# МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ ПРОГРАММНЫХ КАНАЛОВ

**Щербакова О.Г.**
Кафедра ПМиИ ДонГТУ

***Abstract. Shcherbakova O. The Mathematical model Programming Pipe Functioning.*** *It was given the mathematical model for the programming pipe functioning. The model for a stationary mode as a regional task for systems of the linear integrated equations of a type Volterra is received. On the basis of the received model the parameters of functioning of the program pipe is calculated at the various entrance data and it was given the most general recommendation for the pipe's use. In the further given model can be used for realization of a superstructure multiprocesses operational system, for automation of a choice of that or other means of the communications between processes by the system*

## 1. Введение

Как известно, большинство современных операционных систем, имеющих повсеместное распространение в мире, такие как Unix, Windows NT, OS/2, обладают хорошо развитыми встроенными средствами взаимодействия параллельных процессов и аппаратом системных вызовов, позволяющих системному программисту использовать их непосредственно. К числу наиболее известных простейших средств явного взаимодействия относятся программные каналы, очереди сообщений (обменники) и общие сегменты оперативной памяти. В современных многопроцессных ОС выбор того или иного средства коммуникации целиком возлагается на программиста и зависит от его опыта и интуиции. Однако функционирование упомянутых средств может быть описано как система массового обслуживания в терминах случайных процессов, что позволит вырабатывать обоснованные рекомендации по выбору параметров, рассчитывать показатели производительности и эффективности, и в конечном итоге, оценивать целесообразность применения того или иного средства взаимодействия для решения каждой конкретной задачи, а также автоматизировать выбор средства коммуникации между процессами самой системой, что полностью освободит данный аспект программирования от субъективного фактора.

## 2. Входные данные и характеристики

Построим аналитическую вероятностную модель функционирования программных каналов.

По сути программный канал – это некоторый файл, разделяемый среди некоторого числа процессов, в который можно писать информацию или читать информацию из него. Такой файл должен быть одновременно открыт на чтение и на запись. Обычно канал имеет сравнительно небольшой объем (от 512 до 4096 байт). Это пространство используется как кольцевой буфер между процессами-писателями и процессами-читателями в соответствии с дисциплиной FIFO и правилами синхронизации по чтению/записи [2].

Введем вложенный Марковский процесс по моментам завершения каналом обслуживания запроса на чтение или запись. Похожий прием используется, как известно, при анализе классической системы M/G/1 [1]. Состояние системы определим тройкой $(n_1, n_2, x)$, где

- $n_1, n_2$ – количество запросов в очереди на чтение и на запись соответственно. Оба эти параметра дискретны. Модель будем строить для стационарного

режима, следовательно $0 <= n_1 <= N_1$, $0 <= n_2 <= N_2$, где – $N_1$, $N_2$ - число процессов, открывших канал на чтение и запись соответственно;

- x – заполненность канала. Это непрерывный параметр, причем $0 <= x <= C$, где C – объем канала.

Для составления уравнений необходимо рассмотреть все возможные варианты состояния системы $(k_1,k_2,v)$ в предыдущий контрольный момент и ее последующее поведение, приводящее к данному состоянию $(n_1,n_2,x)$. Стационарную вероятность таких состояний будем обозначать $P_{n1n2}(x)$ и $P_{k1k2}(v)$ соответственно. Для определения перехода $(k_1, k_2, v) \rightarrow (n_1, n_2, x)$ нам потребуются следующие данные:

- распределение длительности промежутков времени между обращениями процессов к каналу на чтение и на запись. Будем считать, что эти промежутки распределены экспоненциально с параметрами $\lambda_1$ и $\lambda_2$ соответственно, хотя легко показать, что условие экспоненциальности этих распределений не является существенным для построения марковского процесса и приняты для облегчения аналитических выкладок;
- $\beta_1$ и $\beta_2$ - плотности распределения для заказываемых объемов на считывание и запись соответственно;
- $c_1$ и $c_2$ – коэффициенты пропорциональности затрачиваемого процессами времени на чтение и запись объему обрабатываемой информации соответственно.

### 3. Основные принципы построения модели

Модель будем строить с учетом приоритета заявки на чтение: если в контрольный момент времени очередь на чтение не пуста, то запрос на обслуживание берется из нее. Модель с приоритетом на запись строится аналогично. Для упрощения примем, что частично удовлетворенные запросы (заявки на чтение, которые запрашивали объем больший, чем количество данных, находящихся в канале, либо заявки на запись, которым требуется объем больший свободного места в канале) покидают систему частично неудовлетворенными. В реальной же системе такие запросы переводятся в состояние ожидания, то есть фактически снова поступают в очередь с остаточной длиной запроса. Учет этого обстоятельства сильно усложнил бы модель, так как для сохранения марковского свойства пришлось бы учитывать еще и остаточную длину неудовлетворенного запроса, и трех компонент для обозначения состояния было бы уже недостаточно, при этом пространство состояний резко увеличилось.

Рассмотрим все возможные состояния системы в предыдущий контрольный момент:

1. Предыдущая операция была чтение, и она открыла новый период занятости (под новым периодом занятости понимаем начало обслуживания после простоя, т.е. либо обе очереди были пусты, либо заявки, находящиеся в какой-либо из очередей невозможно было обработать). Свободного места в канале не было, значит в очереди на запись могли быть запросы $(k_1=0, 0<=k_2<=n_2, v=C)$.
2. Предыдущая операция чтения открыла новый период занятости, но в канале было свободное место. В этом случае очередь на запись была пуста, и все заявки пришли во время выполнения операции чтения $(k_1=0, k_2=0, x<v<C)$.
3. Предыдущая операция чтения не открыла новый период занятости, то есть заявка из очереди начала обслуживаться непосредственно сразу же после завершения предшествующей операции $(1<=k_1= n_1+1, 0<=k_2<=n_2, x<v<=C)$.
4. Предыдущая операция была запись, и она открыла новый период занятости, при этом канал был пуст, следовательно, очередь на чтение могла быть не пустой $(0<=k_1= n_1, k_2=0, v=0)$.

5. Предыдущая операция записи открыла новый период занятости, при этом в канале имелись данные, а значит очередь на чтение обязательно была пуста и все $n_1$ заявок на чтение появились во время выполнения записи ($k_1= 0$, $k_2=0$, $0<v<x$).
6. Предыдущая операция записи не открыла новый период занятости, и в канале имелись данные. В этом случае очередь запросов на чтение также была пуста ($k_1= 0$, $1<=k_2<=n_2+1$, $0<v<x$).
7. Предыдущая операция записи не открыла новый период занятости, но канал был пуст, то есть в очереди на чтение могли быть заявки, но обслужить их было невозможно ($0<=k_1= n_1$, $1<=k_2<=n_2+1$, $v=0$).

Каждому из вышеизложенных вариантов соответствует слагаемое, характеризующее вероятность перехода из данного ряда состояний в состояние ($n_1$, $n_2$, x). Граничные условия (x=0 и x=C) были рассмотрены отдельно.

### 4. Вывод уравнений

Начнем с рассмотрения варианта 3, как наиболее простого. Если в данный контрольный момент времени завершилась операция чтения, а x>0, то $0<=n_1<N_1$, $0<=n_2<=N_2$ (поскольку мы приняли, что все запросы, даже частично удовлетворенные покидают очередь и процесс, выдавший запрос на чтение, не может находиться в очереди, значит $n_1$ по крайней мере на единицу меньше, чем $N_1$). Поэтому в слагаемом, соответствующем варианту 3, присутствует множитель $\sum_{k_1=1}^{n_1+1}\sum_{k_2=0}^{n_2}\int_x^C P_{k_1,k_2}(v)\beta_1(v-x)\,dv$ (1).

Далее, обслуживание запроса потребовало $c_1(v-x)$ единиц времени, и оно началось сразу же после завершения обслуживания предыдущего запроса. Отследим, как изменилось за это время число запросов на запись. В предыдущий момент в очереди на запись стояли $k_2$ процессов-писателей, а $N_2-k_2$ были активны. Из них $n_2-k_2$ за время $c_1(v-x)$ выдали запрос на запись, а $N_2-n_2$ - не выдали. Следовательно, у $n_2-k_2$ процессов остаточное время интервала между генерацией запросов оказалось меньше, чем $c_1(v-x)$, а у $N_2-n_2$ процессов - больше. Таким образом, к подынтегральной функции (1) добавляется множитель $H_2(v,x)=C_{N_2-k_2}^{n_2-k_2}(1-e^{-\lambda_2 c_1(v-x)})^{n_2-k_2}e^{-\lambda_2 c_1(v-x)(N_2-n_2)}$. Отследив таким же образом изменение числа запросов на чтение получим $H_1(v,x)=C_{N_1-k_1}^{n_1-k_1+1}(1-e^{-\lambda_1 c_1(v-x)})^{n_1-k_1+1}e^{-\lambda_1 c_1(v-x)(N_1-n_1-1)}$. Таким образом, окончательно слагаемое, соответствующее варианту 3, примет вид:

$$S_3=\sum_{k_1=1}^{n_1+1}\sum_{k_2=0}^{n_2}\int_x^C P_{k_1,k_2}(v)\beta_1(v-x)\cdot H_1(v,x)\cdot H_2(t,x)dv \qquad (2)$$

Перейдем к варианту 2. В предыдущий контрольный момент система находилась в состоянии (0,0,v), причем v<C. Поэтому в слагаемом $S_2$ присутствует множитель $\int_x^C P_{0,0}(t)\cdot\beta_1(v-x)\,dt$. Далее, нужно найти вероятность того, что новый период занятости откроет именно заявка на чтение, и пока она будет выполняться, в очереди появятся еще $n_1$ заявок на чтение и $n_2$ заявок на запись. Эта вероятность равна $\int_0^\infty \lambda_1 e^{-\lambda_1 z}\cdot H_3(z,v,x)\cdot H_4(z,v,x)dz$, где

$$H_3(z,v,x)=N_1\cdot C_{N_1-1}^{n_1}\cdot e^{-\lambda_1 n_1 z}(1-e^{-\lambda_1 c_1(v-x)})^{n_1}\cdot e^{-\lambda_1(z+c_1(v-x))(N_1-1-n_1)}, \qquad (3)$$

$$H_4(z,v,x)=C_{N_2}^{n_2}\cdot e^{-\lambda_2 n_2 z}(1-e^{-\lambda_2 c_1(v-x)})^{n_2}\cdot e^{-\lambda_2(z+c_1(v-x))(N_2-n_2)} \qquad (4)$$

Таким образом $S_2 = \int_x^C P_{0,0}(t) \cdot \beta_1(v-x) \int_0^\infty \lambda_1 e^{-\lambda_1 z} H_3(z,v,x) H_4(z,v,x) dz dv$ (5)

Основываясь на аналогичных рассуждениях получаем

$$S_1 = \sum_{k_2=0}^{n_2} P_{0,k_2}(C) \cdot \beta_1(C-x) \int_0^\infty \lambda_1 e^{-\lambda_1 z} H_3(z,C,x) H_4^*(z,C,x) dz \text{, где} \quad (6)$$

$$H_4^*(z,v,x) = C_{N_2-k_2}^{n_2-k_2} \cdot (1-e^{-\lambda_2(z+c_1(v-x))})^{n_2-k_2} \cdot e^{-\lambda_2(z+c_1(v-x))(N_2-n_2)}. \quad (7)$$

Для $H_4^*$, когда канал заполнен до отказа, заявки на запись могут поступать и в течение периода простоя, поскольку они все равно не будут обслуживаться и нового периода занятости не откроют.

Аналогично рассуждая получаем слагаемые для всех остальных вариантов:

$$S_7 = \sum_{k_1=0}^{n_1} \sum_{k_2=1}^{n_2+1} P_{k_1 k_2}(0) \cdot \beta_2(x) \cdot H_9(x) \cdot H_6(0,x) \text{, где} \quad (8)$$

$$H_9 = C_{N_1-k_1}^{n_1-k_1} (1-e^{-\lambda_1 c_2 x})^{n_1-k_1} \cdot e^{-\lambda_1 c_2 x (N_1-n_1)} \quad (9)$$

$$S_6 = \sum_{k_2=1}^{n_2+1} \int_0^x P_{0,k_2}(v) \cdot \beta_2(x-v) \cdot H_5(x,v) \cdot H_6(x,v)\, dv \text{, где} \quad (10)$$

$$H_5(v,x) = C_{N_1}^{n_1} \cdot (1-e^{-\lambda_1 c_2 (x-v)})^{n_1} \cdot e^{-\lambda_1 c_2 (x-v)(N_1-n_1)} \quad (11)$$

$$H_6(v,x) = C_{N_2-k_2}^{n_2-k_2+1} \cdot (1-e^{-\lambda_2 c_2 (x-v)})^{n_2-k_2+1} \cdot e^{-\lambda_2 c_2 (x-v)(N_2-n_2-1)} \quad (12)$$

$$S_5 = \int_0^x P_{0,0}(v) \cdot \beta_2(x-v) \int_0^\infty \lambda_2 e^{-\lambda_2 z} H_7(z,v,x) H_8(z,t,x) dz dv \text{, где} \quad (13)$$

$$H_7(z,v,x) = C_{N_1}^{n_1} \cdot e^{-\lambda_1 n_1 z} (1-e^{-\lambda_1 c_2 (x-v)})^{n_1} \cdot e^{-\lambda_1 (z+c_2(x-v))(N_1-n_1)} \quad (14)$$

$$H_8(z,v,x) = N_2 C_{N_2-1}^{n_2} \cdot e^{-\lambda_2 n_2 z} (1-e^{-\lambda_2 c_2 (x-v)})^{n_2} \cdot e^{-\lambda_2 (z+c_2(x-v))(N_2-n_2-1)} \quad (15)$$

$$S_4 = \sum_{k_1=0}^{n_1} P_{k_1,0}(0) \cdot \beta_2(x) \int_0^\infty \lambda_2 e^{-\lambda_2 z} H_7^*(z,0,x) H_8(z,0,x) dz \text{, где} \quad (16)$$

$$H_7^*(z,t,x) = C_{N_1-k_1}^{n_1-k_1} \cdot (1-e^{-\lambda_1 (z+c_2(x-t))})^{n_1-k_1} \cdot e^{-\lambda_1 (z+c_2(x-t))(N_1-n_1)} \quad (17)$$

Для $n_1=N_1$ слагаемые $S_4$, $S_5$, $S_6$, $S_7$ будут отсутствовать, поскольку предыдущей операцией могла быть только запись (в противном случае $n_1$ оказалось хотя бы на единицу меньше). Аналогично, для $n_2=N_2$ будут отсутствовать слагаемые $S_1$, $S_2$, $S_3$. Окончательный вид построенной системы интегральных уравнений для $0<x<C$:

$$P_{n_1,n_2}(x) = \begin{cases} S_1+S_2+S_3+S_4+S_5+S_6+S_7, & 0 \le n_1 < N_1,\ 0 \le n_2 < N_2 \\ S_1+S_2+S_3, & 0 \le n_1 < N_1,\ n_2 = N_2 \\ S_4+S_5+S_6+S_7, & n_1 = N_1,\ 0 \le n_2 < N_2 \end{cases} \quad (18)$$

Теперь рассмотрим граничные условия. В уравнения для $x=0$ войдут только слагаемые $S_1$, $S_2$, $S_3$, так как последней операцией могла быть только чтение. При этом необходимо внести некоторые изменения для этих слагаемых, учитывая, что заказываемые объемы на чтение могут превосходить объемы данных, находящихся в канале и даже емкость самого канала. Модификации для этих слагаемых таковы:

$$S^*_1 = \sum_{k_2=0}^{n_2} P_{0,k_2}(C) \cdot \int_C^\infty \beta_1(s) ds \int_0^\infty \lambda_1 e^{-\lambda_1 z} H_3(z,C,0) H_4^*(z,C,0) dz \quad (19)$$

$$S^*_2 = \int_0^C P_{0,0}(v) \cdot \int_v^\infty \beta_1(s)ds \int_0^\infty \lambda_1 e^{-\lambda_1 z} H_3(z,v,0) H_4(z,v,0) dz dv \quad (20)$$

$$S^*_3 = \sum_{k_1=1}^{n_1+1} \sum_{k_2=0}^{n_2} \int_0^C P_{k_1,k_2}(v) \int_v^\infty \beta_1(z)dz \cdot H_1(v,0) \cdot H_2(v,0)\, dt \quad (21)$$

$$P_{n_1,n_2}(0) = S^*_1 + S^*_2 + S^*_3,\ 0 \le n_1 < N_1,\ 0 \le n_2 \le N_2 \quad (22)$$

Аналогично формируем граничные условия в точке С:

$$S^*_4 = \sum_{k_1=0}^{n_1} P_{k_1,0}(0) \cdot \int_C^\infty \beta_2(s)ds \int_0^\infty \lambda_2 e^{-\lambda_2 z} H_7^*(z,0,C) H_8(z,0,C) dz \quad (23)$$

$$S^*_5 = \int_0^x P_{0,0}(v) \cdot \int_{C-v}^\infty \beta_2(s)ds \int_0^\infty \lambda_2 e^{-\lambda_2 z} H_7(z,v,C) H_8(z,v,C) dz dv \quad (24)$$

$$S^*_6 = \sum_{k_2=1}^{n_2+1} \int_0^x P_{0,k_2}(v) \cdot \int_{C-v}^\infty \beta_2(s)ds \cdot H_5(C,v) \cdot H_6(C,v) dv \quad (25)$$

$$S^*_7 = \sum_{k_1=0}^{n_1} \sum_{k_2=1}^{n_2+1} P_{k_1,k_2}(0) \cdot \int_C^\infty \beta_2(s)ds \cdot H_9(x) \cdot H_6(0,x) \quad (26)$$

$$P_{n_1,n_2}(C) = S^*_4 + S^*_5 + S^*_6 + S^*_7,\ 0 \le n_1 \le N_2,\ 0 \le n_2 < N_2 \quad (27)$$

Таким образом, получена модель для стационарного режима в виде краевой задачи для системы линейных интегральных уравнений типа Вольтерра [3]. Ядра полученных уравнений зависят от функций плотности распределения $\beta_1$ и $\beta_2$. В случае, например, экспоненциального распределения ядра будут вырожденными.

Ввиду сложности системы ее аналитическое решение не представляется возможным. Численное решение можно получить путем разбиения отрезка [0;C] на некоторое количество интервалов и применение к каждому из входящих в систему интегралов одной из формул численного интегрирования [4]. Тогда система интегральных уравнений будет сведена к системе линейных уравнений относительно неизвестных $P_{n_1,n_2}(x_i)$, причем одно из уравнений заменяется нормирующим условием $\sum_{n_1}\sum_{n_2}\sum_{i} P_{n_1,n_2}(x_i) = 1$.

В качестве показателей функционирования канала, которые рассчитывались по найденным из данной системы интегральных уравнений стационарным вероятностям были взяты следующие: среднее число запросов в системе на чтение и на запись; вероятность нахождения в какой-либо из очередей более одного запроса (для случая $N_1=N_2=1$ этот показатель будет равен 0, однако при увеличении количества процессов, обменивающихся данными через канал, этот показатель будет возрастать, и в этом случае процессы начинают «состязаться» за овладение каналом, в результате чего нужный порядок чтения-записи может быть нарушен); среднее время пребывания в системе запроса на чтение и на запись; средняя заполненность канала; вероятность того, что канал пуст; вероятность того, что канал заполнен до отказа.

## 5. Результаты экспериментальных исследований

Задавая различные входные данные (полученные экспериментально, путем тестирования ряда программ) и получая численное решение на основании построенной модели, были получены характеристики функционирования канала в различных условиях. Анализ проводился в два этапа: был проанализирован частный случай с

одним процессом-писателем, одним процессом-читателем; отдельно анализировался множественный доступ к каналу.

Основное внимание при анализе частного было уделено характеристикам заполненности канала, так как если канал пуст или заполнен до отказа, запрос может покинуть систему неудовлетворенным. В реальной системе такой запрос снова станет в очередь с остаточной длиной, но в этом случае данные могут быть записаны, либо считаны не непрерывно. Если это обстоятельство не влияет на работу программы, то никаких факторов ухудшающих работу канала нет и канал можно использовать. Если же не непрерывное размещение данных приводит к некорректной работе программы, то следует учитывать следующее. При увеличении разницы интенсивностей поступления запросов на чтение и на запись вероятность того, что канал пуст либо заполнен до отказа возрастает. Существенное влияние на параметры функционирования канала оказывает также объем канала и функции распределения для заказываемых объемов на считывание и на запись. Если заказываемые объемы не намного меньше объема канала, то высока вероятность некорректной работы канала. В этих случаях использовать программный канал не рекомендуется и целесообразнее воспользоваться другим средством коммуникации.

Что же касается множественного доступа к каналу, то вероятности состояний, приводящих к некорректной передаче данных, либо «состязанию» процессов за овладение каналом уже для $N_1>1, N_2>1$ оказались достаточно высокими, поэтому для множественного доступа использование коммуникативной техники каналов не рекомендуется.

### Заключение

Таким образом, была построена модель для стационарного режима функционирования программного канала в виде краевой задачи для системы линейных интегральных уравнений типа Вольтерра. На основании полученной модели была промоделирована работа программных каналов при различных входных данных, и по результатам исследований были выработаны наиболее общие рекомендации по их использованию. В дальнейшем данная модель может быть использована для реализации надстройки многопроцессной операционной системы, для автоматизации выбора того или иного средства коммуникации между процессами самой системой, что полностью освободит данный аспект программирования от субъективного фактора.

### Литература

1. Клейнрок Л. Теория массового обслуживания. –М.:Машиностроение, 1979.-432 с.
2. Maurice J. Bach THE DESIGN OF THE UNIX OPERATING SYSTEM
3. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. – М.: Наука, 1970. –720с.
4. Верлань А.Ф., Сизиков В.С. Методы решения интегральных уравнений с программами для ЭВМ. – Киев: Наукова думка, 1978. –292 с.