并行与并发

12级ACM班 刘一鸣

一些概念

- 并行：多个线程在任何时间点都同时执行。
- 并发：多个线程在某段时间可以同时执行。
- 摩尔定律
- Amdahl定律

一些概念

- 超线程技术：仅使用一个核执行多个线程。（比执行一个线程效率提高约30%）
- 多处理器技术：使用多个物理处理器。二者可结合使用（酷睿）

多线程开发

- 应用程序所采用的设计方法和结构
- 多线程应用程序接口（API）
- 编译环境和运行环境

线程

- 线程：相关的指令序列
- 线程状态：就绪，运行，等待（阻塞），终止
- 层次：用户级，操作系统级（内核级），硬件线程
- 进程：多个线程的集合，有独立的地址空间。

线程的层次

多个

进

程

操作系统线程

硬件线程

操作系统的层次

应用程序层次

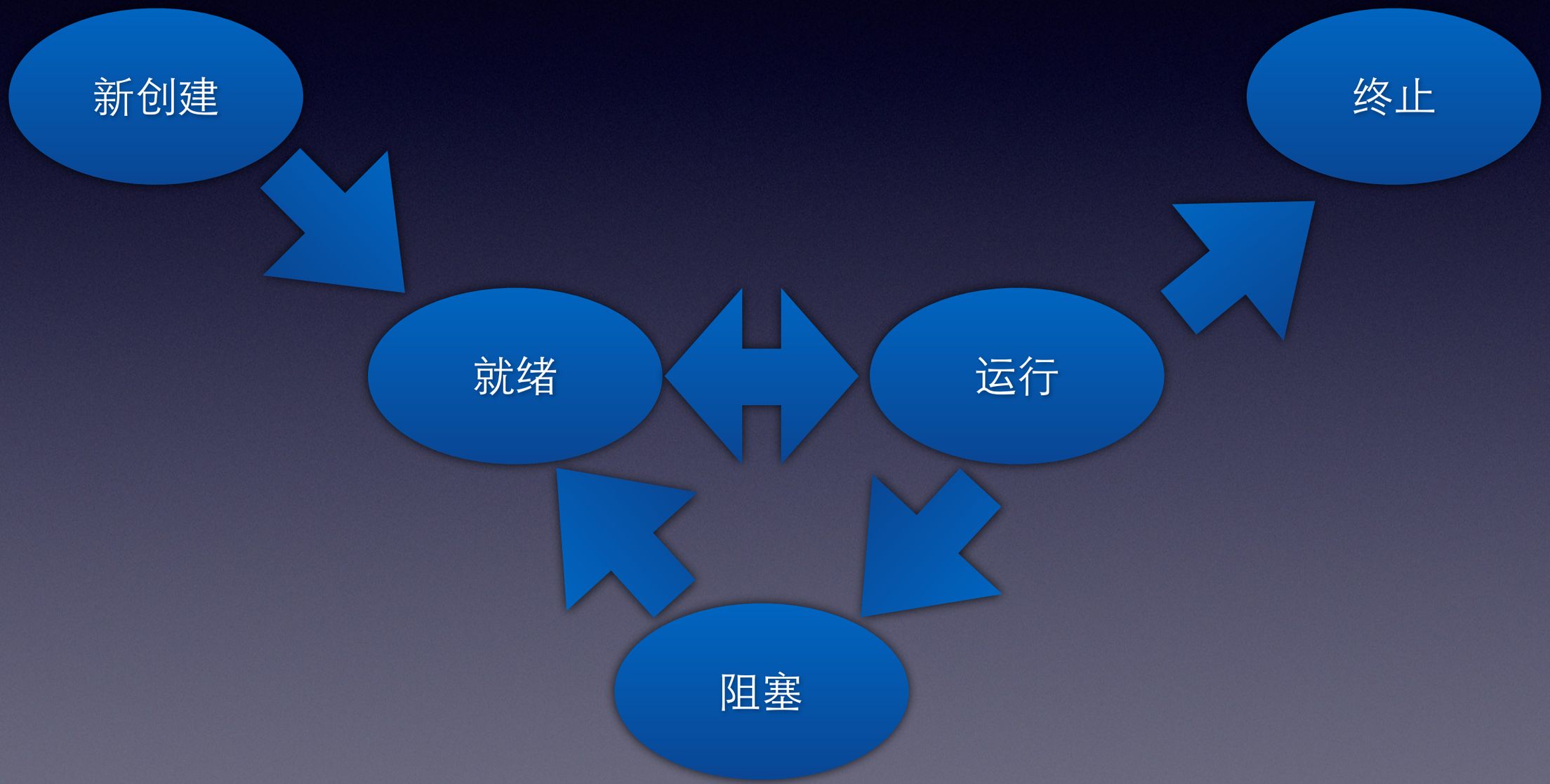
系统库

操作系统

硬件抽象层

体系结构

线程的状态



并行程序设计

- 分解
- 同步
- 通信
- 负载平衡
- 可扩展性

主要分解方式

- 任务分解（IO／显示／计算分离）
- 数据分解（矩阵运算）
- 数据流分解（流水线）

同步

- 临界段：处理多线程共享数据的程序片段
- 临界段入口：保证其他线程处于等待状态
- 临界段出口：允许其他线程进入临界段
- 共享数据通常只能有一个线程可以访问

同步

- 原子操作：是指不会被线程调度机制打断的操作；这种操作一旦开始，就一直运行到结束，中间不会切换到另一个线程。

同步

- 信号量：一个整数sem，有两个原子操作，分别是加减1
- 和判断是否大于0
- 当请求一个使用信号量来表示的资源时，进程需要先读取信号量的值来判断资源是否可用。大于0，资源可以请求，等于0，无资源可用，进程会进入阻塞状态直至资源可用。当进程不再使用一个信号量控制的共享资源时，信号量的值+1，对信号量的值进行的增减操作均为原子操作，

同步

- 锁：类似于信号量。有两个原子操作，获取锁与释放锁。
- 一个锁之多由一个线程获得。
- 死锁：线程对锁的需求形成一个环。

同步

- 锁的使用:
- 定义所有必需的锁
- 临界段开始：获取锁
- 临界段：对锁保护的数据执行操作
- 临界段结束：释放锁

通讯

- 线程之间：通过共享数据（一定要用锁保护好）
- 进程之间：通过管道，信号，Socket等

线程API

- pthread
- winAPI
- Qt Thread
- Java Thread

常见并行程序设计问题

- 线程过多（串行化）
- 死锁或锁竞争过于激烈（数据拷贝）

FishTank架构

QT

RenderEvent

EventManager

Callback／Pipe

Environment

Callback

AI

线程