# 第三章 仿真程序的设计与实现

在本项目的仿真程序中，四种模型的仿真程序主体框架大致相似，所有的操作事件（顾客到达、开始服务、结束服务）在程序中均体现为一个带有时间戳的函数，通过一个简单的事件循环 (Event loop) 程序按生成顺序进行处理。事件循环程序每一轮都从事件列表中取出时间戳最近的函数进行调用，每个函数在处理过程中都可能会生成更多的事件，直到达到停止条件（完成服务的顾客数量>30000）。事件循环的流程图如下图所示：

这里放图

**3.1 “事件”的定义**

对于一个队列，三种事件函数的定义如下：

* 顾客到达事件函数：

1. 若空闲服务员数量>0，则空闲服务员数量-1，调用开始服务事件函数

若空闲服务员数量=0，则队列长度+1

1. 随机生成一个时间戳，与顾客到达事件函数组合为一个事件放入事件列表

* 开始服务事件函数：

随机生成一个时间戳，与结束服务事件函数组合为一个事件放入事件列表

* 结束服务事件函数：

若队列长度>0，则队列长度-1，调用开始服务事件函数

若队列长度=0，则空闲服务员数量+1

**3.2 仿真不同的队列模型**

根据事件的定义和事件循环的运行原理，对于不同的单队列排队模型：

* 仿真不同的到达过程：调整顾客到达事件函数中随机生成的时间戳间隔的概率分布
* 仿真不同的服务过程：调整开始服务事件函数中随机生成的时间戳间隔的概率分布
* 仿真多个服务员的情况：在事件循环开始前写入多个服务事件函数
* 仿真多个队列并列的情况：初始化多对空闲服务员数量和队列长度计数值，并使多个事件处理函数操作不同的计数值
* 仿真服务N个顾客的情况：事件循环的停止条件设置为结束服务事件函数运行次数大于N

因此，若以λ表示总到达强度、μ表示每个服务员的服务强度，易得以下队列仿真参数：

* M|M|n队列：
  + 顾客到达事件生成的时间戳间隔为指数分布E(λ)
  + 开始服务事件生成的时间戳间隔为指数分布E(μ)
  + 初始化服务事件函数为n
* 基于概率分裂的n个M|M|1队列：
  + 初始化n对空闲服务员数量和队列长度计数值
  + 顾客到达事件生成的时间戳间隔为指数分布E(λ)，且随机选择一对计数值进行处理
  + 开始服务事件生成的时间戳间隔为指数分布E(μ)
  + 初始化服务事件函数为n，且每个服务事件函数都始终处理固定的一对计数值
* n个En|M|1队列：
  + 顾客到达事件生成的时间戳间隔为Erlang分布En(λ/n)，且随机选择一对计数值进行处理
  + 其他参数与基于概率分裂的n个M|M|1队列相同
* 选择最短队列长度的n个M|M|1队列
  + 顾客到达事件生成的时间戳间隔为指数分布E(λ)，且随机选择一对计数值进行处理，若有多个
  + 其他参数与基于概率分裂的n个M|M|1队列相同

**3.3 数据收集和统计方式**

在事件函数中添加记录当前时间戳和队列长度的方式即可实现数据收集，对收集到的K个(时间戳,队列长度)数据，易计算：

使用不同的λ、μ和n进行仿真，即可得到不同条件下各队列的Elq仿真值。