

Н. В. М е д в е д е в, А. Ю. Б ы к о в,  
Г. А. Г р и ш и н

## **ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ С ИСПОЛЬЗОВАНИЕМ МЕЖПЛАТФОРМЕННОЙ БИБЛИОТЕКИ ФУНКЦИЙ ЯЗЫКА СИ++**

*Рассмотрены основные подходы к использованию инструментальных средств имитационного моделирования, а также проблема их использования для операционных систем типа МСВС и Linux. Для имитационного моделирования систем массового обслуживания предложено использовать библиотеку функций языка Си++. Приведены особенности и краткое описание данной библиотеки, являющейся межплатформенной, и особенности использования в библиотеке метода понижения дисперсии. Показан пример имитационного моделирования классической системы массового обслуживания типа М/М/п.*

**Обзор подходов к использованию инструментальных средств имитационного моделирования в Windows и мобильной системе вооруженных сил (МСВС).** Для исследования сложных систем массового обслуживания (СМО) широко используется имитационное моделирование (ИМ) [1]. Для операционных систем (ОС) Windows 95/98/ME/NT/XP/2000 реализовано множество инструментальных систем ИМ. Можно выделить два основных подхода к созданию и использованию средств ИМ.

Первый подход основан на использовании инструментальных средств на базе специализированных языков ИМ. Наиболее широкое распространение получил язык ИМ GPSS [2], ориентированный на дискретно-событийное моделирование [1]. На основе данного языка для ОС типа Windows разработано инструментальное средство GPSS World. В ОС типа Windows существуют и другие языки и средства ИМ, например Arena, Extend и др. [1].

Второй подход основан на использовании универсальных языков программирования типа Си, Си++, Pascal, Basic и т.п. С помощью данных языков разрабатываются специализированные библиотеки функций и/или процедур для целей ИМ. Данные подходы имеют свои преимущества и недостатки.

Основная особенность специализированных языков ИМ заключается в том, что с их помощью достаточно просто решить задачи ИМ, но очень трудно реализовать (или иногда невозможно) решение других задач, например многочисленных вычислительных задач. При разработке же специального программного обеспечения (СПО) для упра-

вления сложными организационно-техническими системами требуется разрабатывать комплексы программ, в которых, кроме задач ИМ, должно решаться множество вычислительных задач. Кроме того, следует учитывать, что на базе специализированных языков ИМ разработано множество инструментальных средств для ОС типа Windows, но практически такие средства отсутствуют в ОС мобильной системы вооруженных сил (МСВС).

В настоящее время поставлена задача — перейти от использования несертифицированных ОС типа Windows на ОС МСВС, являющиеся разновидностями ОС Linux и сертифицированные для применения в ВС РФ. Основной проблемой при этом является разработка нового СПО, включающего в себя также программные продукты для целей ИМ. На основе этого можно сделать вывод о том, что проблема разработки инструментальных средств ИМ для ОС МСВС (Linux) является актуальной.

Примером второго подхода к ИМ является применение библиотеки процедур и функций, разработанной на языке Pascal специально для ИМ [3]. Библиотека называется SIMPAS и реализована под ОС типа DOS и Windows. Данная библиотека по своим возможностям в области ИМ и простоте разработки соответствующих программ приближается к языку GPSS, но, кроме того, имеет дополнительные преимущества. Например, код процедур и функций — открытый, использование дополнительных возможностей языка программирования Pascal, в частности для решения различных расчетных задач.

Существуют определенные трудности при использовании данной библиотеки в ОС МСВС. Главная трудность заключается в том, что для МСВС не существует сертифицированного средства разработки на основе языка Pascal. Главный язык разработки, входящий в состав ОС МСВС, язык Си (Си++). Компилятор данного языка входит в состав МСВС и позволяет создавать приложения, выполняемые из командной строки (в режиме терминала). Кроме того, в МСВС входит средство разработки приложений на основе языка Си++ для графической оболочки XWindows-KDevelop [4]. На основе данных соображений для ИМ в среде МСВС является целесообразным использовать библиотеку функций языка Си++.

**Обзор библиотеки SimC.** Библиотека — функция для ИМ, реализованная для ОС типа Windows и МСВС, называется SimC. Причем исходные коды файлов данной библиотеки одинаковые как для ОС типа Windows, так и для ОС МСВС. Это удалось добиться благодаря использованию специальных средств языка Си — директив препроцессора. В частности, используются директивы условной компиляции. Например, при компиляции для ОС МСВС системные функции, входящие в Windows, но отсутствующие в МСВС, заменяются на соответствующие системные функции МСВС.

При разработке библиотеки SimC использовался такой же подход, как и при создании библиотеки SIMPAS. Возможности двух библиотек практически одинаковые. Названия большинства функций библиотеки SimC соответствуют названиям операторов GPSS, выполняющим подобные действия. Результаты при моделировании получаются близкими к результатам, полученным при моделировании с использованием GPSS. При использовании библиотеки SimC предусмотрена возможность сохранения результатов моделирования в текстовом файле.

Кроме того, в библиотеку введена дополнительная возможность для обработки результатов моделирования, заключающаяся в том, что можно использовать метод понижения дисперсии, основанный на дополняющих случайных числах [1]. Так, в библиотеке SimC для создания воспроизводимых потоков псевдослучайных чисел используется линейный конгруэнтный генератор [1]. Особенность реализации функций для получения чисел, распределенных по различным законам, заключается в том, что можно получать дополняющие (“противоположные”) выборки случайных чисел [1], имеющие между собой отрицательную корреляцию.

Суть метода дополняющих выборок заключается в следующем. Основной генератор случайных чисел, равномерно распределенных в интервале от 0 до 1, может работать в двух режимах. В режиме генерации прямой последовательности на  $i$ -м шаге генератор выдает некоторое значение  $U_i$ , а в режиме генерации дополняющей последовательности на  $i$ -м шаге генератор формирует значение  $1 - U_i$ . Таким образом, получаются последовательности псевдослучайных чисел с отрицательной корреляцией. С помощью данного генератора и определенных преобразований [1] можно получить две отрицательно коррелированных последовательности случайных чисел, распределенных по практически любому заданному закону распределения.

Механизм использования метода дополняющих выборок следующий. С моделью выполняются два эксперимента одинакового объема (продолжительности интервала моделирования). В первом эксперименте используются прямые последовательности случайных чисел, а во втором — дополняющие последовательности. Значение оценки определяемого в ходе экспериментов с моделью параметра берется как среднее арифметическое значений оценки этого параметра, полученных в ходе двух экспериментов. Имеется достаточно простое доказательство того, что в этом случае дисперсия оценки параметра будет меньше, чем дисперсия оценки, полученной в результате одного эксперимента [1].

**Установка и использование библиотеки SimC в Windows и МСВС.** Рассмотрим порядок установки и применения библиотеки функций SimC в среде Visual C++ под Windows и в среде KDevelop под МСВС. Исходные тексты библиотеки находятся в двух текстовых

файлах: SimC.cpp (файл реализации функций и определения переменных) и SimC.h (файл описания функций и переменных).

В Visual C++ 6.0 необходимо создать проект типа “Win32 Static Library” (создание статической библиотеки функций). В данный проект необходимо включить файлы с исходными кодами SimC.cpp и SimC.h, предварительно их можно записать в каталог проекта. Затем необходимо откомпилировать проект, в результате будет получен файл SimC.lib. Далее для работы потребуются два файла: SimC.lib (файл объектных кодов функций библиотеки) и SimC.h, эти файлы можно записать в отдельный каталог, например в каталог C:\SimC. Для использования библиотеки функций в каком-либо проекте необходимо в файлах с исходным кодом языка Си++, в которых используются функции библиотеки SimC, с помощью директивы include подключить заголовочный файл SimC.h, например,

```
#include < c : \SimC\SimC.h > .
```

В свойствах проекта (элемент главного меню “Project|Settings”) на вкладке “Link” в текстовое поле с заголовком “Object/library modules:” необходимо записать имя файла библиотеки C:\SimC\SimC.lib.

В ОС МСВС среда разработки KDevelop настраивается следующим образом. Исходные файлы SimC.cpp и SimC.h необходимо поместить в любой каталог пользователя; в режиме терминала, если каталог с файлами является рабочим, вводится команда для компиляции `g++ -c SimC.cpp`, в результате получаем файл с объектным кодом SimC.o. Затем создаем архивный файл библиотеки SimC.a командой `ar -q SimC.a SimC.o`.

Далее создаем каталог `/usr/lib/SimC` и помещаем в него файл SimC.a, файл SimC.h помещаем в каталог `/usr/include`.

При работе в среде KDevelop для использования библиотеки SimC необходимо выполнить следующее: в настройках проекта (пункт меню Проект|Options), во вкладке “Опции редактора связей”, в текстовое поле “Дополнительные библиотеки” записать `usr/lib/SimC/SimC.a`. К тексту программы подключить заголовочный файл SimC.h командой `include`.

**Постановка задачи моделирования работы вычислительного центра.** В качестве примера использования библиотеки SimC и для сравнения результатов моделирования с результатами, полученными с использованием языка GPSS, рассмотрим пример моделирования следующей системы массового обслуживания. Вычислительный центр состоит из 4 ПЭВМ. Задачи на решение поступают в среднем через 5 минут, время между задачами распределено по экспоненциальному закону. Каждая задача решается любой свободной ПЭВМ или ставится в очередь на решение, если все ПЭВМ заняты, длина очереди

неограниченная. Каждая задача решается в среднем 15 минут, время решения распределено по экспоненциальному закону. Требуется в ходе ИМ определить загрузку вычислительного центра (среднее число занятых ПЭВМ) и параметры очереди задач (среднюю длину очереди и среднее время ожидания задачи в очереди).

Для сравнения различных результатов ИМ будем проводить в течение суток, 10 и 100 суток, года, 10 лет. Если обозначить как  $T$  длительность интервала времени моделирования, то для устранения влияния переходных процессов в системе (начальное состояние в системе — задачи отсутствуют) моделирование будем проводить в течение интервала длительностью  $2T$ . После первого интервала длительностью  $T$  осуществим сброс статистики, таким образом, статистика будет накапливаться на втором интервале длительностью  $T$ .

**Решение задачи и результаты моделирования.** Данная система относится к классу СМО типа М/М/п [5] (пуассоновский входной поток заявок, пуассоновский поток обслуживания,  $n$  каналов обслуживания). Для данного класса систем существует аналитическое решение [5], которое позволяет проверить результаты ИМ. Основные соотношения имеют следующий вид:

$$p_0 = \left[ 1 + \rho + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n!(n-\rho)} \right]^{-1},$$

где  $p_0$  — вероятность того, что в системе нет ни одной заявки (не решается ни одной задачи),  $\rho = \lambda/\mu$  — приведенная интенсивность;  $\lambda = 0,2$  задачи в минуту — интенсивность входного потока заявок;  $\mu = 1/15$  задачи в минуту — интенсивность обслуживания,  $n = 4$  — число каналов обслуживания;

$$\bar{k} = \frac{\lambda}{\mu} = \rho,$$

где  $\bar{k}$  — среднее число занятых каналов обслуживания (ПЭВМ);

$$\bar{Q}_{\text{очер}} = p_0 \frac{\rho^{n+1}}{n!n(1-\rho/n)^2},$$

где  $\bar{Q}_{\text{очер}}$  — средняя длина очереди;

$$\bar{T}_{\text{очер}} = \frac{1}{\lambda} \bar{Q}_{\text{очер}},$$

где  $\bar{T}_{\text{очер}}$  — среднее время ожидания заявки (задачи) в очереди.

Подставив заданные значения, получаем результаты:

$$\bar{k} = 3, \quad \bar{Q}_{\text{очер}} \approx 1,5283, \quad \bar{T}_{\text{очер}} \approx 7,6415 \text{ мин.}$$

Сравним с данными результатами результаты ИМ, полученные с использованием библиотеки SimC в среде KDevelop под MSVC, и результаты, полученные в среде GPSS World под Windows.

Текст программы на языке Си++ с использованием библиотеки SimC с комментариями представлен в листинге 1.

Листинг 1

Текст программы на языке Си++

```
#include <SimC.h>
int main()
{
    PSTORAGE pS; // Указатель на многоканальное устройство
    PQUEUE pQ; // Указатель на очередь
    Start(2); // Задаем число прогонов - 2
    NewStorage(pS, "Выч. сеть", 4); // Создаем многоканальное устройство
    NewQueue(pQ, "Очередь"); // Создаем очередь
    InitGenerate(1, 0); // Инициализация генератора транзактов - задач
    InitGenerate(8, 60*24*3650); // Инициализация генератора транзактов -
        // прогонов модели, транзакт поступает через 10 лет

    while(TG1) // Цикл обработки событий
    {
        Plan(); // Протяжка модельного времени

        switch(SysEvent) // Обработка событий
        {
            case 1: Generate(RandExp(0.2, V2)); break; //Поступает транзакт - задача
            // Для получения дополняющей последовательность необходимо записать
            // RandExp(0.2, V2, 1)
            case 2: Queue(pQ); break; // Занимаем очередь
            case 3: Enter(pS); break; // Занимаем многоканальное устройство
            case 4: Depart(pQ); break; // Освобождаем очередь
            case 5: Advance(RandExp(1./15, V3)); break; // Задержка - решение задачи
            // Для получения дополняющей последовательность необходимо записать
            // RandExp(1./15, V3, 1)
            case 6: Leave(pS); break; // Освобождаем многоканальное устройство
            case 7: Terminate(); break; // Уничтожение транзакта - задачи
            case 8: Generate(60*24*3650); break; // Транзакт поступает через 10 лет
            case 9: if (TG1==2) Reset(); break; // Сброс статистики
            case 10: Terminate(1); // Уничтожение транзакта
        }
    }

    PrintAll(); // Печать результатов моделирования
    return 1;
}
```

Результаты моделирования на языке Си++ сохраняются в текстовом файле, вариант результатов моделирования в течение интервала,

длительностью в 10 лет, с использованием прямых последовательно-случайных чисел представлен в листинге 2.

Листинг 2

Результаты моделирования на Си++

```
СИСТЕМНОЕ ВРЕМЯ      : SYSTIME= 10512000.000
СИСТЕМНОЕ СОБЫТИЕ   : SYSEVENT= 10
                        TRANS=NULL
ВСЕГО ВЫПОЛНЕНО СОБЫТИЙ : EVENTALL=15786072
ЗАТРАЧЕНО ВРЕМЯ      : REALTIME= 13.00 СЕК
СКОРОСТЬ ИСПОЛНЕНИЯ  : EVESPEED=1283733.593560 СОБ/СЕК
СРЕДНЕЕ ВРЕМЯ ИСПОЛНЕНИЯ СОБЫТИЯ : MEANTIME= 0.000001 СЕК/СОБ
```

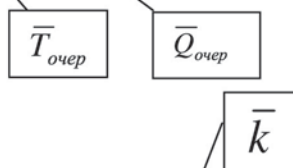
```
СОБ.# ВСЕГО СОБ.# ВСЕГО СОБ.# ВСЕГО СОБ.# ВСЕГО СОБ.# ВСЕГО
1 1051045 3 1051045 5 1051045 7 1051045 9 1
2 1051045 4 1051045 6 1051045 8 1 10 2
```

```
*****
*
* ---- ОЧЕРЕДИ ----
*
*****
```

ОЧЕРЕДЬ	ЧИСЛО С 0 ВРЕМ.ОЖ.	ВХОДОВ ТЕКУЩ.ДЛИНА	МАКС. ДЛИНА	СРЕДН. ВР. ОЖ. БЕЗ УЧ. 0 ВХ.	СРЕДНЯЯ ДЛИНА	% ВХОДОВ В ПУСТУЮ ОЧЕРЕДЬ
Очередь	1051045	34		7.641	1.528	49.183
	516934	0		15.036		

```
*****
*
* --- НАКОПИТЕЛИ ---
*
*****
```

НАКОПИТЕЛЬ	ЕМКОСТЬ	ЗАГРУЗКА	СРЕДН. ВРЕМЯ ПРЕБЫВАНИЯ	ТЕКУЩЕЕ СОДЕРЖ.	МАКС. СОДЕРЖ.	СРЕДНЕЕ СОДЕРЖ.	ЧИСЛО ВХОДОВ
Выч. сеть	4	0.749	14.992	2	4	2.998	1051045



Результаты моделирования в течение интервалов различной длительности при прогонах с прямыми и дополняющими последовательностями случайных чисел и усредненные результаты, а также результаты, полученные с помощью GPSS World, приведены в таблице.

**Выводы.** Анализ результатов ИМ с использованием библиотеки SimC для ОС MCBC или Windows показывает их близость к результатам, полученным в ходе ИМ на GPSS при тех же исходных данных. Также полученные результаты при достаточно большом интервале времени моделирования близки к результатам, полученным на аналитических моделях. Кроме того, использование метода понижения дисперсии, основанного на дополняющих случайных числах, позволяет получить более точные результаты. Таким образом, библиотека функ-

### Результаты имитационного моделирования

Время моделирования	Среднее число занятых каналов: $\bar{k} = 3$				Средняя длина очереди: $\bar{Q}_{очер} \approx 1,5283$				Среднее время ожидания в очереди: $T_{очер} \approx 7,6415$ мин			
	Прямые последовательности	Обратные последовательности	Среднее значение	Результаты, полученные на GPSS	Прямые последовательности	Обратные последовательности	Среднее значение	Результаты, полученные на GPSS	Прямые последовательности	Обратные последовательности	Среднее значение	Результаты, полученные на GPSS
Сутки	3,237	2,714	<b>2,975</b>	<b>2,920</b>	2,76	1,35	<b>2,055</b>	<b>0,591</b>	12,77	8,069	<b>10,42</b>	<b>2,973</b>
10 суток	2,817	3,081	<b>2,949</b>	<b>3,088</b>	1,03	1,581	<b>1,305</b>	<b>1,204</b>	5,349	7,821	<b>6,585</b>	<b>5,984</b>
100 суток	3,009	3,01	<b>3,009</b>	<b>2,986</b>	1,553	1,595	<b>1,574</b>	<b>1,162</b>	7,769	7,938	<b>7,853</b>	<b>5,817</b>
Год	2,992	2,991	<b>2,991</b>	<b>2,994</b>	1,472	1,543	<b>1,507</b>	<b>1,448</b>	7,391	7,705	<b>7,548</b>	<b>7,272</b>
10 лет	2,998	2,996	<b>2,997</b>	<b>3,002</b>	1,535	1,517	<b>1,526</b>	<b>1,524</b>	7,685	7,597	<b>7,641</b>	7,618



ций языка Си++, реализованная под Windows и МСВС, пригодна для ИМ систем массового обслуживания.

## СПИСОК ЛИТЕРАТУРЫ

1. К е л ь т о н В., Л о у А. Имитационное моделирование. Классика СС. 3-е изд. – СПб.: Питер; Киев: Изд. группа BHV, 2004. – 847 с.
2. К у д р я в ц е в Е. М. GPSS World. Основы имитационного моделирования различных систем. – М.: ДМК Пресс, 2004. – 320 с.
3. М а р к о в А. А. Моделирование информационно-вычислительных процессов: Учебное пособие для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 1999. – 360 с.
4. С е к у н о в Н. Ю. Программирование на Си++ в Linux. – СПб.: БХВ-Петербург, 2003. – 368 с.
5. К л е й н р о к Л. Теория массового обслуживания. – М.: Машиностроение, 1979. – 432 с.

Статья поступила в редакцию 23.03.2005

Николай Викторович Медведев родился в 1954 г., окончил в 1977 г. МВТУ им. Н.Э. Баумана. Канд. техн. наук, зав. кафедрой “Информационная безопасность” МГТУ им. Н.Э. Баумана. Автор 45 научных работ в области исследования и разработки защищенных систем автоматической обработки информации.

N.V. Medvedev (b. 1954) graduated from the Bauman Moscow Higher Technical School in 1977. Ph. D. (Eng.), head of “Data Safety” department of the Bauman Moscow State Technical University. Author of 45 publications in the field of study and development of secured systems of automatic data processing.



Александр Юрьевич Быков родился в 1969 г., окончил в 1990 г. ВИКИ им. А.Ф. Можайского. Канд. техн. наук, доцент кафедры “Информационная безопасность” МГТУ им. Н.Э. Баумана. Автор 20 научных работ в области информационной безопасности, имитационного моделирования.

A.Yu. Bykov (b. 1969) graduated from the Military Academy n.a. A.F. Mozhaisky in 1990. Ph. D. (Eng.), assoc. professor of “Data Safety” department of the Bauman Moscow State Technical University. Author of 20 publications in the field of data safety, imitation modeling.

Георгий Александрович Гришин родился в 1979 г., окончил МГТУ им. Н.Э. Баумана в 2003 г. Ассистент кафедры “Информационная безопасность” МГТУ им. Н.Э. Баумана. Автор 4 научных работ в области информационной безопасности.

A.G. Grishin (b. 1979) graduated from the Bauman Moscow State Technical University in 2003. Assistant of “Data Safety” department of the Bauman Moscow State Technical University. Author of 4 publications in the field of the information safety.

