



СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ LICS КАК КОМПОНЕНТ ПОДСИСТЕМ АСУТП АЭС

С.И. Масолкин, В.Г. Промыслов, Е.Ф. Жарко,
А.В. Антонов, О.А. Промыслова, А.С. Степанянц (ИПУ РАН)

Приводятся общие и специальные требования, предъявляемые к системному ПО (СПО) систем верхнего блочного уровня АСУТП АЭС. Обоснован выбор ОС Linux для разработки на ее основе СПО LICS. Приводятся оценка качества СПО LICS и результаты его функционального тестирования.

Введение

АСУТП в последнее десятилетие вышли на качественно новый уровень развития, связанный с возросшим уровнем автоматизации объектов управления и, как следствие, ростом числа диагностических и управляющих сигналов, обрабатываемых системой в единицу времени. С другой стороны, практически линейный рост производительности вычислительных систем, которые могут быть использованы в АСУТП, позволил реализовать значительно более сложные алгоритмы управления и анализа данных.

Произошедший качественный скачок в составе решаемых задач заставил пересмотреть и методы построения АСУТП промышленных объектов, в частности для систем диагностики и управления объектом с повышенной опасностью (в том числе АЭС).

Приведем пример эволюции в области применения системного ПО (СПО). Принятый сравнительно недавно (в 1984 г.) регулирующий документ [1] в своей рекомендательной части существенно ограничивал использование основной компоненты СПО — ОС в вычислительных системах, важных для безопасности АЭС [2]. Положения этого документа широко распространялось вольно или невольно на другие вычислительные системы АЭС. Однако, необходимость построения значительной по вычислительной сложности программы с участием многих компонент, желание отделить части ПО, непосредственно реализующие алгоритмы сбора информации и управления, от вспомогательных программ ведет к разделению программных продуктов на уровне СПО и прикладных программ. Вторым фактором, приводящим к переоценке роли СПО, является необходимость обеспечить срок эксплуатации программно-технического средства (ПТС) из состава АСУТП, значительно превышающий средний достигнутый на данный момент срок службы и хранения технических средств из состава ПТС. Это означает, что должна быть разработана и в процессе эксплуатации АСУТП осуществлена процедура перехода на иные (скорее всего отличающиеся от исходных) ПТС. Сформулированные выше потребности привели к повышению роли СПО в АСУТП в целом и АСУТП АЭС в частности. Данный факт нашел отражение во второй части стандарта, выпущенной в 2000 г. [2].

Сформулируем более формально понятия системного и прикладного ПО.

СПО — ПО, разработанное для специфической вычислительной системы или семейства систем, для облегчения эксплуатации и обслуживания системы и соответствующих программ, например, это операционные системы, компиляторы, утилиты. Системное программное обеспечение обычно состоит из операционной системы и вспомогательных программных средств.

Ограничимся данной формулировкой, хотя часто термин СПО понимается шире и включает любое ПО, не являющееся прикладным.

Прикладное ПО (ППО) (прикладная программа) — программа для выполнения задачи, связанной с контролируемым процессом, а не с функционированием самой вычислительной машины.

В Институте проблем управления РАН было принято решение о разработке собственного СПО под маркой LICS для системы верхнего блочного уровня (СВБУ) АСУТП вновь строящихся и модернизируемых АЭС.

В процессе разработки СПО LICS были выделены следующие фазы:

- 1) определение общих и специальных требований к СПО;
- 2) выбор ОС как основы для дальнейшей разработки;
- 3) анализ эксплуатационной истории кандидата и имеющейся в наличии документации на предмет определения достигнутого качества разработки;
- 4) доработка и расширение функциональности ОС Linux для СПО LICS;
- 5) функциональное и интеграционное тестирование.

Ниже будут рассмотрены более подробно наиболее важные фазы разработки и обобщен накопленный опыт по созданию сложных высоконадежных специализированных программных изделий на основе существующего ПО общего назначения.

Определение общих и специальных требований для СПО СВБУ АСУТП

Выбор ОС для СПО СВБУ АСУТП (далее просто СПО) обусловлен рядом специфических требований, предъявляемых к СПО и определяемых характеристиками объекта управления. В общей форме требова-

Таблица 1. Требуемые функциональные характеристики

Подсистема управления процессами и памятью	Файловая подсистема	Подсистемы ввода/вывода, низкоуровневый, высокоуровневый, ввод/вывод на основе сетевых интерфейсов	Подсистема разработки и модификации ПО	Специальные программные средства
Создание процессов и управление ими	Интерфейс доступа к файлам	Доступ к периферийным устройствам и управление ими с использованием символического и блочного ввода/вывода	Возможность редактирования исходных текстов программных модулей	Средства самодиагностики СПО в части проверки целостности статических элементов (файлов) СПО без учета правильности функционирования компонентов системы
Распределение и управление вычислительными ресурсами	–	Доступ к периферийным устройствам и управление ими с использованием потокового (stream) ввода/вывода	Трансляция исходных текстов программных модулей на языке С и получение их объектных кодов	–
Распределение и управление памятью		–	Отладка программных модулей	
Работа ядра ОС в одно и симметричном двухпроцессорном режимах			Сборка программных модулей	

Таблица 2. Реализация протоколов и программных интерфейсов, требуемых для функционирования ППО

Протоколы	Программные средства	Программные библиотеки	Средства разработки	Дополнительное ПО
TCP/IP	Telnet	Motif	C/C++	Текстовые обработчики (awk, nroff, troff, sed)
X11	Ftp	X11	YACC	WWW браузер
SNMP	NFS		LEX	командные интерпретаторы (C-shell, B-shell)
NTP (RFC 1059, 1119, 1305)	HTTP		Perl	

ния для АЭС изложены в рекомендации европейской комиссии по ядерной энергетике [1, 2] и других нормативных документах [3]. Анализ требований позволил выделить критерии, которым должны удовлетворять ОС в составе СПО:

- возможность реализации своей функции 24 ч в сутки и 365 дней в году;
- обеспечение на штатных технических средствах СВБУ заданных временных характеристик в нормальном и наиболее интенсивном вычислительных режимах;
- высокое качество функций, предоставляемых ОС;
- обеспечение необходимой эксплуатационной документации;
- длительность этапа эксплуатации ОС в составе СПО должна составлять не менее длительности этапа эксплуатации АЭС;
- отсутствие ограничений на поставку на объект управления как со стороны разработчиков ПО, так и законов страны происхождения ПО;
- наличие разрешения на использование ОС со стороны органов и служб, контролирующих поставку на объект (государственных и созданных для строительства и ввода в эксплуатацию данного объекта);
- обеспечение интерфейса между ППО и техническими средствами СВБУ АСУ ТП.

В повседневной практике создания ПО коммерческого и научного характера типична ситуация, когда программа создается для определенной (существующей на рынке) операционной системы (Solrais, MS-DOS, MSWindows и т.д.), т.е. время жизненного цикла ППО меньше, чем время жизненного цикла ОС. Для АЭС характерна ситуация, когда длительность

этапа эксплуатации ППО больше времени эксплуатации технического средства, на котором оно установлено, т.е. возникает потребность замены ОС (или ее обновления) для обеспечения функционирования ППО на новых технических средствах.

Поэтому возникает задача не только предоставления разработчикам ППО некоторого принятого на данный момент набора программных интерфейсов (стандартов), но и обеспечения преемственности с требованиями ранее разработанного ППО.

Исходя из данных предпосылок и анализа существующего на рынке СВБУ АСУ ТП ППО были сформулированы требования к функциональному составу и программным интерфейсам, которые должны присутствовать в ОС, претендующей на включение в СПО. Были выделены следующие необходимые для нормального функционирования ППО функциональные характеристики, сгруппированные по подсистемам ОС, а также протоколы и программные интерфейсы (таблицы 1, 2).

Выбор ОС как основы для дальнейшей разработки Обзор существующих ОС

Выбор ОС для включения в состав СПО был произведен на основе анализа сформулированных выше требований. Дополнительными условиями стали возможности модификации, верификации, обслуживания и сопровождения, поддержка основных производителей технических средств и ПО.

В полной мере всем вышеизложенным требованиям не удовлетворяет ни одна из известных на момент принятия решения ОС (HP-UX, Solaris, Windows, Linux и т.д.).

Уже в ходе предварительного анализа были исключены практически все известные на момент принятия решения коммерческие ОС зарубежных производителей из-за наличия ограничений на поставку на АЭС. Данные ограничения выдвигались как со стороны самих производителей, так и регулирующих органов РФ. Дальнейшему детальному функциональному анализу подвергались так называемые "свободные" ОС (FreeBSD, Linux).

Их сравнение показало, что ОС Linux получила в настоящее время наибольшее распространение и в максимальной степени пользуется поддержкой основных

производителей технических средств и ПО. Поэтому "свободной" некоммерческой операционной системой, рассматриваемой далее в качестве возможной к применению в СПО СВБУ, выбрана ОС семейства Linux.

ОС Linux обладает рядом достоинств:

- на ОС Linux не наложены никакие лицензионные ограничения, кроме GNU-лицензии (GNU-General Public License);
- в состав ОС Linux входят все компоненты и службы, указанные в перечне требований к ОС (таблицы 1, 2);
- ОС Linux поставляется вместе с открытым исходным кодом и обладает модульностью — эти качества определяют возможность для модификации, верификации, обслуживания и сопровождения ОС, а также возможность доработки отдельных модулей под конкретное применение, дают пользователю возможность эффективно распоряжаться своими ресурсами, в частности, отключать ненужные модули [Computer World. Russian edition 2000, 34, с. 26].

Наряду с вышеуказанными достоинствами ОС Linux имеет также ряд недостатков (разрозненность документации, слабое развитие "настольных" приложений, фрагментарная поддержка русского языка). Часть разработчиков операционных систем утверждает, что ОС Linux стала популярной благодаря бесплатности и нетребовательности к аппаратному обеспечению, но вовсе не благодаря техническому совершенству архитектуры ядра. Ядро Linux неплохо подходит для однопроцессорных вычислительных систем, но гораздо хуже — для многопроцессорных конфигураций.

Таким образом, ОС Linux может быть использована в качестве ОС для СПО при условии проведения необходимой доработки, связанной с ее настройкой на конкретные задачи и технические средства проекта СВБУ, а также верификации и аттестации выборочных компонентов СПО с целью увеличения их защищенности и стабильности работы.

Было рассмотрено несколько дистрибутивов этого семейства и в качестве базового (исходного) был выбран RedHat 6.1 Linux, занимавший более 50% рынка Linux на 1999 г. (данные International Data Corp).

Анализ эксплуатационной истории и имеющейся в наличии документации на предмет определения достигнутого качества разработки

Целью введения данной фазы в процесс разработки СПО является необходимость оценки достигнутого уровня разработки модулей, включаемых в состав СПО. Особенностью использования свободно распространяемого ПО является сильная неоднородность качества различных его составляющих. На это влияет и уровень программистов, разработавших отдельный модуль, число пользователей вольно или невольно тестирующих этот модуль, диверсификация сценариев использования данного модуля и т.д. Нами были разработаны следующие методы для оценки достигнутого качества модулей ПО.

Анализ характеристик исходного кода модулей. В характеристики были включены, например, комментируемость кода (число символов комментариев на функцию и на файл), наличие рекуррентных функций. Для интенсификации анализа была разработана программа автоматического расчета данных характеристик и ранжирования результата. После обработки данных модули, имеющие низкую или неравномерную документируемость, большое число рекуррентных вызовов были подвергнуты более тщательному просмотру для принятия решения о качестве исходного кода, при необходимости принималось решение о замене или доработке модуля.

Анализ опыта эксплуатации модуля (или группы модулей). Данный анализ базировался на моделировании процессов проявления и исправления ошибок в ПО неоднородными случайными процессами Пуассона (общий подход к моделированию процессов неоднородными случайными процессами Пуассона дан по формуле (1)).

Пусть, версия ПО проходит испытания, в которых не выявлена ни одна ошибка, $N(t)$ число проявившихся ошибок ПО за время t опытной эксплуатации версии r ПО, есть неоднородный процесс Пуассона с интенсивностью $\lambda(t)$, $t \geq 0$. Обозначим, $N(t_r, t_{r-1}) = N(t_r) - N(t_{r-1})$. Тогда для любых моментов $0 < t_1 < t_2 < \dots$ справедливо

$$P\{N(t_r, t_{r-1}) = k\} = \frac{[m(t_r) - m(t_{r-1})]^k}{k!} \exp\{-[m(t_r) - m(t_{r-1})]\}, \quad (1)$$

где $k = 0, 1, 2, \dots$ K — число ошибок,

$$m(t_r) = \int_0^t \lambda(u) du, \quad (2)$$

$r = 0, 1, 2, \dots$ — ряд последовательно испытываемых версий ПО; $m(t_r)$ — среднее число проявлений ошибок за время t в версии r .

Вероятность того, что на интервале (t, t_s) ошибки не проявятся, определяется в соответствии с (1) и (2):

$$P\{N(t, t_s) = 0\} = \exp\left\{-\int_t^{t+s} \lambda(u) du\right\}. \quad (3)$$

Для оценки показателя (3) необходимо по заданной статистике доработки ПО и по результатам опытной эксплуатации ПО построить функцию, аппроксимирующую $m(t)$ или $\lambda(t)$ и на основании этого получить верхнюю оценку $m(t)$ или $\lambda(t)$ с заданным уровнем доверия. После подстановки этой оценки в (3) можно вычислить показатель безошибочности ПО.

Для выбора аппроксимирующей зависимости в настоящее время нет каких-либо аргументов, поэтому ниже был принят и использован критерий минимума среднеквадратического отклонения. В качестве аппроксимируемой кривой для $m(t)$ была использована экспонента. Причем, в качестве параметра t были взяты дискретные значения номеров, соответствующих версий ПО. Номер версии ПО принимался из натурального ряда чисел: 1, 2, 3, ...

Рассмотрим результаты статистики устранения ошибок в версиях графических модулей X11 (таблица 3).

По этим статистическим данным была построена зависимость

$$m(x) = k_0 \exp(k_1 x),$$

а также проверена ее статистическая адекватность. Найденные коэффициенты имеют следующие значения: $k_0 = 57,943$, $k_1 = -0,69228$.

Для последней девятой версии найдена верхняя оценка $m(9)$ с $\gamma = 90\%$, т. е. вероятность того, что $P\{m(9) \leq m(9)\} = 0,9$, и по формуле (3) вычислено значение показателя. Найденное значение показателя вероятности того, что в девятой версии в процессе эксплуатации не проявится ошибка равно $P\{N(9) = 0\} \geq \exp\{-0,3298\} \approx 0,72$.

Проведенный анализ позволил еще до начала фазы тестирования СПО выделить модули, чей качественный уровень возможно недостаточен для включения в СПО. Отметим, что на данной фазе было исследовано не только прогнозируемое функциональное качество ПО, но и пригодность данного модуля для последующей модификации и сопровождения, т. е. исследованы характеристики, стоящие за рамками обычного функционального и интеграционного тестирования.

Доработка и расширение функциональности ОС Linux для СПО LICs

ОС Linux RedHat 6.1 была подвергнута значительной доработке для того, чтобы включить ее в СПО LICs, для оценки приведем отношение объема оригинального кода включенного в состав начального дистрибутива RedHat 6.1 и кода, замененного при доработке — оно составляет около 40%. Полностью новой разработкой стал модуль самодиагностики (CMS), позволяющий контролировать целостность системных и пользовательских статических данных на файловой системе.

Доработка и расширение функциональности ОС Linux позволили включить ее в состав СПО LICs и полностью удовлетворить предъявленным на СПО функциональным требованиям.

Оценка качества СПО LICs

Оценка качества как составная часть тестирования и верификации программного обеспечения для АСУТП АЭС производится на протяжении всего жизненного цикла ПО [4] в соответствии с руководящим документом для процесса верификации ПО [5]. Создание сложного и качественного ПО невозможно без проведения широкомасштабного тестирования. С одной стороны, любая программа благодаря тому, что реализует некий известный алгоритм работы, может быть полностью протестирована при ограничении входного набора данных. Однако, это утверждение правомерно только для тривиальных либо некоторых специальных случаев ПО. Ресурсы, выделяемые на

Таблица 3. Статистика устранения ошибок в модулях X11 СПО LICs¹

Номер версии	Число устраненных ошибок
1	17
2	8
3	
4	4
5	5
6	1
7	
8	
9	0

¹В таблице приведены только те ошибки, которые оказывают влияние на корректность работы гипотетического ППО.

тестирование ПО, в большинстве случаев недостаточны для обеспечения полного тестового покрытия программы. Поэтому возникает задача классификации частей программного кода для того, чтобы выделить участки, наиболее вероятно содержащие ошибки.

В настоящее время на практике в зависимости от конкретных целей используется большой набор показателей для оценки достигнутого качества ПО [6, 7]. Нами были выделены именно те показатели качества ПО, которые позволили провести анализ кода, не вдаваясь в детали алгоритма функционирования ПО с использованием, где это возможно, средств автоматического анализа кода: объектно-ориентированные показатели,

показатели сложности и размера/объема (проекта в целом, класса, функции, файла). Анализ разработанного кода СПО LICs проводился и на соответствие рекомендациям по разработке программ, используемым в области систем безопасности АЭС [8]. В табл. 4 приведена сводная статистика использования зарезервированных слов, которые могут влиять на корректность выполнения программы, для программных модулей, входящих в состав СПО LICs. В табл. 5 приведены значения некоторых показателей сложности программных модулей (чем выше показатели сложности, тем исходный код тяжелее для изучения, контроля и модификации):

- **JC0** — минимальное число контрольных точек, необходимых для записи всех блоков данных тестового покрытия;
- **JC1** — минимальное число контрольных точек, необходимых для записи основных сегментов данных тестового покрытия;
- **JC1+** — минимальное число контрольных точек, необходимых для записи всех сегментов данных тестового покрытия;
- **JC2** — минимальное число контрольных точек, необходимых для записи всего сегмента условий тестирования данных;
- **CC0** — цикломатическая сложность с учетом case;
- **CC2** — цикломатическая сложность без учета case.

Существует тесная взаимосвязь между сложностью разрабатываемого ПО, затратами на его разработку и тестирование для достижения желаемого уровня качества. Высокая сложность сама по себе является довольно распространенным источником ошибок в программном обеспечении. Сложность, которая, вообще говоря, является достаточно абстрактным понятием, в случае программного обеспечения может быть измерена и соответственно может напрямую использоваться для распределения затрат на тестирование. При прочих равных условиях действует правило: чем сложнее программный модуль, тем с большей вероятностью он содержит ошибки. Принимаемая во внимание тесная взаимосвязь между сложно-

стью и ошибками, были сосредоточены усилия на тестировании тех частей или модулей ПО, которые наиболее подвержены ошибкам.

Функциональное тестирование и интеграция ПТС

Процесс тестирования предназначен для выявления ошибок в программе и может осуществляться различными методами. Для СПО LICS были выбраны два метода тестирования: статистическое и систематическое. При систематическом методе тестирования проводятся испытания каждого модуля в соответствии с критериями, указывающими, что нужно испытать в модуле и ожидаемую реакцию модуля на тест. Результаты испытаний сверяются с контрольными и выносятся оценка о наличии ошибки в модуле.

В основном подверглись систематическому тестированию модули, оценка уровня качества которых была признана низкой. ОС, как правило, трудно анализировать из-за диалоговой природы и большого числа входных параметров. Ограниченность ресурсов тестирования приводит к необходимости ограничения тестового покрытия.

Статистические методы использовались в дополнение к систематическим методам для выявления ошибок в ОС. Методика тестирования строилась таким образом, чтобы удовлетворялись следующие условия:

- последовательность и число тестов не влияли на результат одного теста;
- прогон теста контролировался так, чтобы каждый испорченный прогон или сбой детектировался как таковой;
- число прогонов теста задавалось достаточно большим.

Конечным этапом тестирования был процесс интеграции ТС и СПО в единый ПТС, способный выполнять специальные функции.

Заключение

В результате работ по созданию СПО СВБУ АСУТП АЭС в ИПУ РАН на базе свободно распространяемой ОС Linux был создан коммерческий продукт СПО LICS. В разработке достигнут высокий уровень качества программного обеспечения и интегрируемости с техническими средствами, представлен пользователю и разработчику широкий спектр современных средств и библиотек программирования. СПО LICS прошло испытания и было рекомендовано для использования в современных СВБУ АСУТП АЭС как элемент системы важной безопасности АЭС по классу 3Н. Специалисты ИПУ РАН накопили большой опыт работы по созда-

Таблица 4. Сводная статистика использования зарезервированных слов в программных модулях

Зарезервированное слово	eepr00.c	pci-scan.c	pci-scan.h	savage_accel.c	savage_bci.h	savage_cursor.c	savage_dgac	savage_driver.c	savage_driver.h	savage_i2cc.c	savage_image.c	savage_regs.h	savage_shadow.c	savage_vbe.c	savage_vbe.h	savage_video.c
switch	3	2	0	4	0	0	8	0	0	0	0	0	0	0	0	7
default	2	1	0	2	0	0	3	0	0	0	0	0	0	0	0	4
Goto	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
return	33	20	0	12	0	5	6	95	0	1	0	0	5	0	0	36

Таблица 5. Показатели сложности программных модулей

Файл	Число строк	JC0	JC1	JC1+	JC2	CC0	CC2
cgilib.h	30	0					
status.h	36	0					
upsc.c	123	21	31	32	42	14	14
upsclient.c	811	110	170	172	242	85	84
upsclient.h	125	0	0	0	0	0	0
upssched.c	604	108	163	168	232	83	82
upsset.c	20	0					
atexit.c	22	1					
inet_aton.c	34	5	7	9	3	0	0
snprintf.c	914	134	194	210	290	116	91
strerror.c	503	3	4	1	3	0	0
strncpy.c	68	5	11	12	18	7	0
upsconf.c	100	15	24	38	13	0	0

нию высоконадежных программных комплексов. Информация по СПО LICS доступна по адресу <http://www31.ipu.rssi.ru/~lics>

Список литературы

1. IEC 60880 Ed. 1.0 b:1986, Software for computers in the safety systems of nuclear power stations.
2. IEC 60880-2 Ed. 1.0 b:2000, Software for computers important to safety for nuclear power plants — Part 2: Software aspects of defence against common cause failures, use of software tools and of pre-developed software.
3. ПНАЭ Г-01-011-97. Общие положения обеспечения безопасности атомных станций. ОПБ- 88/97. Госатомнадзор России. 1998.
4. Лунаев В.В. Обеспечение качества программных средств. Методы и стандарты. М.: СИНТЕГ, 2001.
5. ANSI/ANS-10.4-1987; R1998: Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry.
6. ГОСТ 28195-89. Оценка качества программных средств. Общие положения.
7. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению.
8. Review Guidelines on Software Languages for Use in Nuclear Power Plant Safety Systems (NUREG/CR-6463).

Станислав Ильич Масолкин — научный сотрудник, Виталий Георгиевич Промыслов — канд. физ.-мат. наук, ст. науч. сотрудник, Елена Филипповна Жарко — канд. техн. наук, ст. научный сотрудник, Александр Владимирович Антонов — канд. техн. наук, ст. научный сотрудник, Ольга Александровна Промылова — инженер, Армен Суренович Степанянц — ст. научный сотрудник Института проблем управления им. В.А. Трапезникова РАН.
Контактный телефон (095) 334-75-71. E-mail: poletik@ipu.rssi.ru