# 第二章 进程的描述与控制

# 本章小结

程序顺序执行的三大特征；

**程序并行执行时的三大特征；**

**进程的五大特征；**

**进程的七种状态及其切换（三种基本状态）；**

进程六种控制原语；

**进程的两种制约关系；**

**进程与程序的区别；**

**进程同步机制应遵循的四大规则；**

**四种信号量机制；**

**前趋图及信号量实现前趋关系；**

**管理机制与条件变量；**

**进程与管程的区别；**

**进程同步的三大基本问题；**

**进程通信的四大类型；**

消息传递通信的两种实现方式；

**进程和线程的区别；**

线程的四大特性；

线程的三种类型及实现方式；

重要概念：进程、线程、管程、原语、进程实体、临界区、PCB、轻型进程、互斥、同步、条件变量、信号量、前趋图、进程图；

## 2.1 前趋图和程序执行

### 2.1.1 前趋图

1. 顺序和并发的执行过程，有向无循环图；

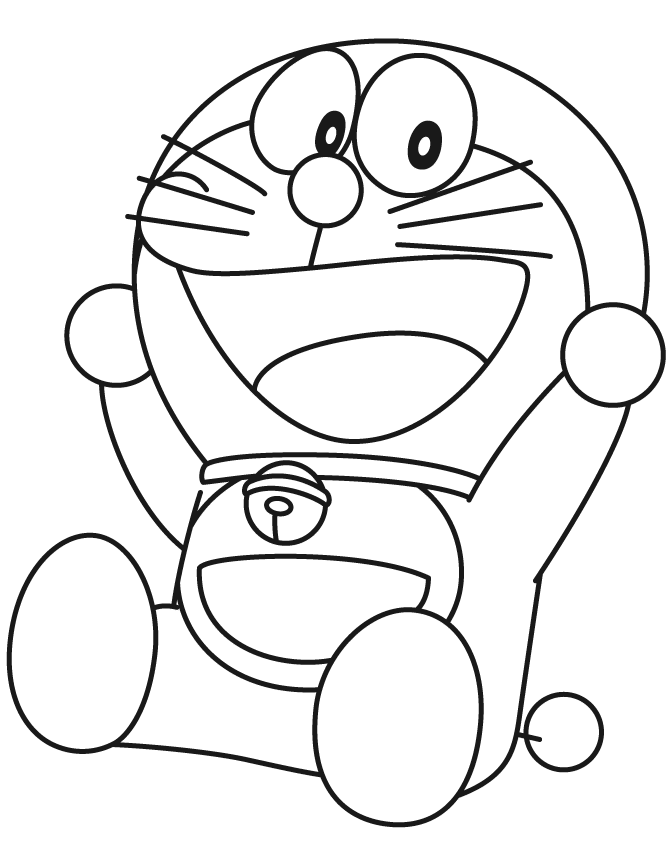
2. 直接前趋，直接后继，初始结点，终止结点；

### 2.1.2程序顺序执行及其特征

1. 特征：顺序性，封闭性（独占全机资源），可再现性（初始条件和环境相同，结果相同）；

### 2.1.3 程序的并发执行及其特征

1. 同一时间间隔内同时执行；

2. 特征：间断性，失去封闭性，不可再现性；

## 2.2 进程的描述

### 2.2.1 进程的定义和特征

1. 进程实体（PCB）：程序段+数据段+进程控制块;

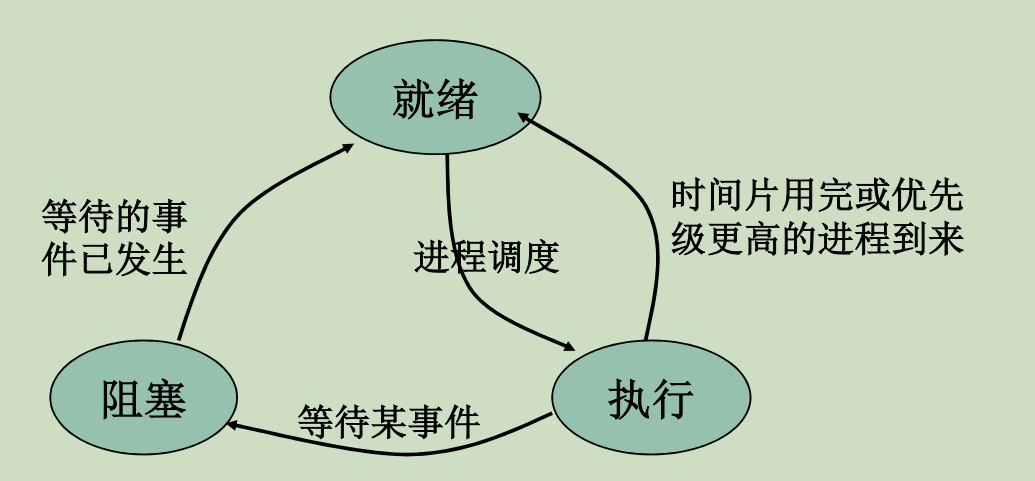
2. 进程的特征：**动态性**，并发性，独立性，异步性；

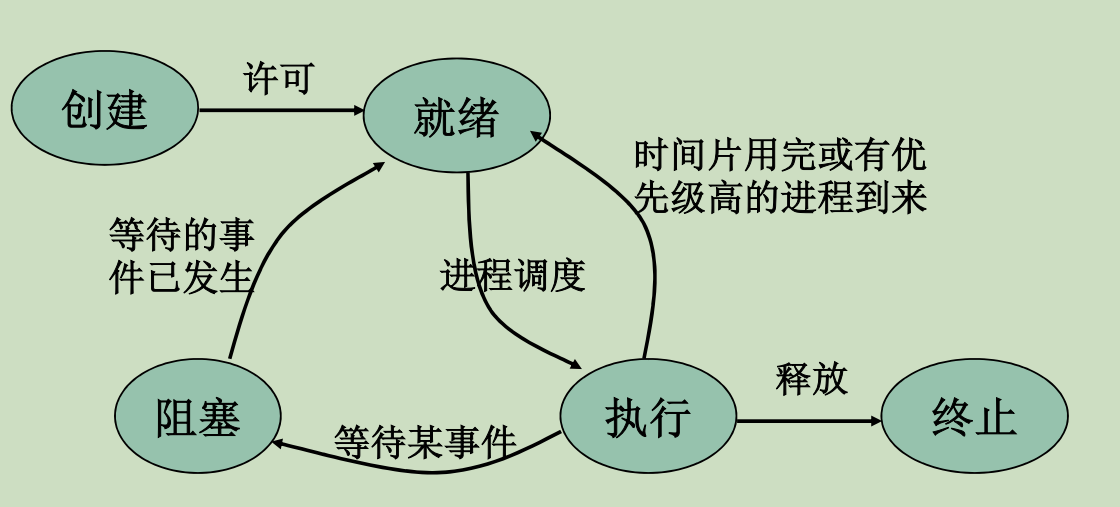
3. 进程和程序的区别

|  |  |
| --- | --- |
| 进程 | 程序 |
| 动态 | 静态 |
| 暂时 | 永久 |
| 有结构特征 | 指令的有序集合 |
| 并非一一对应关系 | |

### 2.2.2 基本状态及转换

1. 进程的三种基本状态：（1）就绪状态（获得除CPU以外的全部资源）；（2）执行状态（已获得CPU程序正在执行）；（3）阻塞状态（无法继续执行）；



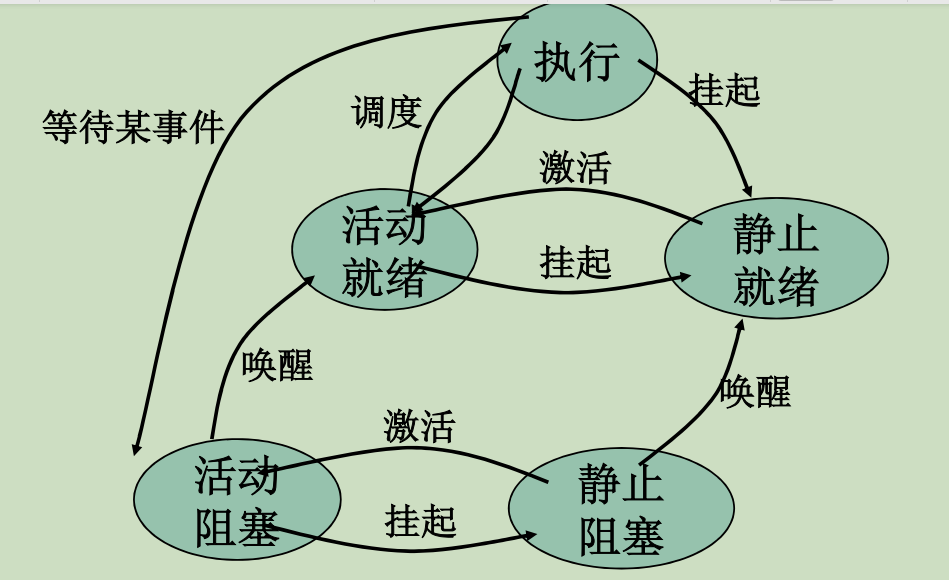
1. 创建状态和终止状态：（1）创建状态：创建，拥有自己的PCB，未分配到内存资源；（2）终止状态：正常结束、异常结束、受外界干预结束；释放其所拥有的全部资源，回收PCB；
2. 挂起状态（换出内存状态）：
3. 原因：①终端用户的请求；

②父进程请求；

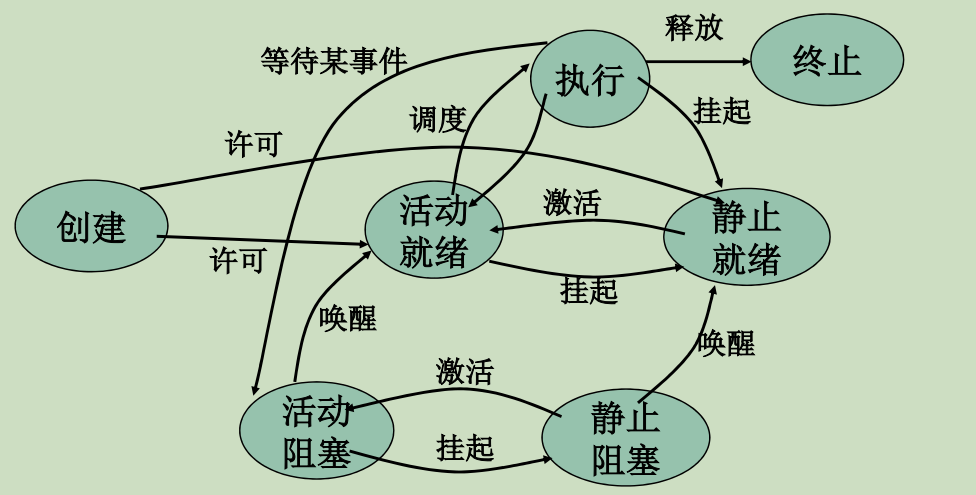
③负荷调节的需要；

④操作系统的需要；

1. 进程的转换：



1. 七种状态的转换：



1. 具有挂起状态的进程转换：

①挂起：活动阻塞→静止阻塞；活动就绪→静止就绪；

②激活：静止就绪→活动就绪；

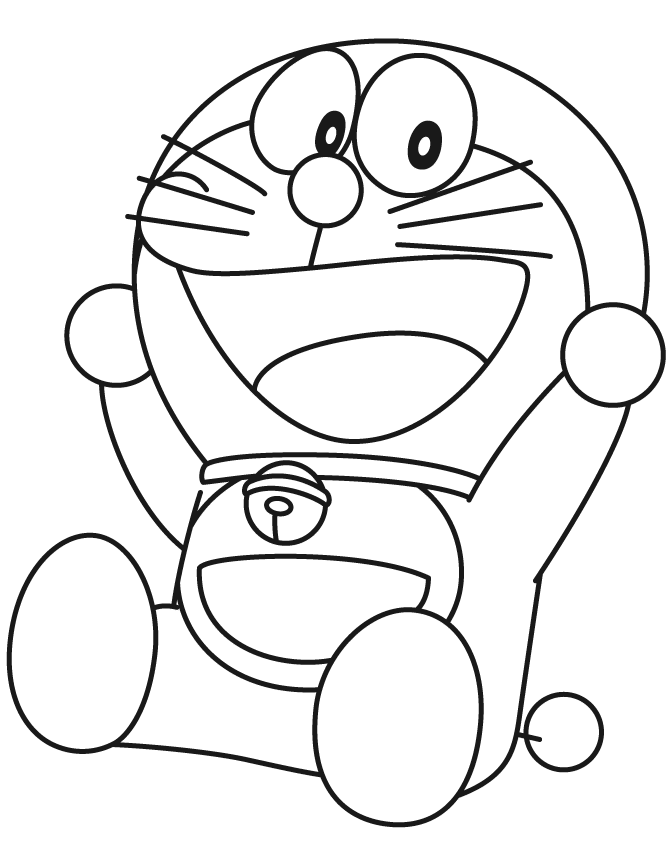
③唤醒：活动阻塞→活动就绪；静止阻塞→静止就绪；

### 2.2.3 进程控制块

1. PCB的作用：（1）作为独立运行基本单位的标志；

（2）能实现间断性运行方式；

（3）提供进程管理所需要的信息；

 （4）提供进程调度所需要的信息；

（5）实现与其他进程的同步与通信；

2. 组织方式：线性方式，链接方式，索引方式；

## 2.3 进程控制

### 2.3.1 基本概念

1. 系统态和用户态；

2. 原语和原子操作；

3. 进程控制是进程管理中最基本的功能，一般由原语来实现；

### 2.3.2 进程的创建

1. 进程图：有向树，子可继父，撤父必撤子；

2. 引起创建进程的事件：（1）用户登录（2）作业调度

（3）提供服务（4）应用请求；

3. 进程的创建：（1）申请空白PCB；

（2）为新进程分配资源；

（3）初始化PCB；

（4）将新进程插入就绪队列；

### 2.3.3 进程的终止（destroy原语）

1. 正常结束：运行完；

2. 异常结束；

3. 外界请求；

### 2.3.4 进程的阻塞与唤醒

1. 阻塞（block原语）；

2. 唤醒（wakeup原语）；

### 2.3.5 进程的挂起与激活

1. 挂起suspend原语；

2. 激活active原语；

## 2.4 进程同步

### 2.4.1 进程同步的基本概念

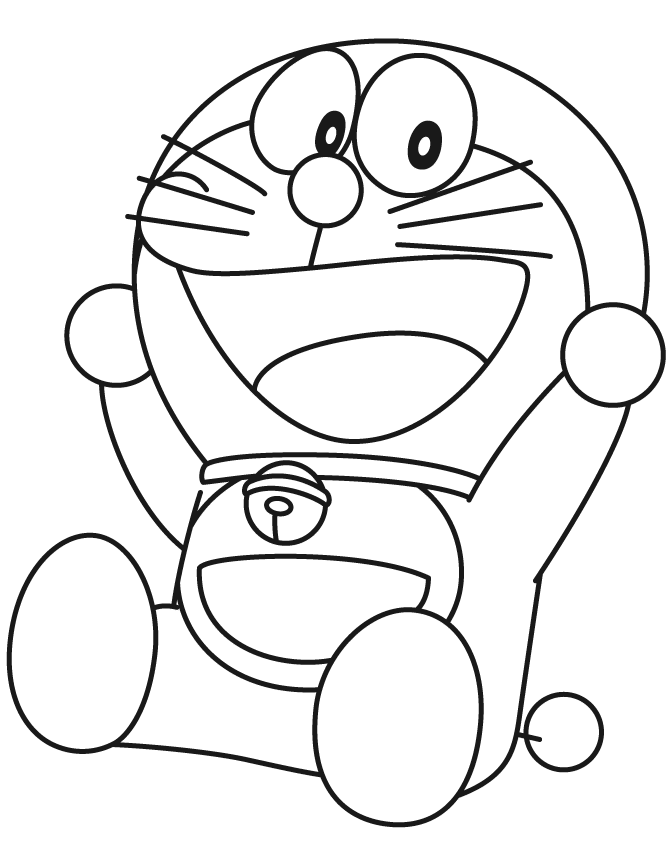
1. 进程同步：是对多个相关进程在执行次序上进行协调，使并发执行的进程之间能按照一定的规律共享系统资源，并能很好地相互合作，从而使程序的执行具有可再现性；

2. 进程间的两种制约关系：（1）间接制约：互斥；

（2）直接制约：同步、合作关系；

3. 临界资源：一段时间内只允许一个进程访问；

4. 临界资源共享冲突：程序并发执行的不可再现性通常是因为对临界资源没有互斥访问造成的；

5. 临界区：每个进程中访问临界资源的那段代码；

6. 同步机制应遵循的规则：空闲让进，忙则等待，

优先等待，让权等待；

### 2.4.2 硬件同步机制

1. 锁，测试和关锁操作必须是连续的，不允许分来进行；

2. 关中断：计算机系统不响应中断；

缺点：可能导致严重的后果，影响系统效率，不适用于多CPU系统；3. 利用Test-and-Set指令实现互斥：lock=false（空闲），lock=true（正在被使用）；

1. 利用Swap指令实现进程互斥；

### 2.4.3 信号量机制

一个信号量代表一类资源的可用数量；

1. 整型信号量：仅能通过两个原子操作wait和signal来访问，即P(请求资源）V（释放资源）操作，成对出现；

缺点：未遵循让权等待原则，造成忙等；

1. 记录型信号量：让权等待，消除忙等；

缺点：实现多类资源共享，易造成死锁；

（一个同步P操作和一个互斥P操作在一起时，先同步后互斥，两个V操作的顺序无关紧要；当P.V操作为互斥操作时，在一个进程内，当同步操作时，则不在一个进程中）；

1. AND信号量机制：实现多类资源共享，避免死锁；

基本思想：将进程在整个运行过程中需要的所有资源，一次性全部分配给进程，进程使用完后一起释放；

1. 信号量集：每一个信号量代表分配时一定下线要求，且一次需要分配一定数量的资源；

### 2.4.4 信号量的应用

1.利用信号量实现进程互斥；

2. 利用信号量实现前趋关系；

### 2.4.5 管程机制

1. 管程：由一组代表着共享资源的局部变量、对局部变量进行初始化的语句序列和操作的过程以及为了实现进程同步而引入的一组条件变量共同构成的一种资源管理模块；

2. 管程的特性：模块性，抽象性、安全性、互斥性、共享性、集中性；

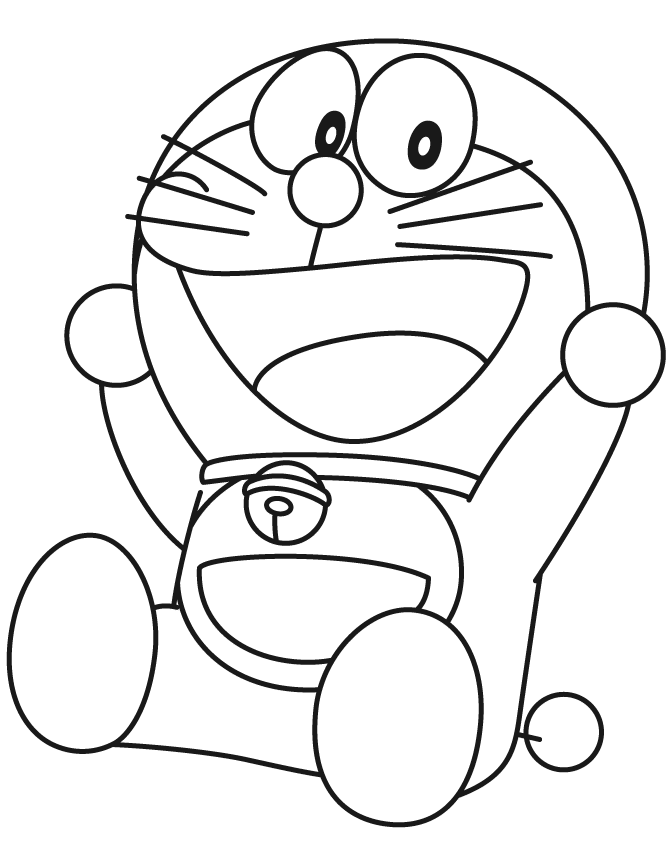
3. 管程实现同步——条件变量

只能通过wait、signal两个原语操作访问；

是一种链表结构，对其作wait操作时用来记录不满足相应条件而阻塞的所有进程；

## 2.5 经典进程同步问题

### 2.5.1生产者消费者问题

1. 问题描述：生产者的生产消费者的消费保持同步；
2. mutex,empty,full;

## 2.5.2哲学家进餐问题

## 2.5.3 读者-写者问题

允许多个读者进行读操作，但不允许一个写者和多个写者、读者同时访问共享对象；

## 2.6 进程通信

1. 进程之间的信息交换；

2.低级、高级进程通信；

### 2.6.1 进程通信的类型

1. 共享存储器系统（低级）：基于共享数据结构的通信方式，基于共享存储区的通信方式；

2. 管道通信系统：连接一个发送进程（写文件）和一个接受进程（读文件）之间通信的共享文件；

特点：效率高，能有效传送大量数据；

必须提供的三方面的协调能力：互斥，同步，双方均存在；

1. 消息传递系统：以通信命令来实现，进程间的数据交换以格式化的信息为单位，当前应用最为广泛；直接通信、间接通信、
2. 客户机-服务器系统：实现方法：套接字（文件型、网络型）、远程文件调用、远程方法调用；

### 2.6.2 消息传递通信的实现方式

1. 直接通信方式：利用OS所提供的发送命令，直接把消息发送给目标进程；

（1）直接通信原语：对称寻址方式、非对称寻址方式；

（2）进程同步的方式：发送/接受，阻塞/不阻塞；

（3） 通信链路；

2. 间接通信方式：进程之间的通信需要信箱作为媒介来进行通信；

（1）优点：可实时通信，又可非实时通信；

（2）信箱的结构：信箱头、信箱体；

（3）信箱通信原语：信箱的创建于撤销，消息的发送和接受；

（4）信箱类型：私用信箱、公用信箱、共享信箱；

### 2.6.3 直接消息传递系统实例

## 2.7 线程

### 2.7.1 线程的基本概念

1. 进程的两个基本属性：（1）进程是一个可拥有资源的独立单位；

（2）进程是一个可独立调度和分派的基本单位；

2. 程序并发执行时所需付出的时空开销：创建进程、撤销进程、进程切换；

3. 线程——作为分派和调度的基本单位；

4. 线程的三种状态：执行、就绪、阻塞；

5. 进程与线程的比较

|  |  |  |
| --- | --- | --- |
|  | 进程 | 线程 |
| 调度性 | 资源和独立调度 | 调度和分派 |
| 并发性 | 可 | 可 |
| 拥有资源 | 拥有资源的基本单位 | 访问其所属进程的资源 |
| 独立性 | 高 | 低 |
| 系统开销 | 大 | 小 |
| 支持多处理机系统 | 一个 | 可支持多个 |

### 2.7.2 进程间的同步与通信

1. 互斥锁；

2. 条件变量；

3. 信号量机制：私用一个、公用多个；

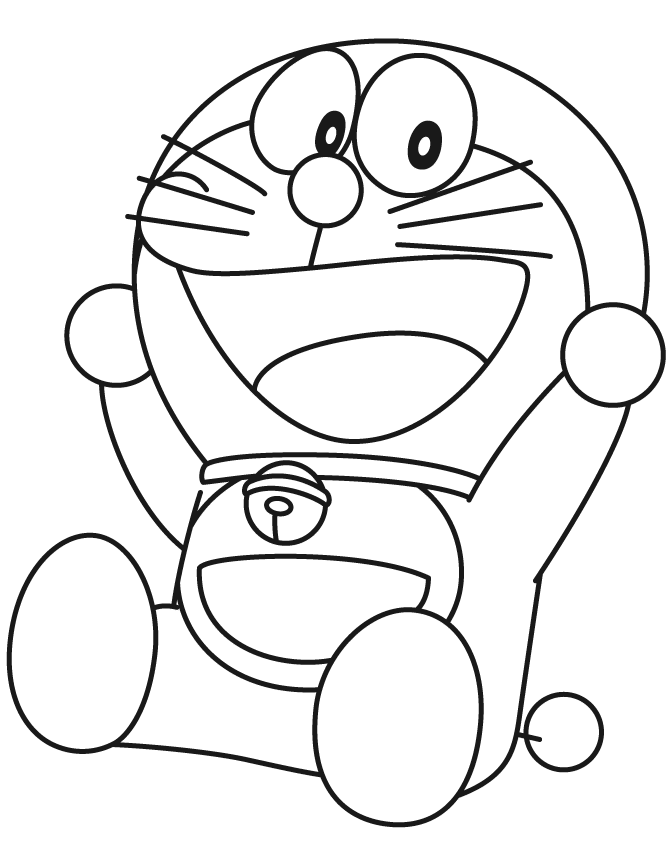
### 2.7.3 线程的实现方式（线程的类型）

1.内核支持线程：内核条件的支持下运行；

2. 用户级线程：仅存在于用户空间中，于内核无关；

3. 组合式线程；

### 2.7.4 线程的实现

1. 内核支持线程实现；

2. 用户级线程和组合级线程的实现：