Особенности организации процесса функционального диагностирования управляющего компьютера повышенной живучести

Владимир Г. Зарубский, *Пермский институт Федеральной службы исполнения наказания России, Пермь, Россия, e-mail: volen3030@rambler.ru*

Владимир Г. Зарубский

Резюме. **Цель.** *Надежность охраны объектов особой важности и повышенной опасности на сегодняшний день достигается применением интегрированных систем безопасности с интеграцией подсистем на базе управляющих компьютеров. Повышение живучести специализированных компьютеров является актуальной задачей, решение которой может лежать в использовании компьютеров, обладающих свойством структурной устойчивости. Практическая реализация такого компьютера связана с решением задачи его функционального диагностирования и последующей функциональной адаптацией. В статье предлагается к рассмотрению решение задачи функционального диагностирования структурно-устойчивого управляющего компьютера как функциональной системы, принципиально отличающейся от традиционного процесса контроля персонального компьютера, осуществляемого известными программами самопроверки.* **Методы.** *Для решения задачи функционального диагностирования предложена математическая модель тестового контроля, который может стать основой функционального диагностирования управляющего компьютера. Кроме этого, на основании предложенной математической модели осуществлен анализ возможных исходов тестового контроля.* **Результаты.** *В результате анализа предложенной математической модели определены варианты сведения к минимуму рисков I и II рода, то есть причисления неисправных функций к множеству исправных (риск потребителя) и причисления исправных функций к множеству неисправных (риск поставщика), что достигается путем использования общепринятой для электронных вычислительных машин методики диагностирования по принципу «раскрутка». Суть методики заключается в поиске работоспособного «ядра» – набора базовых функций позволяющих в дальнейшем осуществлять диагностирование оставшихся функций системы команд компьютера. Таким образом «ядро» с любым обнаруженным дефектом к дальнейшему функционированию не допускается, а исправное может служить достаточно надежным средством контроля. При использовании данной методики получение нормы одиночного теста не гарантирует отсутствие риска I рода, что объясняет сложившуюся практику проверки каждой функции системы команд достаточной последовательностью проверочных тестов, при этом риск II рода не возрастает.* **Выводы.** *Представленная в статье модель проверочного теста функционального диагностирования позволила сформулировать стратегию построения данного процесса для структурно-устойчивого управляющего компьютера, которая заключается в реализации ряда частных задач таких как: выделение в особый ряд задачи идентификации работоспособного «ядра», как возможной причины зарождения риска I рода, служащей источником возникновения риска II рода; поочередне диагностироване остальной части функций, как в вычислительной среде с развитым свойством постепенной деградации функций; оптимизация расширяющейся последовательности проверочных тестов для каждой уменьшающей риск I рода функции, необратимо приводящая к возрастанию временных границ контроля дефицитных для предэтапной самопроверки; продолжение тестирования при отрицательных результатах, но другой программной реализацией для снижения риска II рода; разработка специальной методики для обоснования продолжительности тестирования каждой функции управляющих компьютеров.*

Ключевые слова: *управляющий компьютер, живучесть, структурная устойчивость, функциональная избыточность, функциональное диагностирование, достоверность контроля, риски I и II рода.*

Формат цитирования: *Зарубский В.Г. Особенности организации процесса функционального диагностирования управляющего компьютера повышенной живучести // Надежность. 2016. №3. С. 35-38. DOI: 10.21683/1729-2646-2016-16-3-35-38*

Введение

Возросшая в последнее время террористическая активность выдвигает повышенные требования к обеспечению надежной защиты объектов особой важности. Для решения этой задачи широкое распространение получили интегрированные системы безопасности (ИСБ). Интеграция подсистем, входящих в состав ИСБ, в большинстве случаев осуществляется на базе управляющего компьютера (УК), в качестве которого используется обычный персональный компьютер (ПК), как правило, импортного производства со «штатной» операционной системой. Очевидно, что выход из строя управляющего компьютера по причине как преднамеренного, так и непреднамеренного характера, приводит к недопустимым изменениям работы всей системы в целом. В данной ситуации надежность охраны объектов особой важности становится весьма сомнительной. В связи с вышеизложенным, идея разработки отечественного УК, обладающего свойствами повышенной живучести к угрозам различного характера, представляется весьма актуальной.

В качестве такого УК может быть предложен компьютер, обладающий свойствами структурной устойчивости [1], принципы работы которого основаны на функциональной избыточности любого современного компьютера. В свою очередь практическая реализация такого компьютера связана с решением двух частных задач – функциональное диагностирование УК и последующая его функциональная адаптация.

Все современные компьютеры являются многоуровневыми устройствами и каждый из этих уровней обладает свойствами функциональной избыточности [2], в связи с чем в данной статье процессы функционального диагностирования будут рассмотрены на примере командного уровня архитектуры.

Функциональное диагностирование структурно-устойчивых (СтУ) УК как функциональной системы принципиально отличается от традиционного процесса контроля ПК, осуществляемого известными программами самопроверки с целью установления вида технического состояния: «исправен – неисправен», «работоспособен – неработоспособен». В современных ПК функциональное диагностирование центрального процессора лишено всякого смысла, поскольку непрохождение любого теста делает нецелесообразным получение ПК готовности, т.к. усеченная система команд становится несоответствующей специальному программному обеспечению. Несколько иначе обстоит дело с групповым оборудованием специальных цифровых вычислительных комплексов систем вооружения, обеспечивающим трехгранную (трехканальную) структуру, особенно эффективную в борьбе со сбоями и их последствиями. Здесь встречаются элементы функционального диагностирования, ставящего целью выявления ряда частных отказов, не препятствующих выполнению боевых задач, но обязательно устраняемых на регламенте ввиду снижения запасов живучести, необходимых в экстремальных условиях эксплуатации. К этой категории отказов относятся: неспособность канала быть ведущим (ведомым) в двухканальной структуре, неспособность мажоритарных механизмов парировать одиночные сбои при вводе информации, полная или частичная утрата функций межканального обмена и др**.** Однако в этом случае имеет место не самоконтроль, а определение технического состояния устройств обмена функциональными средствами центрального процессора.

Функциональное диагностирование структурно-устойчивого управляющего компьютера

Функциональное диагностирование центрального процессора, актуальное для этапа восстановления готовности СтУ УК, по сути является самодиагностированием, т.е. идентификацией функционального состояния $\tilde{\rho}^F$ в условиях стохастически неопределенного разбития функциональной системы F на классы $\tilde{\rho}^F$ и $\overline{\tilde{\rho}}^F$:

$$F = \tilde{\rho}^F \cup \overline{\tilde{\rho}}^F, \tilde{\rho}^F \cap \overline{\tilde{\rho}}^F = \varnothing, \quad (1)$$

при стремлении

$$\tilde{\rho}^F \to \rho_\phi^F, \overline{\tilde{\rho}}^F \to \overline{\rho}_\phi^F, \quad (2)$$

где ρ_ϕ^F – текущее функциональное состояние СтУ УК, и ограничении продолжительности процесса контроля

$$t_{\phi\partial} \le t_{\phi\partial}^{\max}. \quad (3)$$

Выражение (2) означает сведение к минимуму рисков I и II рода, то есть причисления неисправных функций к множеству исправных (риск потребителя) и причисления исправных функций к множеству неисправных (риск поставщика).

В общем случае подобная задача не может иметь удовлетворительных решений ввиду определенной недостоверности уже первичных результатов самопроверки и стремительного ее распространения в последующие процессы контроля. При этом нарушается основной принцип любого процесса контроля, требующий, чтобы средства контроля были более высокого класса, чем объект контроля. Это условие выполняется в СтУ УК с развитым свойством постепенной деградации [3, 4], для которой выбирается методология организации программы самопроверки как средство идентификации текущего функционального состояния электронной вычислительной машины (ЭВМ).

Действительно, на первом этапе процедуры функционального диагностирования осуществляется поиск работоспособного функционально полного «ядра» по общепринятой для ЭВМ методике: «раскрутка» с признанием его неработоспособности по

первому непрохождению проверочного теста. Т.е. «ядро» с любым обнаруженным дефектом к дальнейшему функционированию не допускается. Исправное «ядро» функциональной системы УК может служить достаточно надежным средством контроля для одиночных функций из остальной части системы команд. Этому способствует развитое свойство постепенной деградации функций, предполагающее существование для каждой проверяемой функции обслуживающей только ее части оборудования, для которого можно подобрать достаточно полную последовательность проверочных тестов.

Поскольку полностью снять проблему достоверности процесса контроля функций нельзя, следует проанализировать условия ее повышения при снижении возможных последствий в последующих процессах. Для этого необходима адекватная модель процесса тестового контроля, описывающая элементарные операции контроля и их композиции по части поддержания требуемого уровня достоверности.

Рассмотрим процесс контроля произвольной команды ϑ как функции системы команд θ ЭВМ, организуемый на участке самопроверки с целью разрешения альтернативы

$$\vartheta \in \rho_{\phi}^{F} \mid \vartheta \in \overline{\rho}_{\phi}^{F}, \tag{4}$$

в действительности, превращающийся в разрешение альтернативы

$$\vartheta \in \tilde{\rho}^{F} \mid \vartheta \in \overline{\tilde{\rho}}^{F} \tag{5}$$

на множестве Π_F вариантов разбиения (1).

Исправная функция ЭВМ ϑ определена в конечном дискретном пространстве состояний S ЭВМ, компоненты которого суть ячейки памяти и регистры общего назначения, принимающие всевозможные значения в пределах их разрядности. Это означает, что для любой точки S' произвольного подмножества $S_{\vartheta}, |S_{\vartheta}| \le |X'_{\vartheta} \cup X''_{\vartheta}|$, где $X'_{\vartheta}, X''_{\vartheta}$ – множество входных и выходных переменных команды ϑ, существует образ $S'' \in S_{\vartheta}$, т.е. имеет место преобразование

$$\vartheta : \mathrm{S}' \to \mathrm{S}'' \tag{6}$$

Каждая пара (S', S'') может составить основу проверочного теста, который вместе с объектом контроля ϑ образует оперативную систему контроля, если имеет средства приведения вычислительного процесса в точку $S' - a'_{\vartheta}$ (воздействия на объект контроля ϑ), и средства a''_{ϑ} оценки факта

$$\vartheta(S') = S'', \tag{7}$$

т.е. реакции объекта контроля на заданное воздействие. Как принято, a'_{ϑ} и a''_{ϑ} играют роль средства контроля. В целом построенный тест a_{ϑ} имеет вид

$$a_{\vartheta} = a'_{\vartheta}(\hat{\rho}^{F}) \vartheta a''_{\vartheta}(\hat{\rho}^{F}) = \\ = \vartheta \notin \tilde{\rho}^{F} \mid (\vartheta \in \tilde{\rho}^{F} \mid \vartheta \notin \tilde{\rho}^{F}) \mid \hat{\rho}^{F} \cup \vartheta \tag{8}$$

где $\hat{\rho}^{F}$ – состояние процесса идентификации $\tilde{\rho}^{F}$ перед прохождением теста a_{ϑ}

$$\hat{\rho}^{F} \subseteq \tilde{\rho}^{F}. \tag{9}$$

Результатом процесса контроля ϑ тестом (8) может быть отнесение ϑ к классу $\overline{\tilde{\rho}}^{F}$ при непрохождении теста по признаку (7), неопределенность ($\vartheta \in \tilde{\rho}^{F} \mid \vartheta \notin \tilde{\rho}^{F}$), если процесс контроля ϑ будет продолжен последующими тестами типа (8), либо отнесение ϑ к классу $\tilde{\rho}^{F}$ и подключение данной функции к идентифицированной части функционального состояния $\hat{\rho}^{F}$, если этот тест был завершающим в процессе контроля ϑ (рис. 1).

Несмотря на наличие всего двух исходов каждого отдельно рассматриваемого проверочного теста, внутренних ситуаций в процессе контроля возникает значительно больше (рис. 2). Проведем анализ ситуаций 1-15, происхождение которых связано:

- с надежностью определения работоспособного «ядра», принимаемого в начале за исходную, а затем за текущую идентифицированную часть функционального состояния

$$\hat{\rho}^{F} := \rho^{F}_{\text{Я}\eta}. \tag{2.10}$$

При этом возможны два варианта:

$$\hat{\rho}^{F} \subset \rho^{F}_{\phi} \tag{2.11}$$

и

$$\hat{\rho}^{F} \not\subset \rho^{F}_{\phi}; \tag{2.12}$$

- с надежностью установки задаваемых проверочным тестом исходных данных S'. Для случая (2.11) неизбежно

$$a'_{\vartheta}(\hat{\rho}^{F}) : \tilde{S}' = S', \tag{2.13}$$

где $\tilde{S}'$ – фактический результат выполнения фрагмента теста a'_{ϑ}, построенного на базе подмножества команд $\hat{\rho}^{F}$. Для случая (2.12) из-за несовершенства теста кроме результата

$$a'_{\vartheta}(\hat{\rho}^{F}) : \tilde{S}' \neq S' \tag{2.14}$$

возможен результат (2.13);

- с неопределенностью состояния объекта контроля ϑ, реакции которого на исходные данные S', могут быть как

$$a'_{\vartheta}(\tilde{S}') : \tilde{S}'' = S'', \tag{2.15}$$

всего 5 случаев (1, 2, 4, 5, 8), так и

$$a'_{\vartheta}(\tilde{S}') : \tilde{S}'' \neq S'', \qquad (2.16)$$

4 случая (3, 6, 7, 9), где $\tilde{S}''$ – фактический результат выполнения команды ϑ;

- с надежностью оценки результатов типа (2.15), (2.16) последовательностью команд $a''_{\vartheta}(\hat{\rho}^F)$ проверочного теста: N (норма) для 8 вариантов (1, 2, 4, 6, 8, 10, 12, 14), $\overline{N}$ (ненорма) для 7 вариантов (3, 5, 7, 9, 11, 13, 15).

Нормовые ситуации распадаются на две группы. Первая (1, 4, 10) – соответствует случаю $\vartheta \in \rho^F_{\phi}$. Их объединяет невозрастание риска II рода, поскольку исправная команда идентифицируется как исправная. Вторая группа (2, 6, 8, 12, 14) соответствует случаю $\vartheta \notin \rho^F_{\phi}$ и своим подтверждением нормы способствует нарастанию риска I рода.

Ненормовые ситуации, в свою очередь, распадаются на две группы. Первая (3, 7, 9, 13, 15) – соответствует случаю $\vartheta \notin \rho^F_{\phi}$. Их объединяет невозрастание риска I рода, поскольку неисправная команда идентифицирована как неисправная. Вторая группа соответствует случаю $\vartheta \in \rho^F_{\phi}$ и своим утверждением о ненорме способствует нарастанию риска II рода.

Заключение

Из изложенного в статье следует, что получение нормы одиночного теста не гарантирует отсутствие риска I рода, что объясняет сложившуюся практику проверки каждой функции системы команд достаточной последовательностью проверочных тестов. При этом риск II рода не нарастает. Поскольку нормовые ситуации являются лишь частью полной группы событий 1-15, то можно утверждать, что в последовательности разнообразных проверочных тестов, проходящих по норме, риск I рода уменьшается. Это объясняется тем, что с каждым новым тестом проверяется очередной вариант функционирования аппаратуры, а число непроверенных вариантов сокращается. В случае проверки всех вариантов риска I рода после нормы последнего из них, данный риск полностью исключается. Однако, в случае наличия ограничений на время проверки, данный результат, может быть, не достигнут.

В то время, как было отмечено выше, не устраненный риск I рода порождает риск II рода в форме отрицательного результата теста проверяемой функции при норме последней. В этом случае условия функционального диагностирования ухудшаются. Судя по ситуации 5 исхода теста, положение можно улучшить повтором теста (S', S''), используя иные команды из состава $\hat{\rho}^F$. Отрицательный результат подтвердит неисправность функции ϑ, а положительный – позволит обоснованно перейти к новому функциональному «ядру».

Модель проверочного теста функционального диагностирования позволит сформулировать стратегию построения этого важного для СтУ УК процесса:

1) идентификации работоспособного «ядра» должно быть уделено особое внимание, поскольку на этом этапе зарождается риск I рода, служащий источником возникновения и риска II рода;

2) диагностирование остальной части функций УК следует провести поочередно, как в вычислительной среде с развитым свойством постепенной деградации функций;

3) расширяющаяся последовательность проверочных тестов для каждой функции УК уменьшает риск I рода, однако при этом растет время контроля, дефицитное для предэтапной самопроверки, на которую дополнительно возлагаются задачи адаптации к текущему функциональному состоянию;

4) для снижения риска II рода при отрицательных результатах тестирования, его целесообразно продолжить при тех же исходных данных, но другой программной реализацией;

5) обоснование продолжительности тестирования каждой функции УК требует разработки специальной методики.

Библиографический список

1. Зарубский В.Г. Вопросы разработки перспективных интегрированных систем охраны, отвечающих требованиям повышенной живучести, на базе структурно-устойчивых управляющих компьютеров. Вестник Пермского института ФСИН России. Выпуск 1 (5)/ 2012. С 4-9.

2. Зарубский В.Г., Рыбаков А.П. Математическая модель процесса адаптации управляющего компьютера интегрированной системы к текущему функциональному состоянию. Вестник Воронежского института МВД России. Выпуск 1/2012. С. 170-178.

3. Харитонов В.А. Основы теории живучести функционально-избыточных систем. С.-Пб.: СПИИРАН, 1993. – 60 с.

4. Тюрин С.Ф. Синтез адаптируемой к отказам цифровой аппаратуры с резервированием базисных функций/ Приборы и системы. Управление, контроль, диагностика. Выпуск №1/ 1999. С 36-39

Сведения об авторе

Владимир Г. Зарубский, кандидат технических наук, доцент кафедры, Пермский институт Федеральной службы исполнения наказания России, Пермь, Россия, e-mail: volen3030@rambler.ru

Поступила 10.03.2016