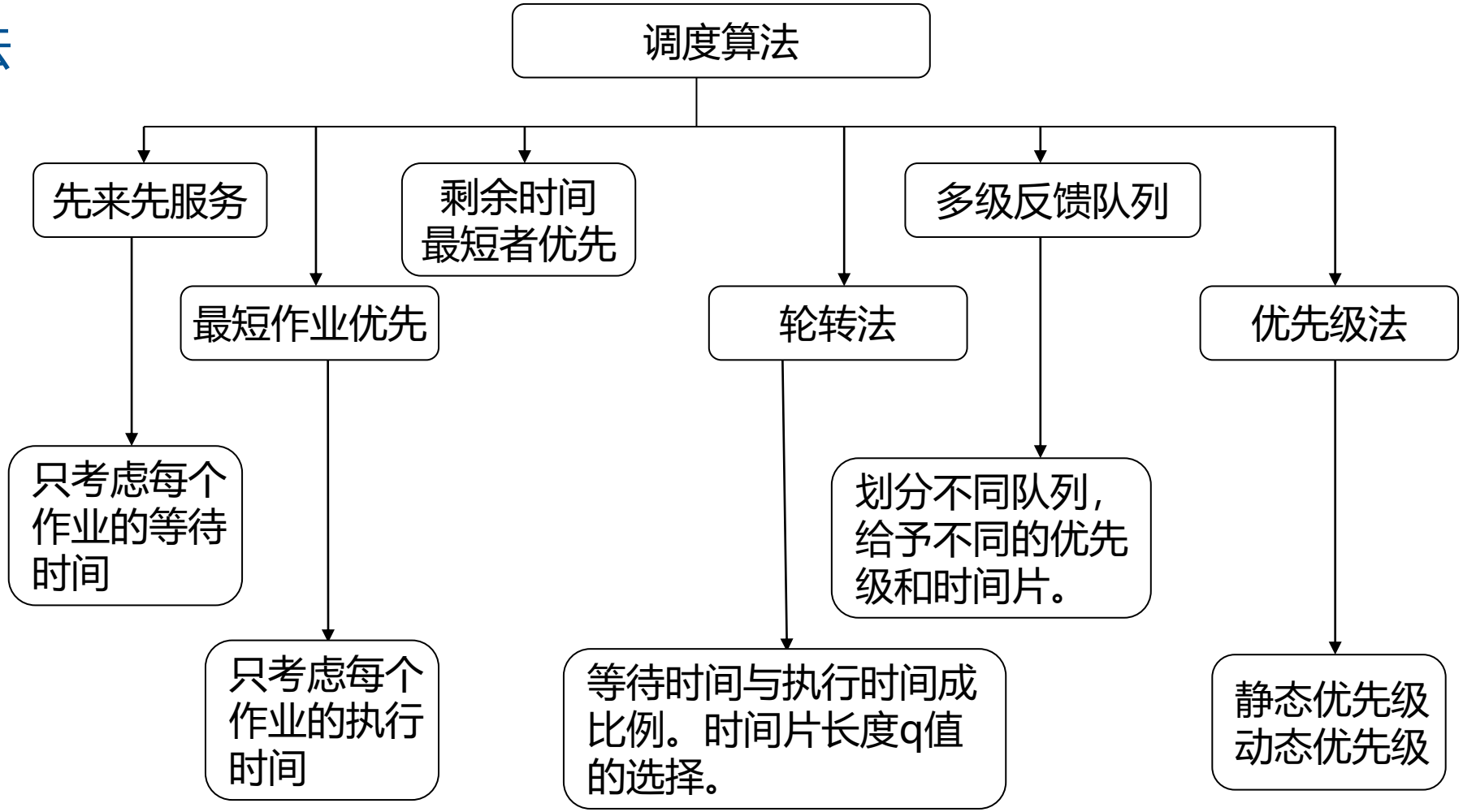
https://www.geeksforgeeks.org/process-schedulers-in-operating-system/

中程调度



https://ravindrabhati919.medium.com/process-scheduling-3a0240efef18

调度算法



性能衡量指标：平均等待时间、周转时间、带权周转时间、响应时间

练习

有5个进程P1,P2,P3,P4,P5, 它们同时依次进入就绪队列，它们的优先数和执行所需的处理器时间如表所示。忽略进程调度过程中的相关延迟，请回答下列问题：

（1）采用先来先服务、非抢占式优先级调度的调度顺序；

（2）计算各进程在就绪队列中等待的时间，以及平均等待时间。

进程	处理器时间	优先数
P1	10	4
P2	1	1
P3	2	3
P4	1	4
P5	5	2

练习

进程的上下文包括如下各项，除了（ ）。

A、用户打开文件表　　B、PCB

C、中断向量　　D、核心栈

练习

下列关于处理器调度功能描述，错误的是（ ）。

A、长程调度的主要功能是根据某种算法从外存后备队列中选择合适的作业（Job）调度内存，并为其创建进程
B、中程调度的主要功能是当内存紧张时挂起部分暂时不运行的进程，并在内存有空闲时激活部分被挂起的进程，以提高内存利用效率
C、进程调度的主要功能是根据某种算法从就绪队列中选择合适的进程调度到处理器运行
D、长程调度是调度层次中最基本、最高级的调度，所有类型的操作系统都必须配备

练习

下列关于处理器调度功能描述，错误的是（ ）。

A、长程调度的主要功能是根据某种算法从外存后备队列中选择合适的作业（Job）调度内存，并为其创建进程

B、中程调度的主要功能是当内存紧张时挂起部分暂时不运行的进程，并在内存有空闲时激活部分被挂起的进程，以提高内存利用效率

C、进程调度的主要功能是根据某种算法从就绪队列中选择合适的进程调度到处理器运行

D、长程调度是调度层次中最基本、最高级的调度，所有类型的操作系统都必须配备

练习

进程的相关时间参数如下表。采用轮转调度算法，时间片大小=2ms。那么请给出其调度的甘特图，并计算进程的平均等待时间。

Process	Burst Time	Arrival Time
P1	5 ms	0 ms
P2	4 ms	1 ms
P3	2 ms	2 ms
P4	1 ms	4 ms

主观题 10分

设置

进程的相关时间参数如下表。采用轮转调度算法，时间片大小=2ms。那么请给出其调度的甘特图，并计算进程的平均等待时间。

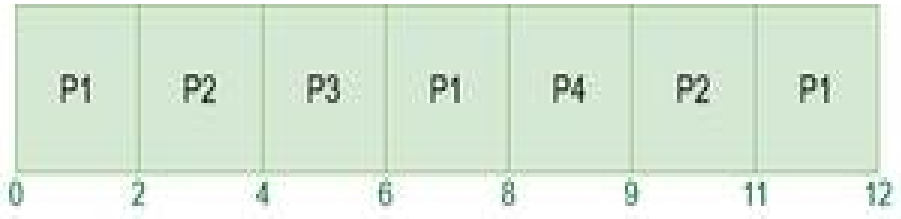
Process	Burst Time	Arrival Time
P1	5 ms	0 ms
P2	4 ms	1 ms
P3	2 ms	2 ms
P4	1 ms	4 ms

作答

练习 进程的相关时间参数如下表。采用轮转调度算法，时间片大小=2ms。那么请给出其调度的甘特图，并计算进程的平均等待时间。

Process	Burst Time	Arrival Time
P1	5 ms	0 ms
P2	4 ms	1 ms
P3	2 ms	2 ms
P4	1 ms	4 ms

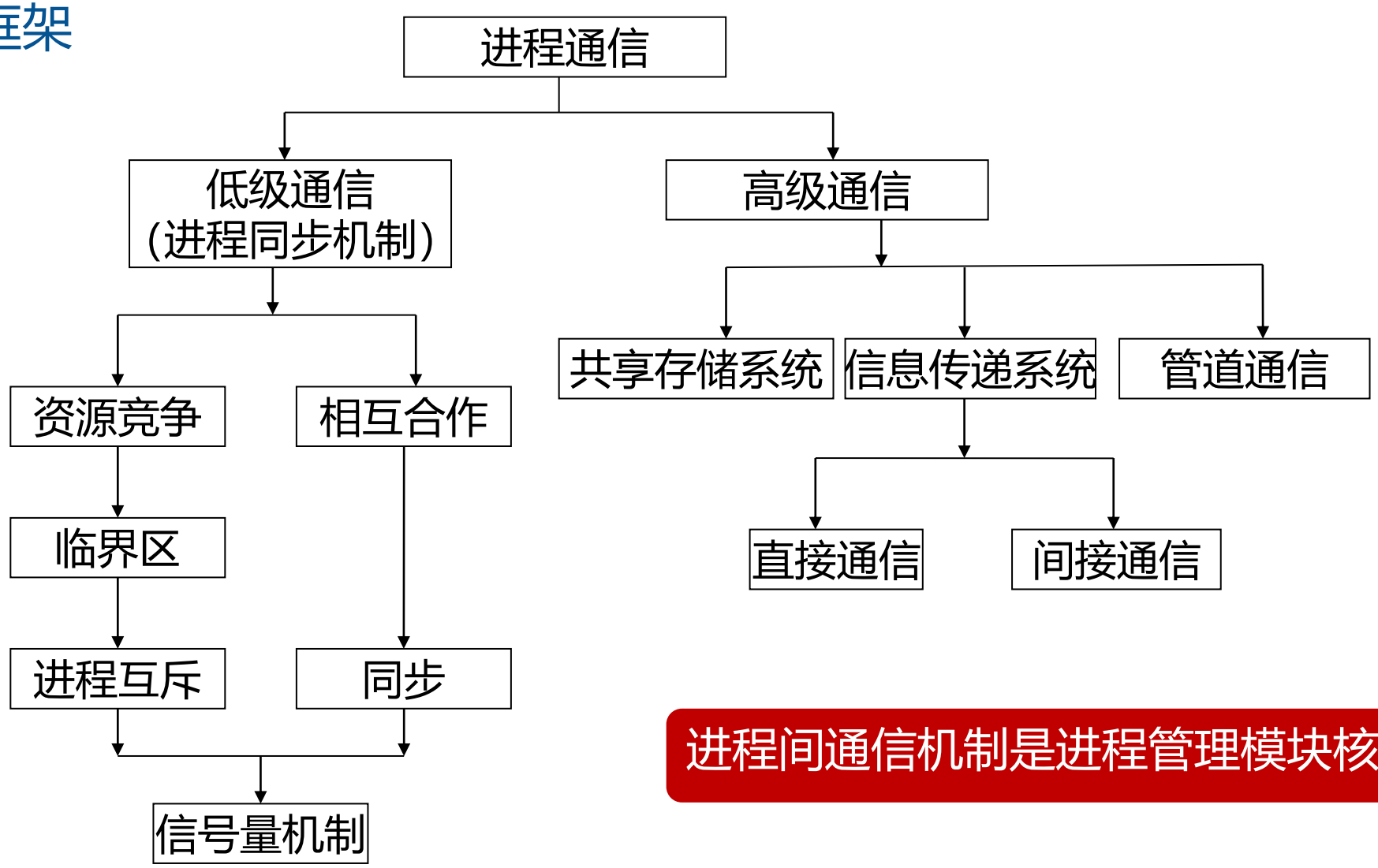
答: 甘特图如下：



平均等待= ((12-5) + (11-1-4) + (4-2) + (9-4-1)) / 4 = 19/4 = 4.75 ms

进程间通信

IPC概念框架



进程间通信机制是进程管理模块核心内容

练习

在多道程序并发环境下，进程之间是否完全独立运行?

答：多个逻辑上相对独立的进程在多道程序环境下并发运行时，他们之间并不是绝对独立的，相互之间由于资源竞争和进程合作存在着两种制约关系：
(1) 直接制约关系
(2) 间接相互制约

单选题 1分

设置

处于临界区中的进程不可被中断。这句话是否正确?

A 正确

B 错误

提交

练习

处于临界区中的进程不可被中断。这句话是否正确?

答：否。

进程进入临界区，表明进程正在访问某个临界资源，即不允许其他进程进入访问同一临界资源的临界区。
但该进程在临界资源的访问过程中，如正在使用打印机，它可能由于等待打印的完成而处于阻塞状态，此时系统可以调度另一个进程执行。
因此，处于临界区的进程是可以被中断的。

练习

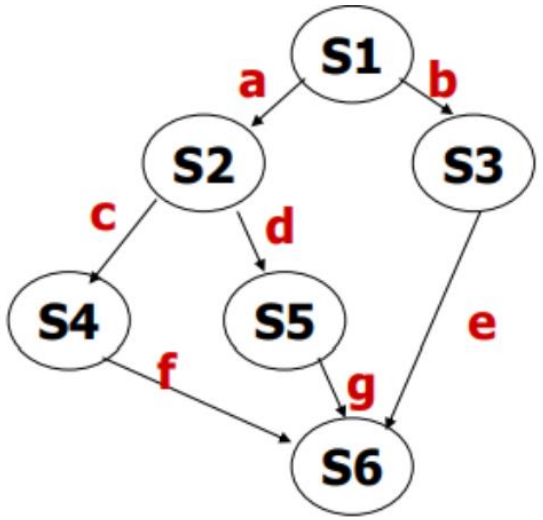
某寺庙，有小和尚、老和尚若干．庙内有一水缸，由小和尚提水入缸，供老和尚饮用。水缸可容纳 30 桶水，每次入水、取水仅为1桶，不可同时进行。水取自同一井中，水井径窄，每次只能容纳一个水桶取水。设水桶个数为5个，试用信号量和PV操作给出老和尚和小和尚的活动。

```
semaphore empty=30;   // 表示缸中目前还能装多少桶水，初始时能装30桶水
semaphore full=0;   // 表示缸中有多少桶水，初始时缸中没有水
semaphore buckets=5;   // 表示有多少只空桶可用，初始时有5只桶可用
semaphore mutex_well=1;   // 用于实现对井的互斥操作
semaphore mutex_bigjar=1;   // 用于实现对缸的互斥操作
```

```
young_monk(){
while(true){
    P(empty);
    P(buckets);
    go to the well;
    P(mutex_well);
    get water;
    V(mutex_well);
    go to the temple;
    P(mutex_bigjar);
    pour the water into the big jar;
    V(mutex_bigjar);
    V(buckets);
    V(full);
}}
```

```
old_monk(){
while(true){
    P(full);
    P(buckets);
    P(mutex_bigjar);
    get water;
    V(mutex_bigjar);
    drink water;
    V(buckets);
    V(empty);
}}
```

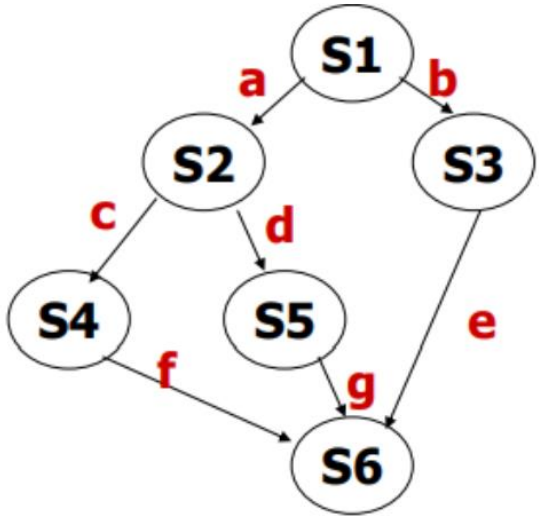
以下是表示若干进程为了完成任务，必须遵照的前驱图。请问如何对进程S1-S6进行并发控制，才能保证按照前驱图所要求的先后顺序正确完成。



主观题 10分

设置

以下是表示若干进程为了完成任务，必须遵照的前驱图。请问如何对进程S1-S6进行并发控制，才能保证按照前驱图所要求的先后顺序正确完成。

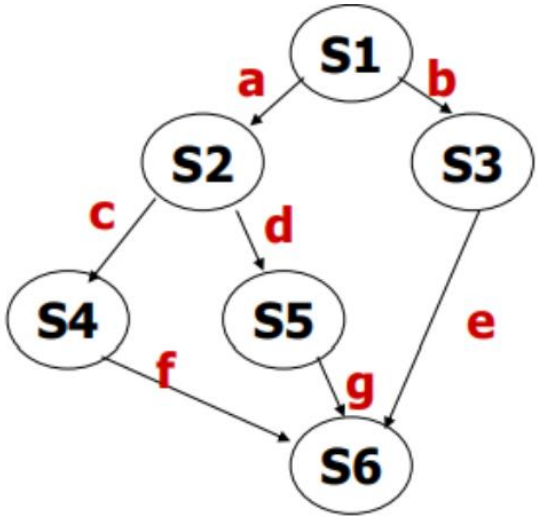


作答

信号量a,b,c,d,e,f,g的初值均设为0

```
P1( ) {
    S1;
    signal(a);
    signal(b);
}
P2( ) {
    wait(a);
    S2;
    signal(c);
    signal(d);
}
P3( ) {
    wait(b);
    S3;
    signal(e);
}
```

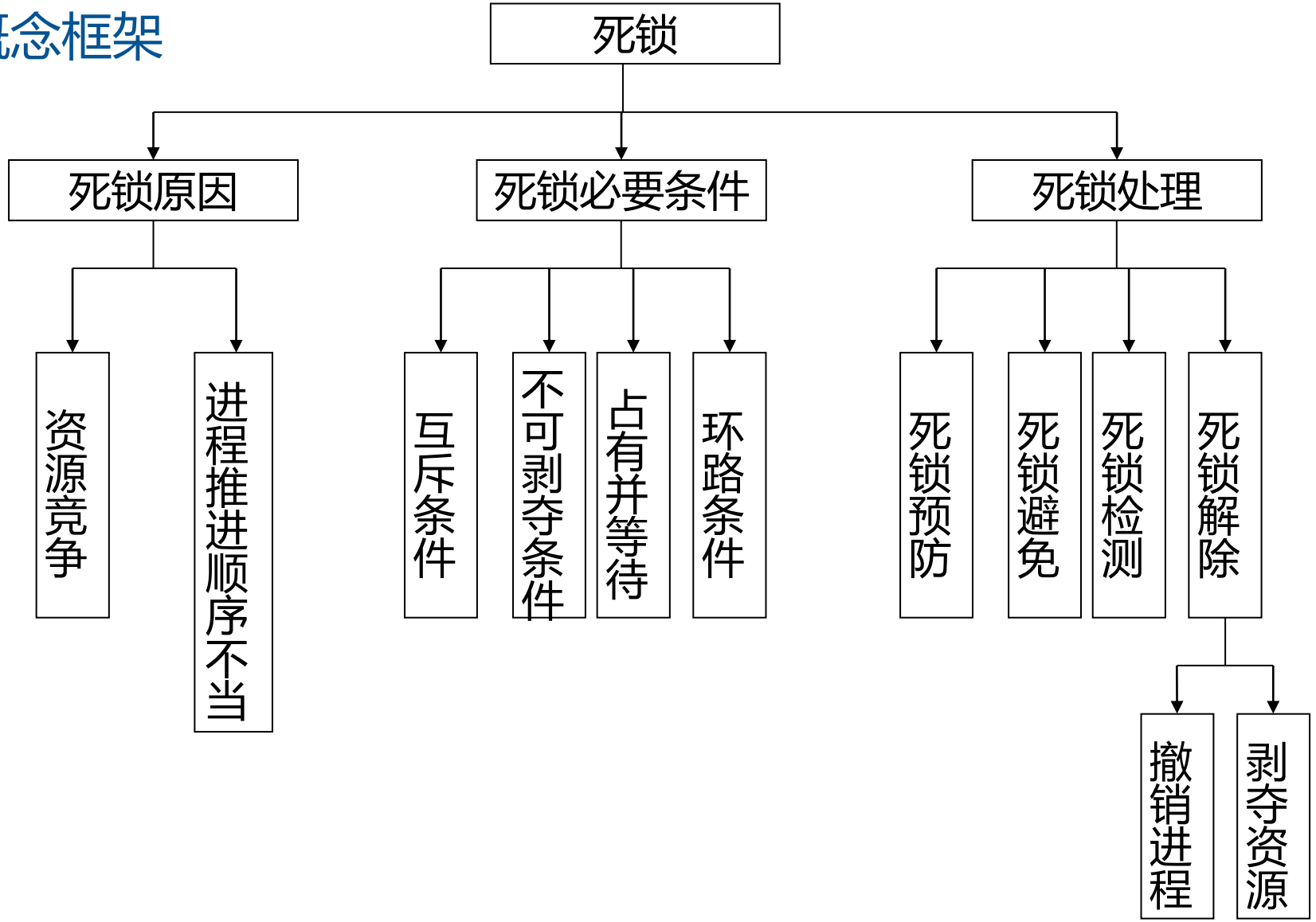
```
P4( ) {
     wait(c);
     S4;
     signal(f);
}
P5( ) {
    wait(d);
    S5;
    signal(g);
}
P6( ) {
    wait(e);
    wait(f);
    wait(g);
    S6;
}
```



https://developer.aliyun.com/article/885149

死锁

Deadlock概念框架



练习

死锁避免是根据（）采取措施实现的。

A. 配置足够的系统资源

B. 使进程的推进顺序合理

C. 破坏死锁的4个必要条件之一

D. 防止系统进入不安全状态

练习

资源的按序分配可以破坏_____条件。

A 互斥

B 不可抢占

C 部分分配

D 循环等待

可以证明，资源按序分配的约束之下，系统中的进程不会形成资源等待环路。

练习

银行家算法是一种_____算法

A. 死锁预防

B. 死锁避免

C. 死锁检测

D. 死锁解除

死锁避免（Deadlock Avoidance）

Q: 银行家算法进行死锁避免的具体思路是怎样的?

银行家算法练习

系统使用银行家算法进行死锁预防，其中管理3类资源X,Y,Z，系统中有3个进程。当前系统状态如下表所示：

	Allocation			Max		
	X	Y	Z	X	Y	Z
P0	0	0	1	8	4	3
P1	3	2	0	6	2	0
P2	2	1	1	3	3	3

系统当前剩余资源Available=(3,2,2)。若进程P0,P1分别要提出请求REQ0=(1,0,2)，REQ1=(2,0,0)，请问以下说法是正确的?

A. 仅REQ0可被满足
B.仅REQ1可被满足
C.REQ0和REQ1都可被满足
D.REQ0、REQ1都不可被满足

单选题 1分

设置

系统使用银行家算法进行死锁预防，其中管理3类资源X,Y,Z，系统中有3个进程。当前系统状态如下表所示。系统当前剩余资源Available=(3,2,2)。若进程P0,P1分别要提出请求REQ0=(1,0,2)，REQ1=(2,0,0)，请问以下说法是正确的?

	Allocation			Max		
	X	Y	Z	X	Y	Z
P0	0	0	1	8	4	3
P1	3	2	0	6	2	0
P2	2	1	1	3	3	3

A 仅REQ0可被满足

B 仅REQ1可被满足

C REQ0和REQ1都可被满足

D REQ0、REQ1都不可被满足

提交

内存管理

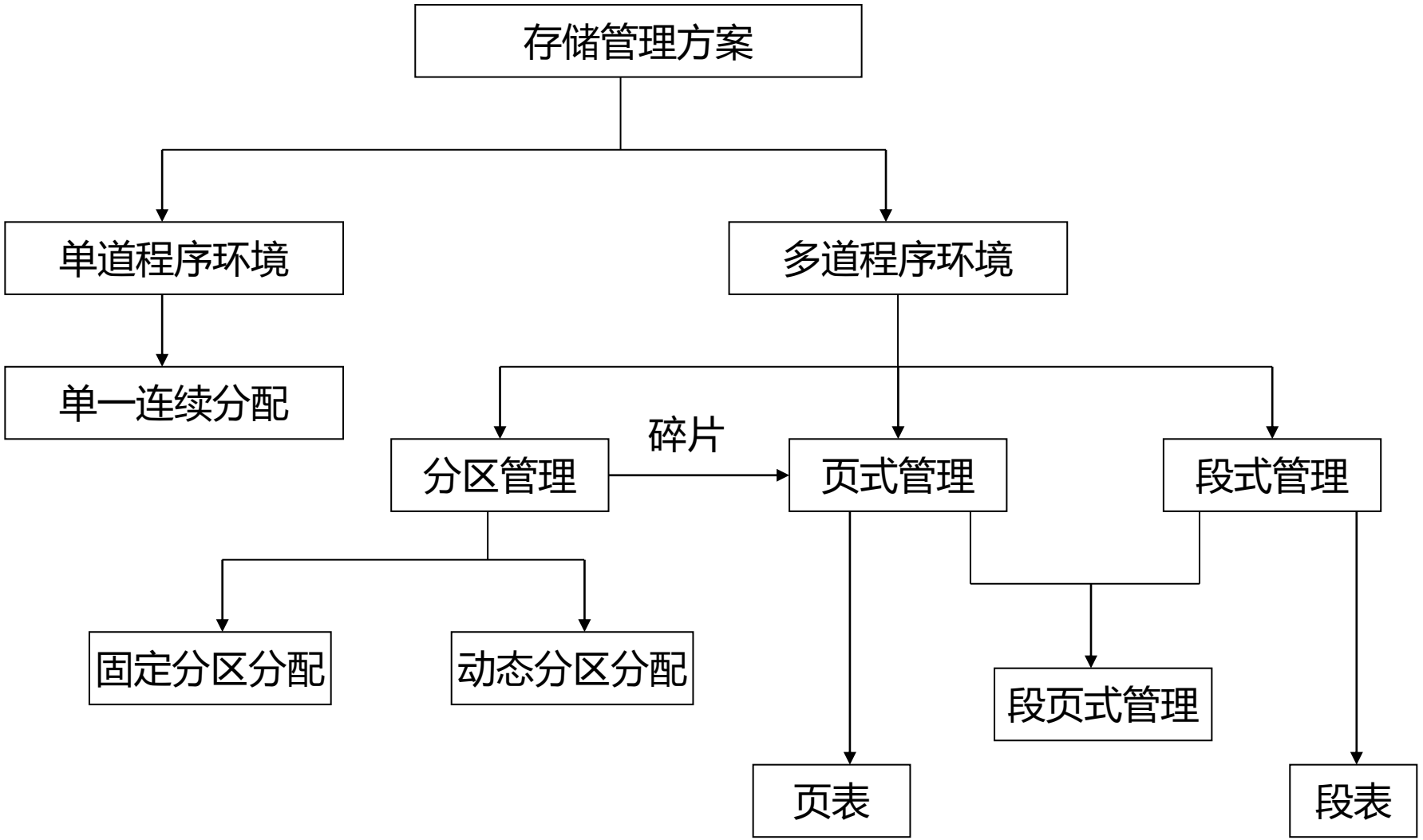
Memory Management Review

02

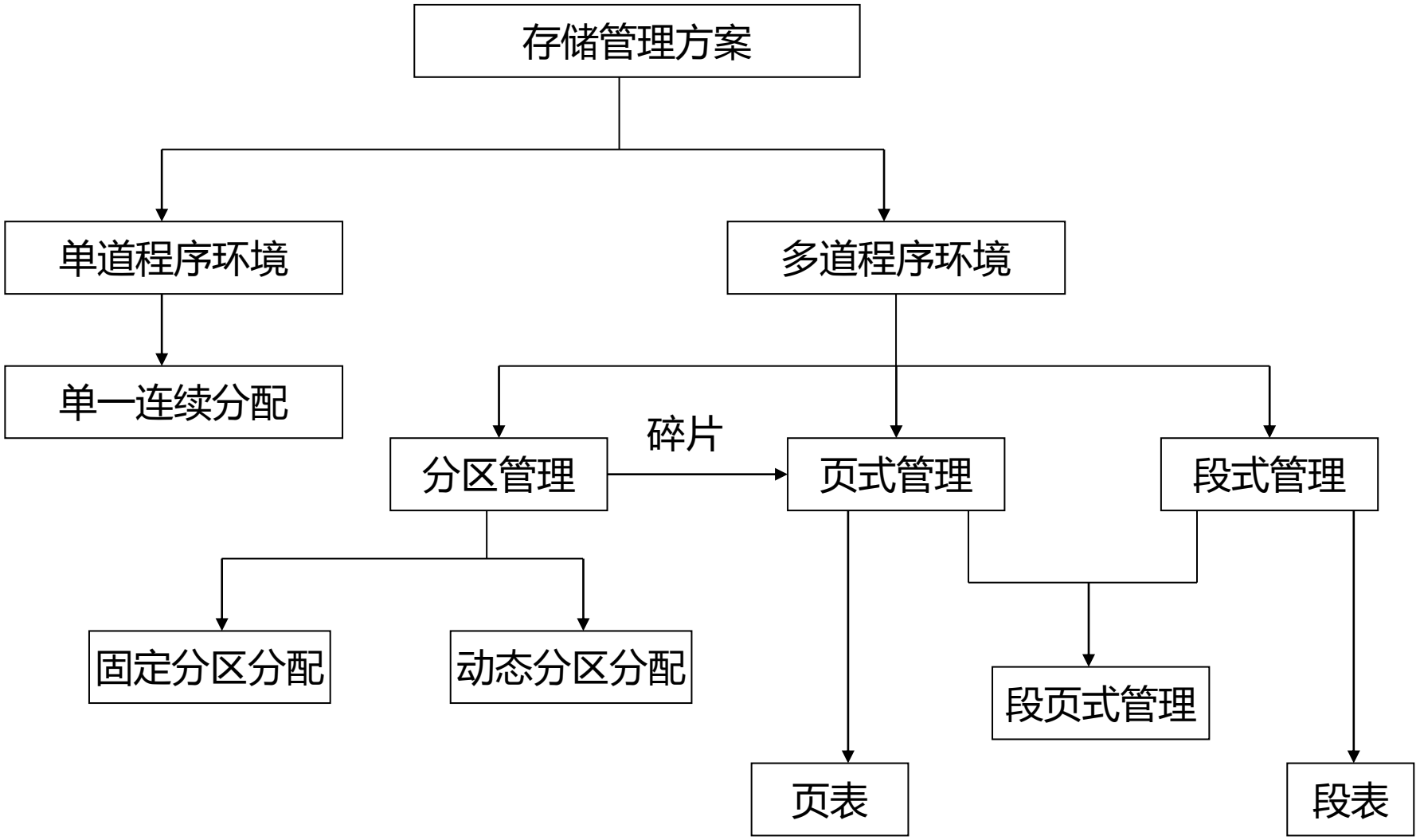
知识结构简图

- 内存管理
  - 地址绑定
    - 编译时绑定 } 静态重定位
    - 加载时绑定 } 静态重定位
    - address transformations are done before execution of a program begins.
    - 运行时绑定 —— 动态重定位
    - address transformations being done during execution of a program.
  - 内存分配：单一连续分配、固定分区分配、可变分区分配、页式分配
  - 内存扩充：覆盖、交换
  - 内存保护：内存分区越界保护、页式保护
  - 内存共享：借助进程逻辑地址空间到物理地址空间的映射机制，实现内存共享

知识结构简图


存储管理方案
单道程序环境
多道程序环境
单一连续分配
分区管理
碎片
页式管理
段式管理
固定分区分配
动态分区分配
段页式管理
页表
段表

知识结构简图


存储管理方案
单道程序环境
多道程序环境
单一连续分配
分区管理
碎片
页式管理
段式管理
固定分区分配
动态分区分配
段页式管理
页表
段表

练习

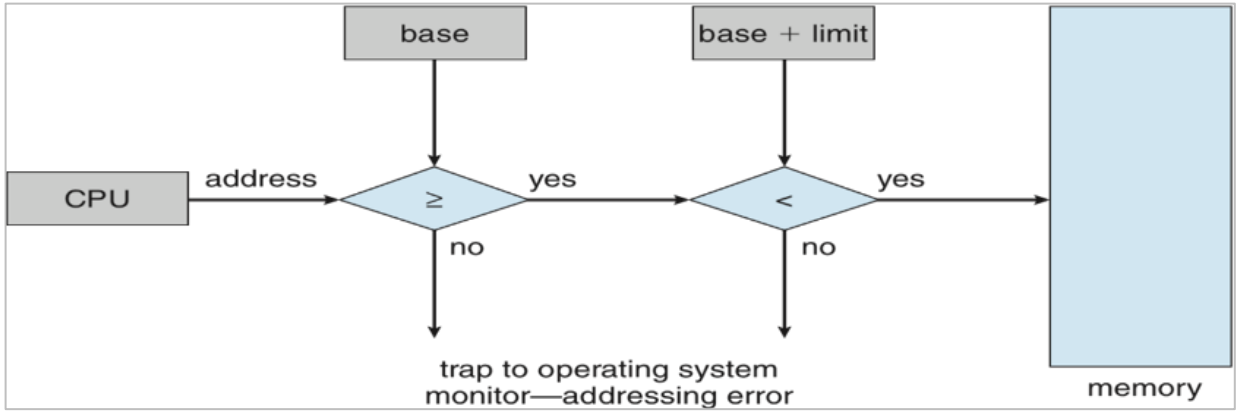
- 多进程能够在内存中彼此互不干扰的环境下运行，操作系统是通过（）来实现的?
  - A、内存分配　　B、内存保护
  - C、内存扩充　　D、地址映射

练习

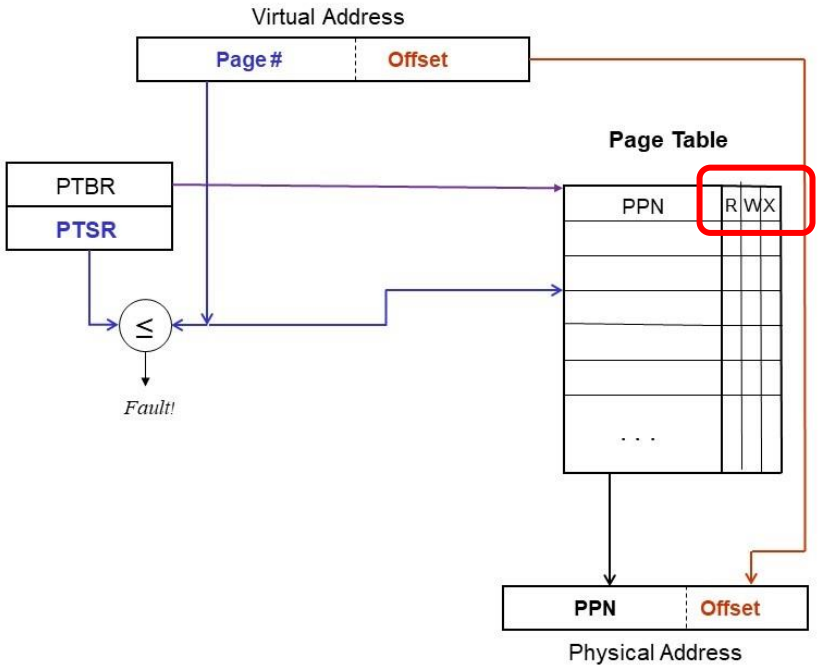
- 多进程能够在内存中彼此互不干扰的环境下运行，操作系统是通过（）来实现的?
  - A、内存分配　　B、内存保护
  - C、内存扩充　　D、地址映射

- 答案B
- 存储器的主要功能：内存分配、内存保护、地址映射和内存扩充等。
  - 内存分配：为进程分配内存空间；
  - 内存保护：确保每个进程都在自己的内存空间运行；
  - 地址映射：逻辑地址到物理地址的转换；
  - 内存扩充：借助虚拟存储技术，逻辑上扩充内存容量，一般借助请求调入和置换功能实现。

内存保护



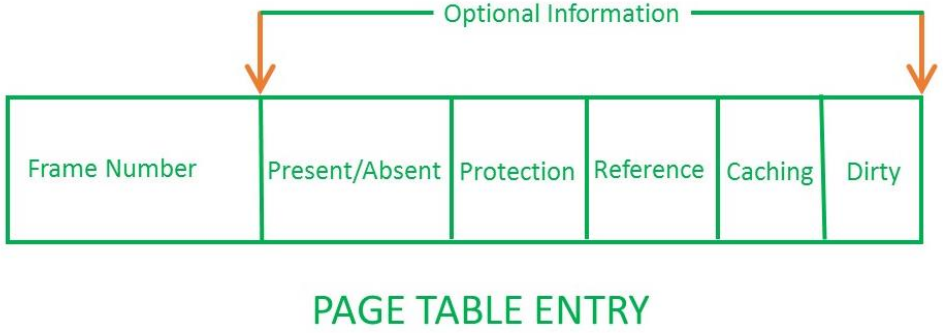
分区内存保护



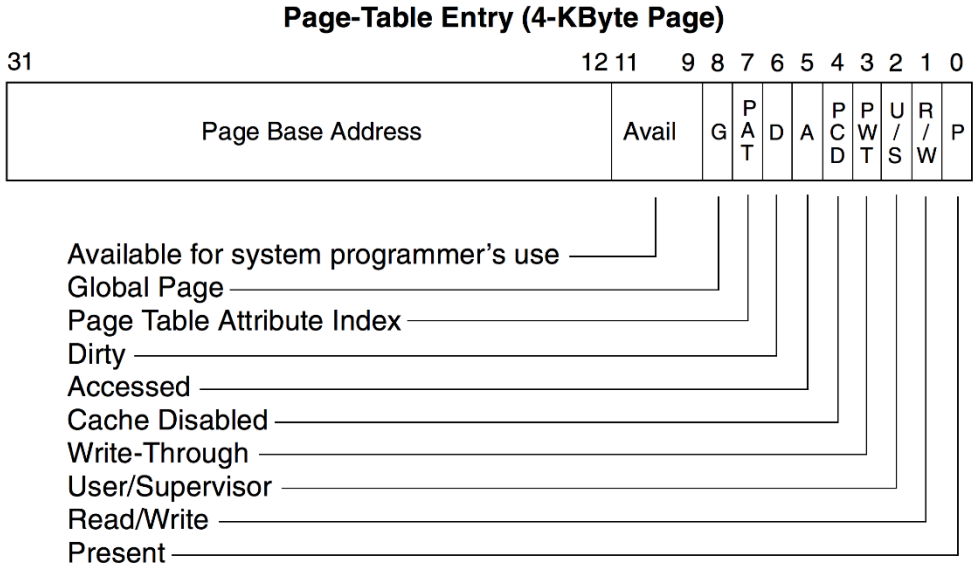
页式内存保护

内存保护

Q: 分页机制下，页表项(PTE)的位如何充分利用?

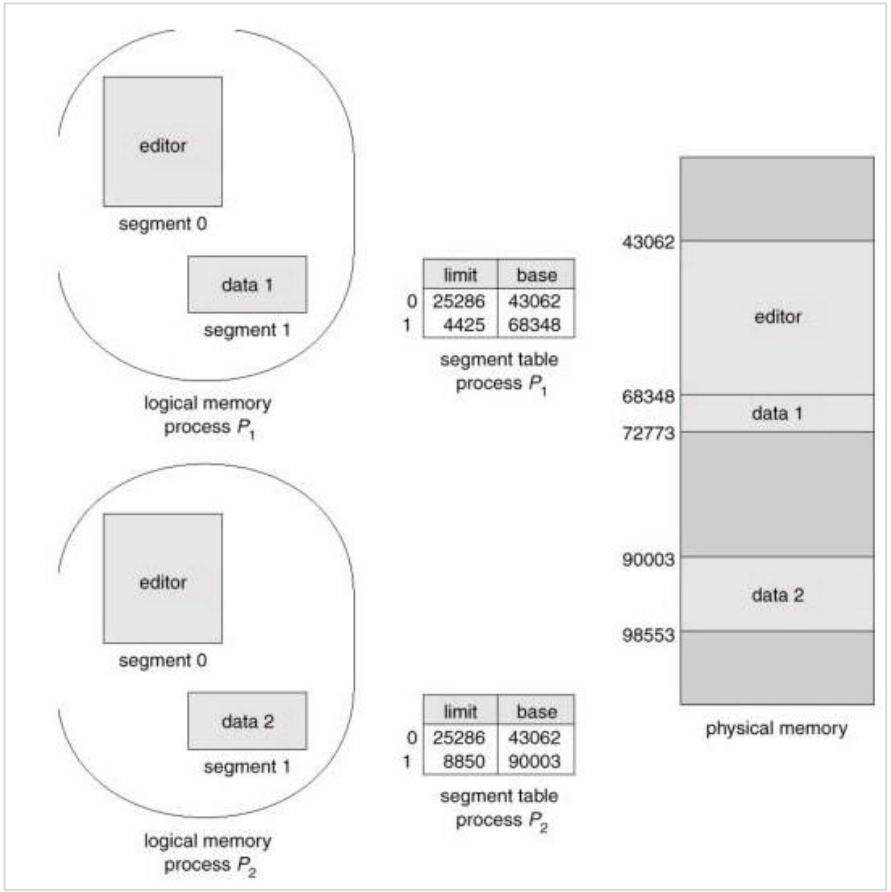


PAGE TABLE ENTRY

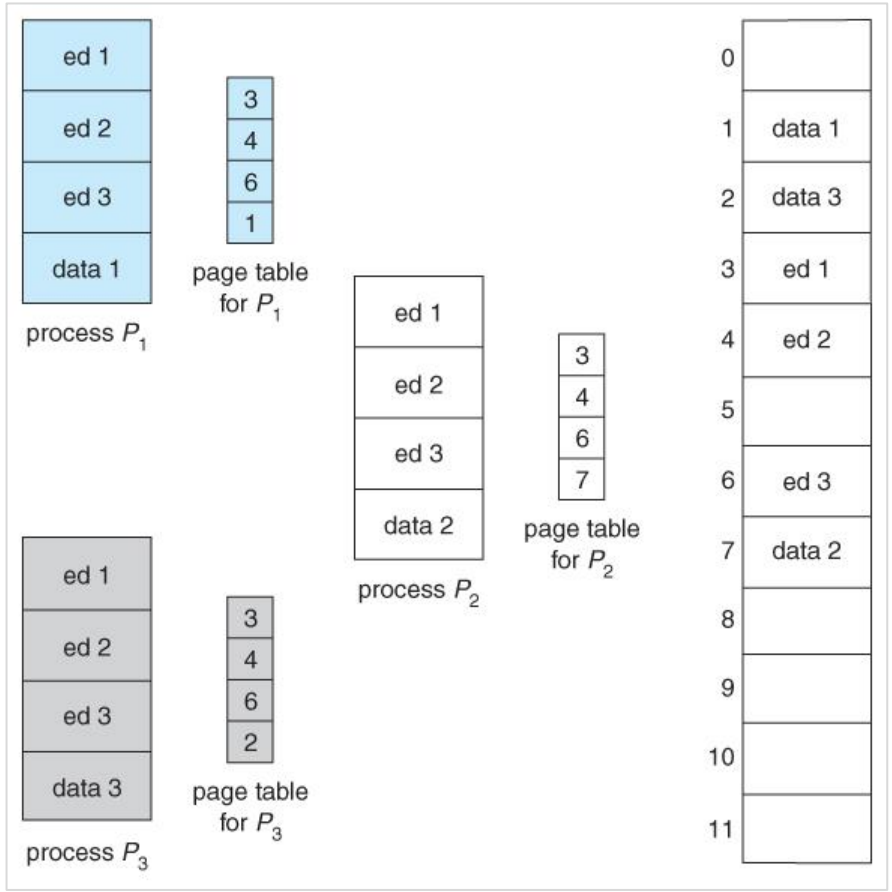


Intel IA32 PTE

内存共享



Sharing of segments



Sharing of pages

练习

- 动态重定位是在程序的（）中进行的。
  - A、编译过程　　B、装入过程
  - C、链接过程　　D、执行过程

练习

- 动态重定位是在程序的（）中进行的。
  - A、编译过程　　B、装入过程
  - C、链接过程　　D、执行过程

- 答案D
- 地址重定位的2种方式：静态重定位，指在程序运行前由链接或加载（linker or loader）程序一次性的完成地址转换；动态重定位，在程序运行过程中，当要存取指令或数据的时候，在存取之前完成逻辑地址到物理地址的转换。

练习

问题3：静态重定位与动态重定位

- 什么是静态重定位?
- 什么是动态重定位?
- 试比较静态重定位和动态重定位。

问题3：静态重定位与动态重定位

- 静态与动态重定位的比较
- 答：
- 静态重定位的特点：
  - 实现容易，无需增加硬件地址变换机构
  - 通常要求为每个程序分配连续的内存区
  - 程序执行期间，装入内存的代码不能移动，难以做到程序和数据的共享

问题3：静态重定位与动态重定位

- 静态与动态重定位的比较
- 答：
- 动态重定位的特点：
  - 实现需依靠硬件地址变换机构
  - 可重定位代码直接原样装入内存
  - 动态重定位机制下，能够很容易地支持将程序分配到不连续的内存区
  - 支持程序移动，便于利用零散的内存空间，利于实现信息共享和虚拟存储：所以，现代计算机多采用动态重定位的方式

内存扩充

覆盖

- 什么是覆盖?
- 答:
- 覆盖是早期内存扩充方式
- 所谓覆盖，是指同一内存区可以被不同的程序段重复使用覆盖技术要求程序员必须把程序分成若干个程序段，并规定好它们的执行和内存覆盖顺序

动态分区分配算法

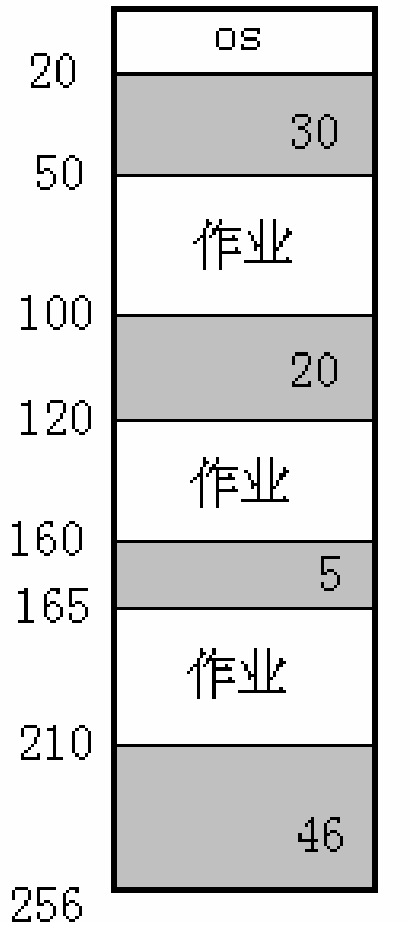
首次适应（最先适配）

最佳适应

最差适应

连续内存分配

习题



- 请求表中的作业序列：
- （1） 作业A要求18K；作业B要求25K，作业C要求30K
- 3种分区分配法各自怎么根据作业的内存请求进行分配

主观题 10分

设置

进程对内存的请求序列：

（1） 作业A要求18K；作业B要求25K，作业C要求30K

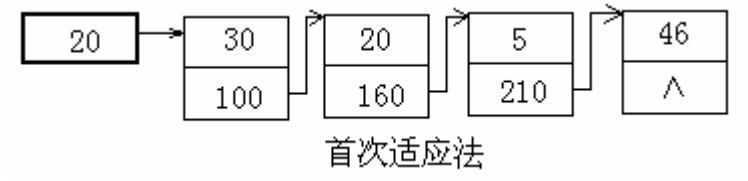
3种分区分配法各自怎么根据作业的内存请求进行分配

作答

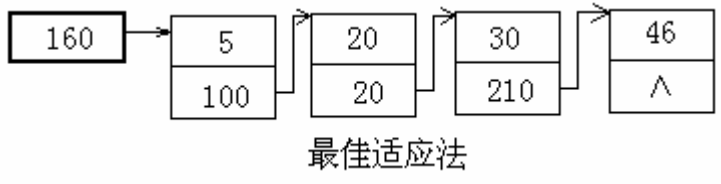
连续内存分配

习题

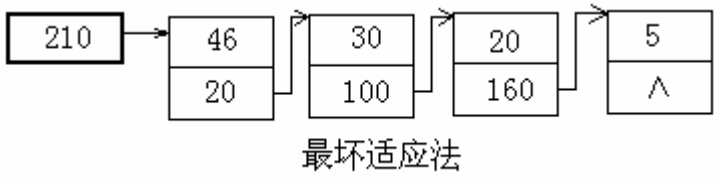
问题中的空闲链表组织（首次适应、最佳适应、最差适应）



首次适应法



最佳适应法



最坏适应法

Request: A(18K), B(25K),C(30K)

内存访问

- 在分页、分段和段页式存储管理中，当访问一条指令或数据时，需要访问内存几次？各做什么处理？
  - 分页和分段中，访问一条指令或数据时至少需要访问内存两次，一次是访问内存中的页表（或段表），实现逻辑地址到物理地址的转换，二是访问真正的物理地址得到相应的指令或数据。
  - 段页式管理中，至少需要访问内存3次，一是访问内存中的段表，查找段内页表的起始地址；二是访问内存中存放的页表，实现逻辑地址到物理地址的转换；三是访问真正的物理地址得到相应的指令或数据。

内存利用率

- 内存利用率不高主要体现为哪几种形式?
  - 内存中存在大量分散、难以利用的碎片;
  - 暂时或长期不能运行的程序和数据占用了大量的内存空间;
  - 大内存需求进程多，系统内存只能装入少量进程，当这些进程处于阻塞状态时可能导致CPU被闲置，从而降低内存利用率;
  - 内存中存在重复的复制件。

内存利用率

- 可以通过哪些途径提高内存利用率?
  - 采用离散分配方式（如分页），减少内存碎片；
  - 增加对换机制，将暂时不用的程序和数据换出到外存；
  - 采用虚拟存储管理技术，增加内存中并发进程数；
  - 引入动态加载和链接技术，只将需要的部分装入内存；
  - 提供存储共享能力，减少内存中重复复制件的存在。

信号量1

题目

- 桌子上有一只盘子，每次只能向其中放入一个水果。爸爸专向盘子中放苹果，妈妈专门向盘子中放橘子。儿子专等吃盘子中的橘子,女儿专等着吃盘子中的苹果。只有盘子空时，爸爸或妈妈才可向盘子中放一个水果。仅当盘子中有自己所需的水果时，儿子或女儿可以从盘子中取出水果。请用信号量问题解决此问题。

主观题 10分

设置

桌子上有一只盘子，每次只能向其中放入一个水果。爸爸专向盘子中放苹果，妈妈专门向盘子中放橘子。儿子专等吃盘子中的橘子,女儿专等着吃盘子中的苹果。只有盘子空时，爸爸或妈妈才可向盘子中放一个水果。仅当盘子中有自己所需的水果时，儿子或女儿可以从盘子中取出水果。请用信号量问题解决此问题。

作答

专项练习

信号量2

题目

有一个仓库，可以存放 A 和 B 两种产品，仓库的存储空间足够大，但要求: (1)一次只能存入一种产品(A 或 B); (2)-N < (A 产品数量-B 产品数量) < M。 其中，N 和 M 是正整数。试用“存放 A”和“存放 B”以及 P、V 操作描述产品 A 与 产品 B 的入库过程。

主观题 10分

设置

有一个仓库，可以存放 A 和 B 两种产品，仓库的存储空间足够大，但要求: (1)一次只能存入一种产品(A 或 B); (2)-N < (A 产品数量-B 产品数量) < M。 其中，N 和 M 是正整数。试用“存放 A”和“存放 B”以及 P、V 操作描述产品 A 与 产品 B 的入库过程。

作答

谢 谢!
Thank you!