УДК 681.32.019

**Г.С. ТЕСЛЕР**

# КОНЦЕПЦИЯ ПОСТРОЕНИЯ ГАРАНТОСПОСОБНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

***Abstract:*** *In the paper the conception of building dependable computer systems with taking into account the conceptual positions, elaborated at the last time. Side by side with understanding the term dependability as system and functional reliability it integrates in itself such notions as simple reliability, survivability, availability safety and so on. Existent conception of dependability is completed by regulations, following from cybernetic approach and cinergetic.*
***Key words:*** *reliability, survivability, fault-tolerance dependability, information, stability.*

***Анотація:*** *У статті розглянуто концепцію побудови гарантоздатних обчислювальних систем з урахуванням концептуальних положень, розроблених останнім часом. Поряд з розумінням терміна гарантоздатності як системної і функціональної надійності, він інтегрує в собі такі поняття, як проста надійність, живучість, відмовостійкість, готовність, безпека, робастність і т.д. Існуюча концепція гарантоздатності доповнюється положеннями, які слідують із кібернетичного підходу і сінергетики.*
***Ключові слова:*** *надійність, живучість, відмовостійкість, гарантоздатність, інформація, стабільність.*

***Аннотация:*** *В статье рассматривается концепция построения гарантоспособных вычислительных систем с учетом концептуальных положений, выработанных в последнее время. Наряду с пониманием термина гарантоспособность как системной и функциональной надежности, он интегрирует в себе такие понятия, как простая надежность, живучесть, отказоустойчивость, готовность, безопасность, робастность и т.д. Существующая концепция гарантоспособности дополнена положениями, следующими из кибернетического подхода и синергетики.*
***Ключевые слова:*** *надежность, живучесть, отказоустойчивость, гарантоспособность, информация, стабильность.*

## 1. Введение

Современная наука находится на этапе интеграции, что является объективным отражением общих тенденций ее эволюционного развития. Сам термин «гарантоспособность» появился относительно недавно, и его суть сводится к тому, что гарантоспособность в конечном итоге является средством (технологией), гарантирующим достоверность информации в результате ее преобразования, хранения и передачи компьютерными средствами, невзирая на наличие внешних и внутренних возмущений, воздействующих на работу вычислительной системы. В этой связи обеспечение необходимого уровня гарантоспособности тесно связано с такими понятиями, как надежность, живучесть, отказоустойчивость и другими, хорошо известными современной науке.

Но если бы гарантоспособность сводилась только к данным понятиям, то введение этого нового понятия не имело бы смысла. Дело в том, что приведенные выше понятия в основном связаны с обеспечением работоспособности технических средств вычислительной системы и лишь частично ее программного обеспечения, но при этом не рассматривается весь технологический цикл, связанный с получением достоверной информации, выдаваемой вычислительной системой в результате ее работы.

В настоящее время отсутствует устоявшееся понятие гарантоспособности. Существующие понятия в той или иной степени отражают суть описанной выше проблемы. В общем случае для решения ее необходимо учитывать также исходные составляющие технологии выдачи достоверной информации вычислительной системы. Это, прежде всего, связано с достоверностью входной информации, свойствами используемых методов и порожденных или алгоритмов, перешедших в рабочие программы вычислительной системы. Основным свойством методов, алгоритмов и программ, а также входной информации должна быть адекватность моделируемых

вычислительной системой реальных процессов. Понятно, что основная ответственность обеспечения этой адекватности лежит на разработчиках математического и программного обеспечения. Но не менее важны такие свойства системы, как устойчивость и робастность. Причем эти свойства распространяются не только на математическое и программное обеспечение, но и на технические средства. Обеспечение устойчивости системы связано с ее архитектурой и структурой. В этой связи стоит обратить внимание на то обстоятельство, что природа в результате своего эволюционного развития биологических объектов нашла решение многих проблем, стоящих перед создателями гарантоспособных вычислительных систем и прежде всего путем реагирования на внутренние и внешние возмущения. Понятно, что некоторые из них, такие как клеточное деление т.п., пока неприемлемы для вычислительных средств. Но функциональное дублирование, эффективные методы контроля и мониторинга, наличие большого числа обратных связей заслуживают самого пристального внимания разработчиков гарантоспособных вычислительных систем. Исходя из вышеизложенного, в статье предпринята попытка сформулировать концепцию построения такого класса вычислительных систем и дать свою формулировку понятия их гарантоспособности. Понятно, что данная формулировка не окончательная и будет уточняться по мере развития этого научного направления.

Актуальность этого направления связана со следующим. В работе [1] автор показал все возрастающее значение информации, знаний и высоких технологий в существующем сегодня постиндустриальном информационном обществе. Еще больше важную роль эти проблемы будут играть в обществе знаний, которое постепенно придет на смену информационному. В этой связи приобретают особое значение не только массовость выпуска и расширение сфер использования вычислительных средств, но и их качественные показатели, среди которых находится и гарантоспособность вычислительных систем.

Важную роль фактор гарантоспособности играет в критических областях использования вычислительных средств. К таким областям относятся поддержание и обеспечение технологических процессов на производстве, в системах управления особо опасными объектами, включая АЭС, в банковском и интернетном деле, военном, на транспорте, в энергетике, медицине, связи и т.д. Особо следует отметить роль гарантоспособности при работе автономных мобильных вычислительных средств и, прежде всего, бортовых, ремонт которых во многих случаях невозможен. Например, в системах управления спутниками, что может привести к их катастрофе.

Но даже в обычных компьютерах типа персональных и серверах, работающих в технологической цепи супермаркетов, и т.д., нестабильная работа ОС причиняет много неприятностей пользователям. Это относится к наиболее массово используемой ОС Windows фирмы Microsoft. Именно поэтому грамотные пользователи в подобных случаях используют Unix подобные ОС и ОС Linux. Но это решение только небольшой части проблем гарантоспособности. Приведенный пример показывает актуальность рассматриваемой проблемы как в настоящее время, так еще больше в недалеком будущем.

**2. Постановка проблемы**

Как уже отмечалось во введении, проблема гарантоспособности является комплексной. При ее решении целесообразно максимально использовать все средства и методы, выработанные наукой

и практикой для достижения высокого уровня надежности, живучести и отказоустойчивости вычислительных систем. Как известно [2], условием для этого является использование различных видов избыточности: аппаратной, структурной, информационной, алгоритмической, программной, функциональной, нагрузочной, эксплуатационной, надежностной, семантической и вероятностной (статистической). При этом необходимо использовать адаптацию к условиям применения, включая элементно-технологический, организационный, информационный, алгоритмический и эксплуатационный базисы вычислительной техники и, кроме необходимых, следует удовлетворять также условиям обеспечения высокого уровня отказоустойчивости, которыми являются эффективные методы и средства контроля и мониторинга правильности функционирования системы, а также парирование (амортизация) возникших отказов и сбоев [2].

Особую роль при этом играет функциональная избыточность, которая позволяет интегрировать аппаратную, информационную, семантическую и программную избыточность, что чрезвычайно важно для обеспечения гарантоспособности. Естественно, не всей, а только ее части.

Для достижения высокого уровня гарантоспособности играют и информационная, и семантическая избыточность, а также их контроль. Последний требует привлечения таких средств, как распознавание нечётких образов и нейроподобных систем, а также статистических методов. Такой подход является не совсем привычным для обеспечения высокого уровня живучести и отказоустойчивости вычислительных систем. Но использование подобного подхода не исключает использование в нашем случае и привычных для этих областей методов контроля и мониторинга. Чрезвычайно важной для обеспечения гарантоспособности является нормальная работа всех элементов вычислительной системы, которая непосредственно связана с устойчивостью работы как технических средств, так и программного и информационного обеспечения. Для технических средств это прежде всего связано со структурной устойчивостью, теория которой даже для традиционных областей, какой является теория управления, сегодня разработана недостаточно [3, 4]. Что касается численных методов, то данная проблема частично решена в рамках решения некорректно поставленных задач [5]. В статистических методах она решается в рамках создания и использования робастных методов и алгоритмов [6].

Для обеспечения высокого уровня гарантоспособности, как уже отмечалось выше, необходимо использование методов и средств, применяемых в биологических и физиологических системах вообще и в человеке, в частности. В этой связи для обеспечения высокого уровня гарантоспособности целесообразно использовать кибернетический подход, связанный с развитием нейрокомпьютеров, семантических и целенаправленных саморазвивающихся систем вообще и синергетики в частности [7–11]. Такой подход, совместно с методами и средствами, используемыми для достижения высокого уровня надежности, живучести и отказоустойчивости, позволит решить большинство задач по повышению уровня гарантоспособности вычислительных систем.

### 3. Концептуальные положения гарантоспособности вычислительных систем

Отметим, что недостаточность либо отсутствие внимания к уровню гарантоспособности вычислительных систем ведёт к громадным экономическим потерям и оказывает вредное

психическое и психологическое влияние на их пользователей. Однако сегодня этому пока уделяют очень мало внимания, что частично объясняется достаточной надёжностью вычислительной техники, а это обусловлено использованием высоких технологий при её изготовлении. Но учитывая, что потребность в росте производительности вычислительных средств, их усложнение и расширение сфер использования не имеет видимых границ, а также установка их в поддержании различных технологических процессов на производстве, выдвигает на передний план обеспечение их свойством гарантоспособности.

Гарантоспособность прежде всего связана с достоверностью получаемой из вычислительной системы информации и связана с нормальной (штатной) её работой, невзирая на наличие допустимых внутренних и внешних возмущений, т.е. система имеет определенный запас устойчивости (стабильности).

В биологических и кибернетических системах такая стабильность достигается путем гомеостазиса. Напомним, что гомеостазис – особое свойство живых организмов удерживать существенные переменные характеристики организма (температуры, давления крови, содержания сахара, гормонов, кислорода в крови и т.д.) в допустимых для его существования пределах при обеспечении оптимального режима внутренней среды. Представление гомеостазиса тесно связано с понятиями ультраустойчивости и адаптивности. Стремление организма удерживать существенные переменные в физиологических пределах связаны с процессами саморегуляции на основе обратных связей. Эти процессы направлены на ликвидацию последствий возмущений в тех или иных подсистемах организма. Гомеостазис высшего уровня имеет вероятностный характер и связан с поиском адекватности планов и структуры физиологических актов организма условиям внешней среды, а гомеостазис, связанный с внутренними системами или локальными участками нервной системы, носит детерминистский характер [11]. Примером кибернетической системы, использующей принцип работы на гомеостазисе, является автопилот.

В работе [12] уделяется большое внимание разным аспектам гомеостазиса в физиологических системах. Так, системный гомеостазис определяется как поддержание постоянства структурно-функциональной организации физиологической системы как целой, независимого постоянства её параметров и выполнения системой необходимых функций. При этом независимое относительно динамическое постоянство параметров внутренней сферы организма представляет собой параметрический гомеостазис, а постоянство выполнения системной функции – функциональный гомеостазис.

Постоянство усложнения системных функций и функционального гомеостазиса при восхождении от нижнего уровня иерархии к верхнему представляет собой еще один вид гомеостазиса – системно-иерархического. И поэтому эти три вида гомеостазиса являются теми механизмами, которые обеспечивают системный гомеостазис организма как целого.

Всё вышеизложенное, с концептуальной точки зрения, может быть перенесено на вычислительные системы, естественно, со своими саморегулирующими состояниями и регулирующими механизмами.

Особое место в этом классе систем занимают целенаправленные системы.

В общем случае класс целенаправленных систем для достижения необходимого уровня гарантоспособности характеризуется наличием множества целей, в том числе вспомогательных, промежуточных и основных, которые могут быть связаны союзами «и» либо «или». Основные принципы стабильности информационных семантических систем следующие [9]:

1. Информационная семантическая система считается стабильной, если цели (цель) достигается. Понятно, что без эффективного мониторинга этот факт установить затруднительно либо невозможно.

2. В процессе завершения семантического диалога семантические объекты могут оказаться «совместимыми» или «несовместимыми» между собой. В нашем случае мы расширяем понятие семантического диалога, который обычно включает человека и машину на межмашинный и даже межблочный диалог между элементами системы. Понятно, что эти элементы должны быть хоть в какой-нибудь мере интеллектуальными.

3. Несовместимость семантических объектов является причиной нестабильности информационной семантической системы (ИСС). Такая несовместимость проявляется «в непонимании» приёмником информации, поступающей от источника и принятии приёмником неправильного решения на основе полученной информации.

4. Нестабильная ИСС может стать стабильной, если ввести внешний семантический объект, совместимый хотя бы с одним семантическим объектом системы.

5. Информационная семантическая система является стабильной, если она внутренне совместима и внешне не изолирована.

При этом необходимо отметить, что все объекты материального мира являются в той или иной степени семантическими вследствие того, что они выполняют функции источников семантической информации (принцип отражения) для интеллектуальной системы.

Гарантоспособность с точки математических методов и соответствующих программ во многом зависит от их устойчивости и сходимости, качеств, влияющих на устойчивость счёта по алгоритму, а также корректность постановки исследуемых задач по Адамару и условной корректности и регуляризации по А.Н. Тихонову [5].

Гарантоспособности в этом случае благоприятствуют:

– использование итерационных процессов, у которых окончательные погрешности метода и округления зависят от последней итерации;
– использование двойных алгоритмов типа повышенного порядка аппроксимаций и сегментных аппроксимаций с невысоким порядком аппроксимирующих выражений звеньев;
– использование интервального подхода;
– использование релаксационных подходов;
– использование комбинированных подходов типа факторизация оператора и использование метода последовательных приближений;
– использование схем типа предикатор-корректор;
– использование ортогонального базиса либо методов ортогонализации;
– использование адекватных норм погрешностей;
– применение методов и понятий теории вероятности для решения некорректных по Адамару задач и т.п.

В отличие от надежности и отказоустойчивости технических средств надежность программных систем в основном зависит от наличия статических и динамических ошибок в программе. Основными средствами борьбы с наличием статических ошибок в программе являются:

верификация – доказательство соответствия программы её спецификации [13], организационные методы разработки, различные технологии программирования и использование различных инструментальных средств. Обнаружение динамических ошибок в программе основывается на моделировании программ, что позволяет не только оценить степень их надежности, но и автоматизировать создание исходного программного кода, полностью соответствующее рассматриваемой модели. Такой подход позволяет отслеживать попадание программной системы в необходимое состояние.

В качестве аппарата моделирования используются сети Петри [14] и их модификации, основанные на объекте технологии [15], ориентированные на тесную взаимосвязь процессов и данных типа UML (Unified Modeling Language). При этом является важной оценка надежности программы [16]. Для оценки степени безошибочности выполнения программы используется метод псевдоотладки, основанный на выявлении искусственно введенных ошибок [36]. Напомним, что надежность программной системы, так же, как и технической, характеризуется способностью безотказно выполнять заданные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью [17].

**4. Основные положения гарантоспособности**

Как уже отмечалось выше, гарантоспособность вычислительных систем представляет собой интеграцию средств, обеспечивающих высокий уровень надежности, живучести, отказоустойчивости, готовности, безопасности и т.д. Но при этом предполагается, что эта интеграция представляет собой не простое суммирование методов и средств перечисленных выше частей понятия гарантоспособности, а получение нового качества.

В работах [18–29] приведены различные аспекты понятия гарантоспособности и его некоторых элементов. Другое определение гарантоспособности, которое автор нашёл, следующее [30]:

«Гарантоспособность (dependability) – это свойство вычислительной системы, позволяющее обоснованно полагаться на выполнение услуг, для которых она предназначена. Услуги, предоставляемые системой, отображаются её поведением, которое предопределяется пользователями; пользователь рассматривается как еще одна система, взаимодействующая с первой».

В работе [36] dependability трактуется как функциональная надежность, гарантоспособность, которая обеспечивает получение достоверных результатов в условиях наличия неисправностей. На наш взгляд, dependability является не только функциональной, но и системной надежностью.

Более современное определение гарантоспособности приведено в [27], где под гарантоспособностью понимается системное свойство для интеграции свойств надежности, готовности, безопасности, живучести, ремонтоспособности (восстанавливаемости) и целостности (интегрируемости) и утверждается, что это является презентацией суммирования фундаментальных концепций для обеспечения гарантоспособности. В этой работе отмечается, что защита и выживаемость сложных информационных систем, внедренных в инфраструктуру,

поддерживающую современное общество, является национальной и мировой проблемой наивысшего приоритета.

В работе [30] проблема гарантоспособности рассматривается несколько с других позиций. На рис. 1 приведена блок-схема основных составляющих гарантоспособности.

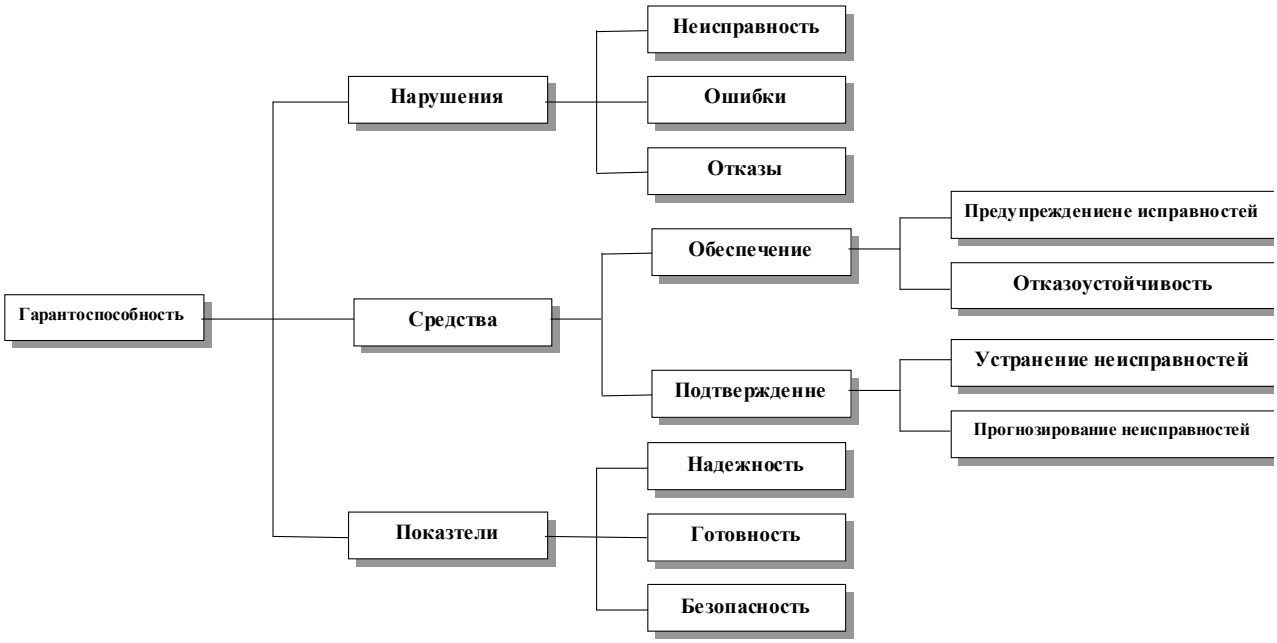


Рис.1. Основные составляющие гарантоспособности

Напомним кратко основные понятия, предшествующие появлению понятия гарантоспособности [31-33].

Живучесть – способность вычислительной системы сохранять работоспособность при выходе из строя части её оборудования.

Отказ – неспособность устройства продолжать автоматическую работу.

Сбой – случайная, нерегулярная ошибка в вычислительном оборудовании; в современных вычислительных системах последствия сбоя устраняются, как правило, автоматически.

Наработка на отказ (сбой) – математическое ожидание времени работы вычислительной системы между двумя последовательными отказами (сбоями).

Контроль – процесс определения правильности работы вычислительной системы и её компонент. Различают контроль работоспособности, аппаратный и программный.

Целостность (интегрируемость) – это отсутствие неприемлемых для системы альтернативных состояний, что является необходимым условием для готовности, надежности и безопасности.

Отказоустойчивость – свойство вычислительных систем, обеспечивающее постоянную их работоспособность и выполнение выполняемых функций при наличии разнообразных отказов и сбоев компонент.

Надёжность (reliability) – свойство систем, машин, аппаратуры и программ выполнять возложенные на неё функции в заданных условиях эксплуатации с заданными показателями качества, достоверностью результатов, пропускной способностью, временем работы и т.д. при сохранении

значений основных параметров в установленных пределах. Оценивается вероятностью безотказной работы компонент системы или системы в целом при выполнении заданного объема работы или в течение заданного промежутка времени.

Готовность (availability) – степень готовности системы в момент, когда она требуется для обработки данных.

Безопасность (security) – защита, обеспечение безопасности данных и программ от несанкционированного доступа.

Ремонтопригодность – способность функционального блока или системы при заданных условиях использования сохранять или восстанавливать состояние, в котором они могут осуществлять требуемые функции, если выполнено обслуживание при данных условиях и использовании установленных процедур и ресурсов.

Как видно из разд. 3 статьи, основные положения вышеприведенных понятий остаются неизменными, но, учитывая необходимость реализации биологозоологического подхода для обеспечения гарантоспособности в вычислительной системе необходимо рассматривать её как кибернетическую систему.

В первом приближении можно дать следующее определение гарантоспособности.

Под гарантоспособностью вычислительных систем будем понимать способность системы правильно, достоверно и устойчиво работать путем сохранения её рабочих параметров в заданных диапазонах их изменения и правильного функционирования при решении задач, невзирая на возникшие внутренние и внешние возмущения.

В случае возникновения в вычислительной системе отказов и сбоев технических и программных средств устойчивость и правильность функционирования обеспечивается путем использования избыточности, соответствующих архитектурно-структурных решений, необходимых уровней отказоустойчивости и надежности, робастности алгоритмов, программных и технических средств, эффективной системы контроля и мониторинга правильности работы системы в контрольных точках, наличия элементов самоорганизации системы, рационального использования необходимых видов избыточности, правильного реагирования на неадекватную информацию на входе и выходе как отдельных элементов системы, так и системы в целом за счет использования робастных процедур, входных фильтров, обратных связей и т.п.

Основным показателем уровня гарантоспособности может служить получение на выходе вычислительной системы с заданным уровнем достоверности оценки вероятности отсутствия ошибок выходной информации. Правильность работы вычислительной системы во многом адекватна понятию работоспособности, т.е. свойству вычислительной системы, которое состоит в её способности выполнять возлагаемые на неё функции с заданными показателями качества и эффективности [35].

В свою очередь, устойчивость вычислительной системы определяет способность системы сохранять в заданных пределах количественные и качественные характеристики при возникновении внутренних и внешних возмущений.

Для обеспечения устойчивости необходимо обеспечить адаптивность и управляемость системы, а иногда и её элементов. При этом понятие адаптивности определяет способность

системы изменять структуру и параметры функциональных элементов с целью приспособления к изменяющимся внутренним и внешним условиям функционирования. В основном это достигается за счет реконфигурации структуры и в некоторых случаях архитектуры системы, в программном обеспечении использования дублирующих копий и других подходах. Реконфигурация в нашем случае – процесс перераспределения и в некоторых случаях перекоммутации функциональных элементов (ФЭ) с целью поддержания работоспособности системы и обеспечения её дальнейшего функционирования после возникновения отказа или сбоя (амортизация отказа или сбоя). В свою очередь, управляемость выражает способность системы в целом и её ФЭ адекватно реагировать на воздействие внутренней и внешней среды. Например, ФЭ верхнего иерархического уровня управляет ФЭ нижнего иерархического уровня [34].

Кроме отказов, сбоев и ошибок в работе вычислительной системы, с чем еще должна бороться гарантоспособная система? Это, прежде всего, следующие свойства, качества информации: неопределенность, неточность, неполнота, нечеткость, несвоевременность, недостоверность и противоречивость. Значение этих терминов приведем согласно работе [34]:

Неопределенность – свойство, которое указывает на наличие нескольких альтернативных отражений процесса.

Неточность – свойство, которое указывает на наличие определённого интервала допусков или погрешности измерений или расчёта в количественных параметрах и/или качественных характеристиках отражения процесса.

Неполнота – свойство, которое указывает на наличие информационных проблем в отражении процесса (что-то пропущено, описано недостаточно и т.д.).

Нечеткость – свойство, которое характеризует расплывчатость отражения процесса, при которой невозможно точно указать наличие или отсутствие определённого свойства или его точную количественную характеристику.

Несвоевременность – свойство, которое характеризует соотношение во времени между моментом наступления какого-то события и получения информации о нём (не позволяющее принять своевременное решение).

Недостоверность – свойство, которое указывает на наличие количественных данных или качественных характеристик, которые не соответствуют истинному состоянию ситуации.

Противоречивость – свойство, которое указывает на наличие количественных или качественных характеристик, которые имеют значение или смысл, противоречащий другим данным.

Гарантоспособная вычислительная система должна функционировать таким образом, чтобы адекватно реагировать на все внутренние и внешние возмущения (отклонения), возникшие в системе или повлиявшие на её работу с целью выдать интересующую потребителя информацию с заданным уровнем достоверности.

Такая постановка задачи требует следующего:

1. Вместо традиционного рассмотрения элементов, узлов, блоков и т.д. – рассмотрения выполняемых системой функций.
2. Совместного рассмотрения отклонений от заданных значений функций, возникших в системе, включающих сбои и ошибки, возникающие в технических средствах, программном обеспечении, методах и алгоритмах решаемых задач, а также в информации (входной, внутренней и выходной).

3. Обеспечения устойчивости функционирования всех компонент, перечисленных в п.2 и определения границ области устойчивости.

4. Эффективных систем локального и комплексного контроля правильности функционирования системы.

5. Использования эффективных методов повышения уровня надежности, живучести и отказоустойчивости, разработанных в рамках этих подходов.

6. Использования достижений работы системы в области устойчивости в рамках структурной устойчивости, устойчивости работы алгоритмов и математико-логических методов, а также информационной устойчивости и достоверности.

7. Использования эффективных методов управления системой парирования и амортизации отказов, включая адаптивное управление с использованием модели.

8. Использования средств саморазвития, разработанных в рамках кибернетических систем и синергетики [37].

9. Использования эффективных архитектурно-структурных решений на основе существующей избыточности системы.

10. Использования принципов биологического баланса и смешанного экстремума для сбалансирования требований системы.

11. Подхода обеспечения высокого уровня отказоустойчивости, подобно биолого-зоологическому, включая и человека.

Устойчивость работы вычислительной системы тесно связана с расширенным понятием робастности. Следуя [35], робастность (robustness) – мера способности вычислительной системы восстанавливаться при возникновении ошибочных ситуаций как внешнего, так и внутреннего происхождения. Например, в робастной системе допускаются ошибки во входных данных или неисправности каких-либо составных частей самой этой системы. Хотя между надежностью и робастностью может существовать определённая связь, это две различные характеристики системы: система, которая никогда не будет восстанавливаться при возникновении ошибочных ситуаций, может быть надежной, не будучи робастной; система с высокой степенью робастности, которая восстанавливается и продолжает работу во множестве ошибочных ситуаций, может быть отнесена к ненадежным, поскольку она не способна заблаговременно, до повреждения выполнить необходимые служебные процедуры.

Понятие восстановления – процесс возврата к нормальной работе после сбоя или отказа. Процесс восстановления системы, в частности, в рабочее состояние после возникновения ошибки может включать в себя возврат в предшествующее состояние, зафиксированное резервной копией, и запуск соответствующих программ восстанавливается.

Поэтому устойчивость работы системы будет зависеть не только от уровня её устойчивости, но и робастности.

Не менее важно для обеспечения гарантоспособности сделать эти системы целенаправленными и самоорганизующимися. Это и будет приближать данные вычислительные системы к биолого-зоологическим системам.

Таким образом, для обеспечения высокого уровня гарантоспособности вычислительных систем необходимо:

1. Рассмотрение понятия гарантоспособности как пересечение понятий робастности и устойчивости систем, а также синергетики, в части управления самоорганизующихся систем вообще и перехода системы от состояния, близкого к энтропийному и детерминированному.

2. Для предотвращения отказов системы и прогноза мест, где они могут возникнуть, необходима оценка остаточного ресурса отдельных элементов системы и системы в целом, а также организации эффективного мониторинга и контроля системы.

3. Мониторинг и контроль, помимо того, что они должны обеспечить контроль за тем, чтобы параметры системы и её элементов оставались в данных диапазонах функционирования, а также обязаны осуществлять контроль системы в контрольных точках, обеспечивать управление фазами вычислительного процесса, которые являются заключительной фазой этого управления.

4. Для повышения эффективности мониторинга и контроля, а также дополнительного обеспечения системы сетевыми каналами связи, использовав электрическую сеть как внутри вычислительной системы, так и вне её для организации сетевого взаимодействия между элементами вычислительной системы и системы в целом.

5. Использование функционального дублирования за счет функционально- аппаратурно-информационной избыточности.

6. Для контроля и мониторинга системы широко использовать средства распознавания образов.

7. Обеспечение системно-иерархического гомеостазиса вычислительной системы для обеспечения её отказоустойчивости.

8. Превращение вычислительных систем в целенаправленные и саморазвивающиеся.

9. Реализация принципа биологического баланса для оптимизации ресурсов системы для достижения необходимого уровня гарантоспособности, основанного на отказоустойчивости, в зависимости от надежности компонент вычислительной системы и частоты её работы.

10. Осуществление системного подхода по целям, задачам, ресурсам, оперативности и результативности взаимодействия компонент вычислительной системы в условиях возникновения внутренних и внешних возмущений.

В связи с тем, что понятие гарантоспособности появилось сравнительно недавно, то концепция построения гарантоспособных вычислительных средств находится только в начальной стадии своего развития.

## 5. Выводы

1. Создание вычислительных систем с высоким уровнем гарантоспособности является одной из найприоритетнейших для информационного общества.

2. Гарантоспособность является интегрирующим средством, объединяющим такие понятия, как надёжность, живучесть, отказоустойчивость и т.п.

3. Гарантоспособность охватывает не только технические средства и программное обеспечение, но и математические методы, алгоритмы, информационный базис.

4. Существующая концепция обеспечения высокого уровня гарантоспособности дополнена положениями, следующими из кибернетического подхода и синергетики.

## СПИСОК ЛИТЕРАТУРЫ

1. Теслер Г.С. Новая кибернетика. – Киев: Логос, 2004. – 404 с.
2. Теслер Г.С. Концепция создания вычислительных средств с высоким уровнем отказоустойчивости // Математические машины и системы. – 2002. – №2. – С. 176–183.
3. Автоматизация производства и промышленная электроника. Энциклопедия современной техники / Под ред. А.И. Берга и В.А. Трапезникова. – М.: Изд-во «Советская энциклопедия», 1965. – Т. 4.– С. 223.
4. Пятницкий Е.С. О структурной устойчивости одноконтурных систем регулирования нормального типа // Автоматика и телемеханика. – 1963. – Т. 24. – С. 5.
5. Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач. – М.: Наука, 1986.
6. Горбань І.І. Теорія ймовірностей і математична статистика для наукових працівників та інженерів. – Київ: ІПММС НАНУ, 2003. – 244 с.
7. Галинская А.А. Модульные нейронные сети: обзор современного состояния разработок // Математические машины и системы. – 2003. – № 3, 4. – С. 87–102.
8. Теслер Г.С. Перспективы развития вычислительных средств с сетевым взаимодействием // Математические машины и системы. – 2004. – № 1, 2. – С. 3–11.
9. Соломатин Н.М. Информационные семантические системы. – М.: Высшая школа, 1989. – 127 с.
10. Научно-технический прогресс: Словарь / Сост. В.Г. Горохов, В.Ф. Халипов. – М.: Политиздат, 1987. – 366 с.

11. Энциклопедия кибернетики / Под ред. В.М. Глушкова. – Киев: Главная редакция украинской советской энциклопедии, 1974. – Т. 2. – С. 232.
12. Биоэкомедицина: единое информационное пространство / Авт. В.И. Гриценко, М.И. Вовк, А.Б. Котова и др. – Киев: Наукова думка.
13. Надежное программное средство как продукт технологии программирования спецификации // http:// sp.cs.msu.ru.– 1998.
14. Котов В.Е. Сети Петри. – М: Наука, 1984. – 160 с.
15. Семенец С.В., Рудой В.В. Реализация решений по реализации надежности программного обеспечения // Математические машины и системы. – 2002. – №1. – С. 128–133.
16. Кабак И.С., Позднеев Б.М. Оценка надежности объектно-ориентированного программного обеспечения // Труды конференции XXV Юбилейная междунар. конф. и дискуссион. научн. клуб «Новые информ. технологии в науке, образовании и бизнесе». – Ялта. –1998. – С. 539–541.
17. Романюк С.Г. Оценка надежности программного обеспечения // http:// www.nilsi.ru / pub 02. htm.– 1994.
18. Jones A. The challenge of building survivable information-intensive systems" // IEEE Computer. – 2000. – Vol. 33, N 8. – P. 39–43.
19. Avizienis A. Design of fault-tolerant computers // Proc. 1967 AFIPS Fall Joint Computer Conf., AFIPS Conf. Proc. – 1967. – Vol. 31. – P. 733–743.
20. Bouricius W.G., Carter W.C., Schneider P.R. Reliability modeling techniques for self-repairing computer systems // Proc. 24th National Conference of ACM. – 1969. – P. 295–309.
21. Randell B. System structure for software fault tolerance // IEEE Transctions on Software Engineering. – 1975. – Vol. SE-1, N. 10. – P. 1220–1232.
22. Avizienis A., Chen L. On the implementation of N-version programming for software fault tolerance during execution // Proc. IEEE COMPSAC 77. – 1977. – November. – P. 149–155.
23. Laprie J.C. Dependable computing and fault tolerance: concepts and terminology // Proc. 15th IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-15). – Ann Arbor, Michigan. – 1985. – June. – P. 2–11.
24. Dobson J.E., Randell B. Building reliable secure computing systems out of unreliable insecure compnents // Proc. 1986 IEEE Symp. Security and Privacy. – Oakland, Calif. – 1986. – April. – P. 187–193.
25. Fray J.M., Deswarte Y., Powell D. Intrusion tolerance using fine-grain fragmentation-scattering // Proc. 1986 IEEE Symp. Security and Privacy. – Oakland, Calif. – 1986. – April. – P. 194–201.
26. Joseph M.K., Avizienis A. A fault tolerance approach to computer viruses // Proc. 1988 IEEE Symposium on Security and Privacy. – Oakland, Calif. – 1986. – April. – P. 52–58.
27. Avizienis A., Laprie J.-C., Randell B. Dependability of computer systems: Fundamental concepts, terminology, and examples // LAAS Report No., UCLA Report No., Newcastle No. – 2000. – October.
28. Landwehr C.E. A taxonomy of computer program security flaws // ACM Computing Surveys. – 1994. – Vol. 26, N3. – September. – P. 211–254.
29. Information Technology Security Evaluation Criteria (ITSEC), Commission of The European Communities, Office for Official Publications of the European Communities. – 1991. – June.
30. Avizienes A. Dependable computing: From concepts to design diversity // IEEE Proc. – 1986. – Vol. 74, № 5. – P.17–49.
31. Заморин А.П., Марков А.С. Толковый словарь по вычислительной технике и программированию. – М.: Рус. язык, 1988. – 221 с.
32. Справочник-словарь терминов АСУ / Авт. В.И. Вьюн, А.А. Кобозев, Т.А. Паничевская, Г.С. Теслер / Под ред. д.т.н. Ю.Е. Антипова, чл.-корр. АН УССР А.А. Морозова. – М.: Радио и связь, 1990. – 128 с.
33. Вычислительная техника и обработка данных. Терминологический толковый словарь IBM.– М.: Статистика, 1978. – 232 с.
34. Згуровський М.З., Панкратова Н.Д. Системный анализ: проблемы, методология, приложения. – Киев: Наукова думка, 2005. – 744 с.
35. Толковый словарь по вычислительным системам / Под ред. В. Иллингуорта и др. – М.: Машиностроение, 1991. – 500 с.
36. Англо-русский словарь по вычислительной технике. Компьютеры, мультимедиа, сети, интернет, телекоммуникации / Под ред. М.Л. Гушкина. – М.: ЭТС, 1998. – 496 с.
37. Хакен Р.Б. Синергетика. – М.: Мир, 1980. – 324 с.