

## ПОСТРОЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ПРОГРАММНО- ТЕХНИЧЕСКОГО КОМПЛЕКСА

*В работе описывается предложенная математическая модель производительности программно-технического комплекса, использование формальных методов в системе анализа построения информационной системы. Показан пример использования формального языка в системе анализа производительности.*

*Ключевые слова: производительность, информационные системы, программно-технические комплексы, формальные языки, расширенная форма Бэкуса-Наура, нотации Вирта.*

О.М.KHOSHABA

Vinnitsa National Technical University

## CONSTRUCTION OF INFORMATION SYSTEM FOR COMPUTER SIMULATION PERFORMANCE

*In article describes the proposed mathematical model of productivity software and hardware complex, the use of formal methods in the analysis of the construction of information systems. In the article shown an example of the use of formal language in the system performance analysis.*

*Keywords: performance, information systems, formal languages, Extended Backus–Naur Form, Wirth syntax notation.*

### Постановка проблемы

К основным этапам компьютерного моделирования относятся [1-3]: 1) постановка задачи, описание исследуемой системы, выявление ее компонентов и закономерных взаимодействий; 2) формализация, то есть создание математической модели, представляющей собой систему уравнений и отражающей сущность исследуемого объекта; 3) разработка алгоритма, реализация которого позволит решить поставленную задачу; 4) написание программы на конкретном языке программирования; 5) планирование и выполнение вычислений на ЭВМ, доработка программы и получение результатов; 6) анализ и интерпретация результатов, их сопоставление с эмпирическими данными.

Разработка компьютерной модели объекта исследования носит итеративный характер. В общем случае, упрощенные этапы проведения процесса компьютерного моделирования можно описать следующим образом. Сначала на основе имеющейся информации об объекте  $S$  строится модель  $M_1$ , проводится серия вычислительных экспериментов, результаты анализируются. Затем, при получении дополнительной информации об объекте  $S$  учитываются новые факторы, получается модель  $M_2$ , расчеты основных показателей которой тоже исследуются на вычислительной системе с заданной точностью. Процесс анализа результатов исследования повторяют до тех пор, пока требуемые значения точности не совпадут с заданными. Однако, такой этап как постановка задачи играет важную роль в процессе компьютерного моделирования, от которого зависит построение самой модели и точности результатов исследования. Поэтому, правильная и качественно разработанная постановка задачи непосредственным образом влияет на точность проведения компьютерного моделирования. Важными аспектами проведения компьютерного моделирования также являются использование формальных методов для спецификации, проектирования, разработки и сопровождения программных средств, применение тестового инструментария.

### Актуальность исследования

Проектирование, разработка и сопровождение информационных систем (ИС) невозможна без решения вопроса производительности программно-технических комплексов (ПТК). Эффективная разработка и сопровождение ИС в области изучения производительности ПТК требуют: определение модели предметной области и формирование спецификаций к проектированию данных систем; создание и генерацию заданных тестовых нагрузок; построение системы с использованием жизненного цикла и определения бизнес-процессов функционирования основных компонент.

Под формальными методами понимаются группы техник, основанных на математическом аппарате для спецификации, проектирования, разработки, сопровождения и тестирования ИС. Данные техники также позволяют разработать бизнес-процессы функционирования ИС и реализовать жизненный цикл ПТК.

### Цель исследования

Целью исследования является построение ИС для компьютерного моделирования ПТК, которая решается с помощью следующих задач: построение математической модели ПТК, применение формальных методов в системе анализа ПТК, построение ИС с целью генерации тестовых нагрузок на ПТК.

### Построение математической модели ППТК

Опишем математическую модель ППТК законом функционирования:

$$Y(t) = F(X, V, H, t),$$

где  $X = (x_1, x_2, \dots, x_n)$  – вектор входных воздействий;

$Y = (y_1, y_2, \dots, y_m)$  – вектор выходных сигналов (откликов, реакций);

$V = (v_1, v_2, \dots, v_k)$  – вектор воздействий внешней среды;

$H = (h_1, h_2, \dots, h_l)$  – вектор собственных параметров системы.

Тогда, функционирование данной системы можно рассматривать как последовательную смену состояний (фаз)  $q_1(t), q_2(t), \dots, q_r(t)$  производительности [4], которым соответствуют некоторые точки в многомерном фазовом пространстве. Множество всех точек  $A = \{a_1, a_2, \dots, a_s\}$ , отвечающих всевозможным состояниям системы [4], называют пространством состояний объекта (или модели).

Любой ПТК имеет некоторую функциональную структуру, производительность которой зависит от совокупности параметров основных четырех составляющих: процессора, оперативно-запоминающего устройства (ОЗУ), жесткого диска (ЖД), сетевого окружения и их производных (контроллера жесткого диска, видеоадаптера и т.д.) и описывается моделью REQS:

$$\text{ППТК} = \langle R, E, Q, S \rangle, \quad (1)$$

где

R (Resours) - ресурсы;

E (Events, tasks) - события (задачи);

Q (Queries) - запросы;

S (Structure) - структура.

Основополагающим элементом модели REQS (1) являются ресурсы (Resours), так как от их состояния всецело зависит производительность ПТК. Структура (Structure) состоит из трех уровней каждого из основных компонентов ресурса (табл. 1) и их производных (контроллеров, видеоадаптеров ПТК и т.д.).

Таблица 1.

Уровни структуры модели REQS

Уровни	Объект исследования (устройства)	Предмет исследования (показатели, некоторые представители)
I	Процессор	mflops (million instructions per second), flop/s, ips (instructions per second)
	Оперативная память	Частота памяти в MHz, передача данных в секунду MT/s или MHz, пиковая скорость передачи данных Мбайт в секунду
	Жесткий диск	Чтение и запись в bite/s
	Сетевые устройства	bite/s, packets/s, errors/s, dropped/s, overruns/s, collisions/s
II	Процессор	Процент загрузки и простоя, пропускная способность, время ответа, общее количество переключений контекста в секунду
	Оперативная память	Общее количество килобайт, которые система выгрузила на диск в секунду, количество страниц памяти, освобожденных системой в секунду
	Жесткий диск	Количество секторов, прочитанных или записанных в устройство (размер сектора 512 байт), средний размер (в секторах) запросов к устройству
	Сетевые устройства	Скорость создания сокетов, передача дейтаграмм по сети
III	Аппаратные и сетевые устройств	Производные от прикладных программ (функции, производные, транзакции)

Первый уровень структуры модели REQS (1) состоит из показателей измерений аппаратных и сетевых устройств ПТК (табл. 1). К сетевым устройствам ПТК в данном случае относят сетевые интерфейсы. Показатели измерений сетевых устройств первого уровня соотносят с физическим уровнем модели TCP/IP (или OSI). Второй уровень структуры модели REQS (1) составляют показатели функционирования ПТК. Для сетевых устройств второй уровень соответствует каналному, транспортному и сетевому уровням модели TCP/IP. Третий уровень структуры модели REQS (1) обеспечивает взаимодействие прикладных программ, баз данных или сетевых сервисов с нижележащими уровнями посредством составных частей этих средств: функций, процедур, транзакций и т.д. Для сетевых устройств третий уровень соответствует прикладному

уровню модели TCP/IP. Такое разделение ресурсов на уровни необходимо для изучения ППТК с помощью таких профессиональных инструментов как DTrace или Systemtap.

Ключевым элементом модели REQS (1) являются запросы (Queries), с которыми связано достаточно большое количество неправильных интерпретаций результатов ППТК. Зная особенности работы с запросами в вычислительных системах удается в несколько раз повысить ППТК [5].

К не менее важному элементу модели REQS относятся события (Events, tasks), которые тесно связаны с процессами ПТК и делятся на системные и пользовательские (прикладные). Процессы, возникающие в результате событий, инициатором которых является система или пользователи, являются связующим звеном взаимодействия верхнего (третьего) уровня модели REQS (1) с нижележащими. Важность процессов проявляется также и в сборе результатов исследований ППТК, механизмы которых на сегодняшний день достаточно хорошо разработаны.

### Использование формальных методов в системе анализа ППТК

Формальные методы представляют собой широкий класс фундаментальных техник теоретической информатики, которые могут быть применены к системе анализа ППТК, общая структура которой показана на рис. 1.

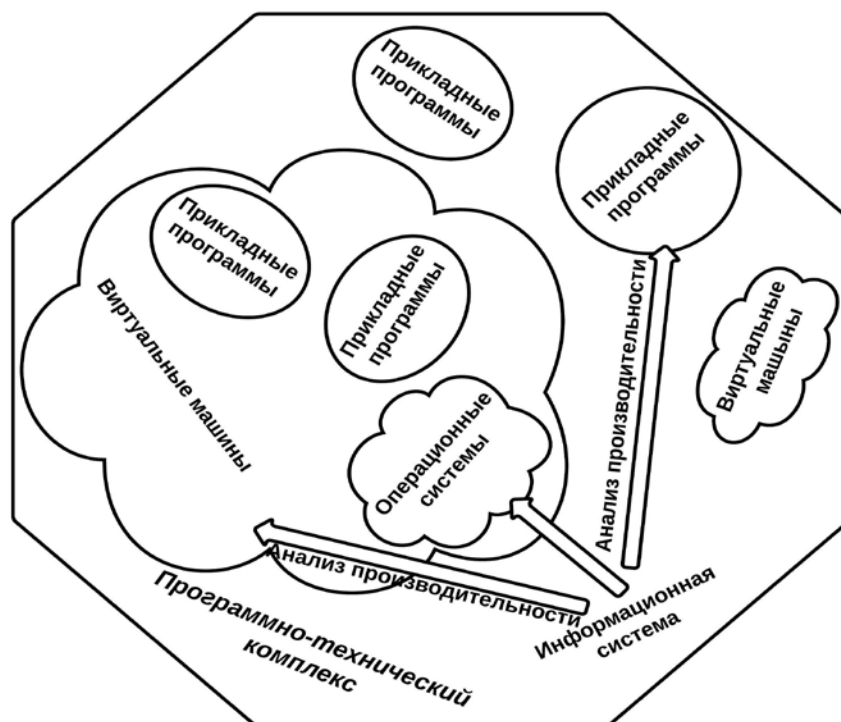


Рис.1. Общая схема анализа производительности программно-технического комплекса

Данная схема (рис.1) показывает возможность проведение анализа ППТК и его основных компонентов для изучения прикладных программ (ПП), операционных систем (ОС) и виртуальных машин (ВМ). Формальные методы, которые используются для проектирования данной ИС в составе ПТК (рис. 1), могут включать в себя исчисления логики, формальные языки, теорию автоматов, формальную семантику, системы алгебраических типов данных и т.д. [6]. В то же время, теория формальных языков предполагает использования множества цепочек, которые задаются одним из следующих способов [7]: перечислить все цепочки языка; указать процедуру порождения цепочек; определить метод распознавания допустимых цепочек. Для описания процедуры создания таких цепочек используются порождающие формальные грамматики, которые с успехом используются в описании синтаксиса языков программирования в формах Бэкуса-Наура (BNF). В этом случае определяются четыре элемента [7]: множество терминальных символов; множество нетерминальных символов; множество произведений (правил грамматики); целевой символ (называемый также стартовым, начальным или аксиомой грамматики). Такая грамматика записывается в виде:

$$G(V_T, V_N, P, S), \tag{2}$$

где

$V_T$  - обозначает множество терминальных символов;

$V_N$  - множество нетерминальных символов;

$P$  - множество произведений;

S - целевой символ.

В дальнейшем, форму Бэкуса-Наура дополняли, появлялись усовершенствованные формы: расширенная форма Бэкуса-Наура (EBNF) и нотации, наиболее удобные для компьютерной обработки: Wirth syntax notation (WSN) [8].

### Пример использования формального языка в системе анализа ППТК

Формальные методы предусматривают разработку формальных спецификаций, которые используются для написания программного кода ИС. Наибольший эффект использования формальных методов достигается при проектировании ИС, когда существует возможность детализировать будущую структуру любого уровня.

Использование формального языка при построении модуля ИС по генерации нагрузок необходимо для обработки входного файла с целью построение специальной программы или скрипта (более подробно — рис. 3). Исходя из грамматики (2) кратко опишем модель REQS (1) с показателями функционирования ПТК от устройства «процессор» из табл.1, используя нотацию WSN:

```
performance = { (level_I | level_II | level_III) } .  
level_I = { (proc_I | mem_I | disk_I | netw_I) } .  
level_II = { (proc_II | mem_II | disk_II | netw_II) } .  
level_III = { (proc_III | mem_III | disk_III | netw_III) } .  
proc_I = { ("p11" | "p12" | "p13" | "p14") } .  
proc_II = { ("p21" | "p22" | "p23") } .  
proc_III = { ("p31") } .
```

Используя данный код существует возможность построения синтетической диаграммы — для нетерминальных (рис. 2а) и терминальных символов (рис. 2б).

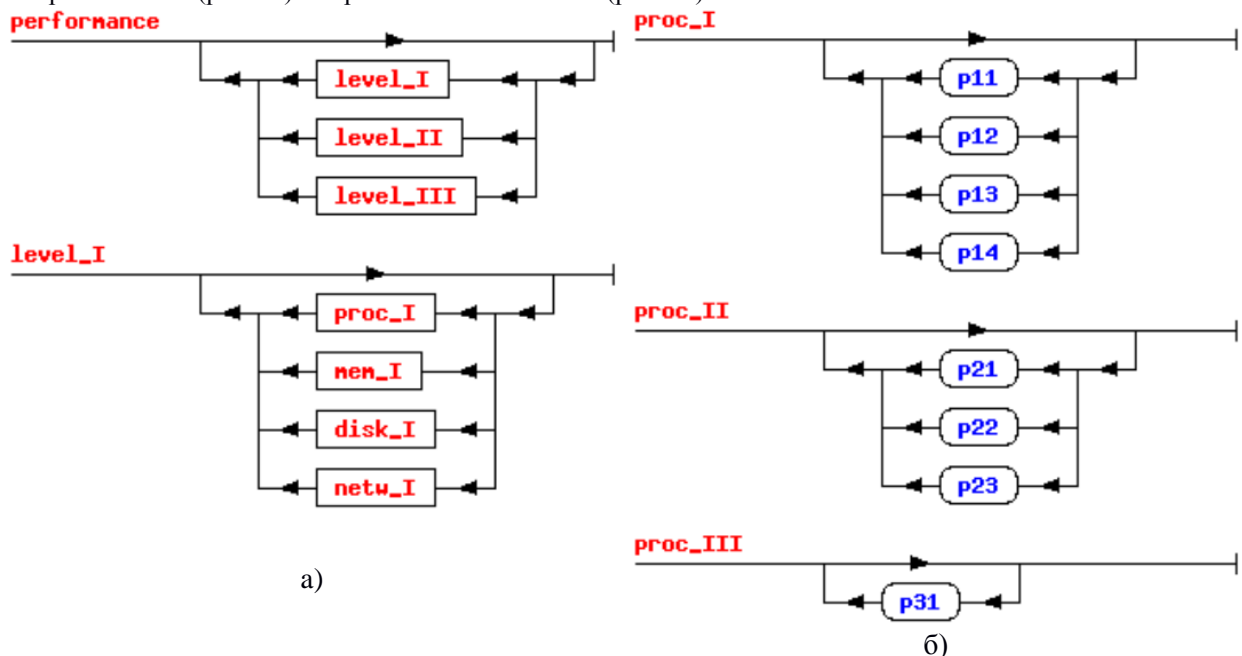


Рис. 2. Синтетическая диаграмма тестовых нагрузок на ПТК, подготовленная с помощью нотаций Вирта (а) — для нетерминальных символов; б) — для терминальных символов)

На рис. 2 показана синтетическая диаграмма тестовых нагрузок на ПТК, подготовленная с помощью нотаций Вирта на которой исходя из правил грамматики (2) нетерминальные символы изображены прямоугольниками, а терминальные — скругленными прямоугольниками. По грамматике анализа ППТК нетерминальные символы представляют собой элементы и устройства ПТК, а терминальные — показатели измерения и функционирования ПТК. Также, используя нотацию WSN, легко построить код XML, с помощью которого легко создавать и управлять объектами программных средств:

```
<syntax meta="xis/ebnf v0.2 http://wiki.perf.com/ebnf/ gpl3" title="EBNF"><rule  
name="performance"><loop><choise><identifier value="level_I"/><identifier  
...  
value="p21"/><terminal value="p22"/><terminal value="p23"/></choise></loop></rule><rule  
name="proc_III"><loop><terminal value="p31"/></loop></rule></syntax>
```

### Построение информационной системы для анализа ППТК

Один из модулей построенной ИС (рис. 3) согласно постановке задачи и используемым нотациям, предназначенных для формирования воздействия тестовых нагрузок на различные объекты ПТК: прикладные программы (ПП), операционные системы (ОС), виртуальные машины (ВМ). Входные

параметры системы создаются с помощью командного файла предписания воздействия рабочих и тестовых нагрузок, подготовленного с помощью разработанного графического средства или текстового редактора. Во входном файле описываются с помощью метаязыка согласно нотациям Вирта компоненты ПТК и особенности нагрузки. В выходном файле генерируется скрипт на языке bash для проведения тестовых нагрузок на различные структуры ПТК.

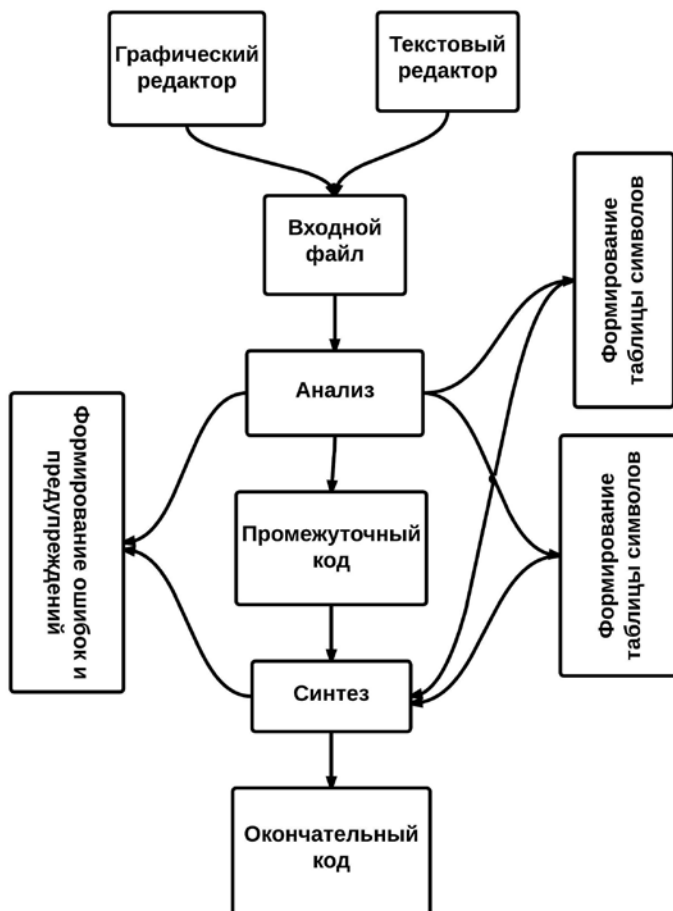


Рис 3. Модуль информационной системы по обработке входной информации

### Выводы

Таким образом, модуль, показанный на рис. 3, состоит из таких основных компонент как анализ и синтез программного кода. В результате проведенного анализа и синтеза программного кода формируется таблица символов. Сообщения об ошибках появляются в случае нарушения правил лексики, синтаксиса или семантики разработанного языка  $G$  (2). Поэтому, анализ программного кода, являющийся основной частью построенной ИС, предусматривает проведение лексического, синтаксического и семантического анализов.

### Список использованной литературы

1. Бусленко Н.П. Моделирование сложных систем / Н.П. Бусленко – М.: Наука, 1968. – 356 с.
2. Шеннон Р. Имитационное моделирование систем: искусство и наука / Р. Шеннон – М.: Мир, 1978. – 302 с.
3. Советов Б.Я. Моделирование систем: Учеб. для вузов / Б.Я. Советов, С.А. Яковлев – М.: Высш. шк., 2001. – 343 с.
4. Хошаба А.М. Математическая модель фаз производительности вычислительных систем / А.М. Хошаба // Четверта Міжнародна науково-практична конференція «Інформаційні технології та комп'ютерна інженерія» Тези допов.- Вінниця, 2014.-С. 301-302.
5. Хошаба А.М. Математическая модель фаз производительности вычислительных систем. / А.М. Хошаба // Вестник Херсонского национального технического университета №3(50), 2014.-Херсон.-С.-523-527.
6. Jean François Monin, Michael Gerard Hinchey, Understanding formal methods, Springer, 2003, P.276.
7. Backus J.W., et al. Revised report on the algorithmic language ALGOL60. Num. Math. 4. P.420-453.
8. [https://en.wikipedia.org/wiki/Wirth\\_syntax\\_notation](https://en.wikipedia.org/wiki/Wirth_syntax_notation).