

ОБЪЕДИНЕННЫЙ  
ИНСТИТУТ  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ  
ДУБНА

P10-86-334

А.М.Ершов

СРЕДСТВА, РАСШИРЯЮЩИЕ ВОЗМОЖНОСТИ  
ОТЛАДКИ ПРОГРАММ  
НА ЯЗЫКЕ КОБОЛ ОС ЕС

Направлено в журнал "Управляющие  
системы и машины"

1986

## Введение

Существенным фактором, во многом определяющим качество создаваемого программного обеспечения различного рода автоматизированных систем, является выбор языка программирования. С точки зрения прикладного программиста среди основных критериев выбора языка можно выделить удобство его для обучения, чтения и написания программ, а также их отладки. При решении задач класса обработки данных одним из языков, наиболее соответствующих перечисленным требованиям, является алгоритмический язык Кобол<sup>/1,2/</sup>. Он широко применяется в практике вычислений, имеет русскую и английскую версии. Другим языком, также часто используемым при разработке автоматизированных систем управления и обработки данных, является язык ПЛ-1. Этот язык, носящий универсальный характер и состоящий в себе черты Фортрана и Кобола, часто критикуется за перегруженность предоставляемыми им возможностями, сложность и невысокую надежность (провоцирование ошибок – как явных, так и скрытых)<sup>/3,4/</sup>. Необъятность языка ПЛ-1 явилась причиной распространения практики выделения из него подмножеств, ориентированных на решение различных классов задач<sup>/5,6,7/</sup>.

Одним из определяющих факторов выбора языка программирования является область и характер решаемых с помощью его задач. В<sup>/8/</sup> проводится достаточно подробный сравнительный анализ функциональных возможностей языков Кобол и ПЛ-1 применительно к задачам обработки данных. В качестве преимуществ языка Кобол следует выделить богатый набор операций обработки строк, проблемно-ориентированные средства обработки таблиц, наличие и явное определение средств сортировки. Большим достоинством Кобола является реализация функций формирования отчетных документов полностью на проблемно-ориентированном уровне. Генерация отчетов производится на основе задаваемых в специальной секции программы форм и правил выдачи, что при необходимости дает возможность с незначительными затратами осуществить модификацию отчета. При этом генерация сложных отчетных форм выполняется автоматически при использовании фактически единственного оператора

( GENERATE ), в то время как в случае программирования на языке III-I для получения подобных отчетов приходится разрабатывать довольно объемные программы.

Следует подчеркнуть, что операции сортировки записей файлов и формирования отчетных документов являются одними из наиболее интенсивно используемых операций в задачах обработки данных, в частности в типовых задачах АСУ.

Язык Кобол часто критикуется за многословность, вместе с тем эта его черта при соответствующем использовании мнемонических имен переменных, параграфов и т.п. дает возможность разработки легко читаемых и понимаемых "самодокументированных" программ. Кроме того, опыт использования Кобола показывает высокую эффективность и гибкость имеющихся в нем средств создания функциональных блоков (например, с помощью оператора PERFORM ). Эти свойства обеспечивают программам на Коболе по сравнению с программами на других языках более высокую степень таких качеств, как читабельность, ясность<sup>/9/</sup>.

Анализ отмеченных черт языка Кобол свидетельствует о целесообразности его применения для решения большинства задач обработки данных, типовых задач АСУ<sup>/8/</sup> и подтверждает вывод о недостаточной обоснованности существующей тенденции преимущественного использования языка III-I для этих целей<sup>/10/</sup>.

Транслятор с языка Кобол в операционной системе ОС ЕС в целом обладает высокими эксплуатационными качествами, им строится достаточно компактный и эффективный объектный код. Существенным недостатком реализации указанного транслятора является неполный объем информации об ошибке, вызвавшей аварийное завершение или прерывание выполнения программы. В этом случае пользователю сообщаются системный код завершения, дающий весьма общее представление об ошибке и ее причинах, а также некоторые дополнительные сведения. Если при этом учесть тот отмечавшийся выше факт, что программа на Коболе при сохранении ясности имеет в среднем больший размер по сравнению с программами на других языках, то становятся понятными те трудности, с которыми сталкивается программист на стадии отладки. Действительно, например, системный код 0С7, свидетельствующий о программном прерывании, является вследствие неправильных данных в операндах операций десятичной арифметики, редактирования или команд преобразования данных. А это, в свою очередь, может быть вызвано ошибками в данных не только в вычислительных операторах языка Кобол, но и при различных сравнениях в операторах IF , при проверке условий в операторах PERFORM и т.п. Системные коды прерываний 0С1, 0С4 - 0С6, сигнализирующие соответственно о некорректности кода операции, нарушении защиты памяти, ошибках адресации и спецификации, могут появиться при невыполнении тре-

бований открытия и закрытия файлов, при неправильном использовании средств сортировки и генератора отчетов языка Кобол, при неверных передачах и Возвратах управления и т.д.

Перечисленные обстоятельства обуславливают серьезные затруднения при локализации возникающих ошибок. Для определения места прерывания программист немало времени затрачивает на анализ значительных по объему частей программы, он вынужден прибегать к трассировке программы, к включению в нее операторов отладочной печати и т.п.

В работе рассматривается комплекс программных средств, позволяющих свести до минимума затраты на поиск оператора, вызвавшего аварийное завершение или прерывание выполнения программы, на выявление причин возникновения ошибки. При этом программист избавлен от необходимости постоянно включать в программу специальные отладочные операторы, что является недостатком стандартных средств отладки языка Кобол ОС ЕС.

Формирование таблицы смещений операторов. Если пользователем не заказано получение дампа оперативной памяти программы<sup>/11,12/</sup>, т.е. отсутствуют операторы DD с именем SYSUDUMP или SYSABEND , то кроме кода завершения в зависимости от версии и режима используемой ОС на печать может быть выдан индикативный дамп. В нем содержатся причина и адрес программного прерывания, информация о состоянии обших регистров, о блоках активных запросов (RV ) и слове состояния программы (PSW ), наименования и адреса точек входа загрузочных модулей.

Стандартный листинг трансляции программы с языка Кобол ОС ЕС не содержит таблицы соответствия между номерами операторов исходного текста и их смещениями относительно начала программной секции. Для получения этой информации могут быть использованы предоставляемые транслятором возможности формирования полного и сокращенного листингов объектной программы. Для этого указываются соответственно параметры (опции) транслятора PMAP и CLIST . Полный листинг представляет собой текст программы на языке Ассемблер. Из него может быть получена практически вся необходимая информация. Сокращенный листинг содержит перечень операторов программы с некоторыми их характеристиками. Недостатком его непосредственного использования является содержащаяся в нем излишняя с точки зрения отладки информация, некомпактное ее размещение и, как следствие, большой объем выдачи на печать. Естественно, этим недостатком страдает использование и полного листинга объектной программы.

Предоставление информации о смещениях операторов относительно начала программной секции во многих случаях обеспечило бы точное определение оператора программы, при выполнении которого произошла ошибка. В частности, значительно упростилась бы локализация большин-

ства программных прерываний, например с кодом 0С7, часто возникающих при отладке программ обработки данных. С целью реализации указанной возможности разработана программа STATLOC, обеспечивающая дополнение стандартного листинга транслятора с Кобола ОС ЕС достаточно компактной таблицей смещений операторов относительно начала программной секции. При формировании таблицы используется сокращенный листинг объектной программы, т.е. указывается опция транслятора SLIST. Выводить на печать такой листинг (имя оператора DD - SYSRINT) нет необходимости - он помещается во временный набор данных, который затем передается программе STATLOC. Программа STATLOC производит анализ и разбор получаемого листинга, выделяет в нем информацию объектной программы и выбирает из нее сведения, касающиеся только операторов исходного текста и их адресов внутри программной секции. Далее сначала выводится на печать стандартный листинг трансляции, а за ним - сформированная таблица смещений операторов относительно начала программной секции. Таблица строится в II двойных колонок, каждая из которых имеет заголовки ST - для номеров операторов в исходном тексте программы и LOC - для соответствующих смещений операторов.

При аварийном завершении программы пользователь по имени выполнявшегося загрузочного модуля (указывается в поле PROGRAM ID=) находит соответствующий адрес точки входа в программу (в поле ENTRY POINT =). При программном прерывании адрес выполнявшейся в этот момент команды указывается явно в конце сообщения: PROGRAM INTERRUPTION (DATA) AT LOCATION

В скобках - причина программного прерывания (DATA - неправильные данные). В случае некоторых других видов прерываний адрес выполнявшейся команды может быть найден исходя из содержимого общих регистров, слова состояния программы /11,12,13/.

Полученные таким образом адреса являются реальными адресами памяти ЭВМ, они представлены в шестнадцатиричной форме. Разница между адресом выполнявшейся команды и адресом точки входа представляет собой относительный адрес выполнявшейся в момент прерывания команды. Пользуясь сформированной таблицей смещений, по нему легко определить номер оператора программы, вызвавшего ошибку.

Следует обратить внимание при расчете относительного адреса команды, если начало программной секции не является точкой входа программы пользователя. Это может быть следствием использования, например, управляющего предложения ENTRY на шаге редактирования связей. Сказанное относится к случаю, когда значение ENTRY ADDRESS в распечатке листинга редактора связей отлично от 0. При этом необ-

ходимо только сделать соответствующий пересчет на величину относительного смещения оператора программы, определяющего ее точку входа.

Для использования возможностей, предоставляемых программой STATLOC, была выполнена незначительная модификация стандартных каталогизированных процедур ОС ЕС, реализующих вызов транслятора с языка Кобол (SOVUS, SOVUSL, SOVUSLG и др.). Модификация выразилась в добавлении в процедуры специального шага, обеспечивающего работу указанной программы и выполняющегося после шага трансляции. При обращении к транслятору среди прочих задана опция SLIST. Если ее отменить (опустить или указать NOSLIST), то будет выдан только стандартный листинг без таблицы смещений операторов.

Средство трассировки отлаживаемой программы. Описанное выше средство поиска ошибочного оператора при отладке программы на языке Кобол, как было указано, удобно применять при появлении, в основном, программных прерываний. В случае многих других видов аварийных завершений, а также и части программных прерываний, порой возникают трудности при определении адреса оператора программы, ставшего источником ошибки. Это относится к случаю, когда прерывание произошло вне пределов программы, например при выполнении одной из системных функций. Для локализации программных ошибок и связанных с ними прерываний любого типа разработано универсальное средство трассировки отлаживаемой программы, обеспечивающее полную автоматизацию выявления и распечатки ошибочного оператора.

Средство трассировки включает в себя программы SOVOLDVBG, STAEDVBG, SOVOLPRT и PRINTDVBG. Программой SOVOLDVBG выполняется синтаксический анализ исходного текста отлаживаемой программы. В процессе анализа выделяются все операторы, являющиеся глаголами языка Кобол, и производится модификация текста программы. Она заключается в том, что непосредственно перед каждым найденным оператором-глаголом в соответствии с синтаксическими правилами языка добавляется оператор MOVE, обеспечивающий запись порядкового номера текущей строки программы в ее оперативной памяти. Для перехвата управления в момент аварийного завершения отлаживаемой программы использовано системное средство сопряжения ABEND-STAE. Для его активизации при модификации текста программы обеспечивается обращение из нее к специальному модулю STAEDVBG. С этой целью в начале программы, а также в других возможных точках входа (после операторов ENTRY) включаются необходимые операторы CALL. При первом выполнении модуля STAEDVBG строится активный блок управления STAE (SCB), в котором указывается адрес программы выхода, в данном случае - PRINTDVBG.

Модифицированный текст отлаживаемой программы передается на трансляцию, редактирование связей и выполнение. Если при выполнении происходит аварийное завершение, то управление получает программа PRINTDBG. На основе передаваемого ей в качестве параметра номера строки, хранящегося в оперативной памяти, она считывает и выводит на печать оператор отлаживаемой программы, выполнявшийся в момент возникновения ошибки. Схема работы комплекса программ трассировки приведена на рис. I.

Для удобного применения средства трассировки создана каталогизированная процедура SOBOLDBG. Ее отличие от стандартной каталогизированной процедуры SOBOLSLG заключается в использовании программ SOBOLDBG и SOBOLPRT. Программой SOBOLPRT удаляется из листинга трансляции модифицированной программы вся информация, дополнительно включенная в исходный текст программы для обеспечения ее трассировки. В результате для пользователя становится практически незаметной работа данного средства отладки, и оно может быть с успехом использовано, начиная с самых ранних этапов разработки программы. Важно отметить, что при этом не накладывается никаких ограничений на применение любых синтаксических конструкций языка Кобол. Единственное отличие при вызове каталогизированной процедуры SOBOLDBG связано с необходимостью передачи исходного текста программы на вход не транслятора, а программы SOBOLDBG. Оно выражается в том, что при обращении опускается индекс шага для оператора DD с именем SYSIN:

```
//SOBOL EXEC SOBOLDBG
//SYSIN DD DSN=LIBRARY(PGM), DISP=SHR
//COB.SYSLIB DD DSN=LIBRARY, DISP=SHR
```

В результате оператор SYSIN автоматически добавляется к первому шагу каталогизированной процедуры, на котором и выполняется программа SOBOLDBG (сов - имя шага трансляции программы).

Дополнительные расходы, связанные с работой средств трассировки отлаживаемой программы, незначительны и составляют около 5-10% от затрат времени центрального процессора при использовании обычной каталогизированной процедуры SOBOLSLG. Операции записи текущих номеров операторов в основную память практически не вызывают замедления работы отлаживаемой программы.

Автоматизация анализа дампа оперативной памяти. Один из способов отладки программ заключается в использовании для анализа и выявления причин ошибок дампов оперативной памяти ОС/11,12,13/. Дамп представляет собой распечатку всей или наиболее значимой части основной памяти в момент завершения задачи. Безусловно, он является мощным инструментом отладки, вместе с тем его непосредственное ис-

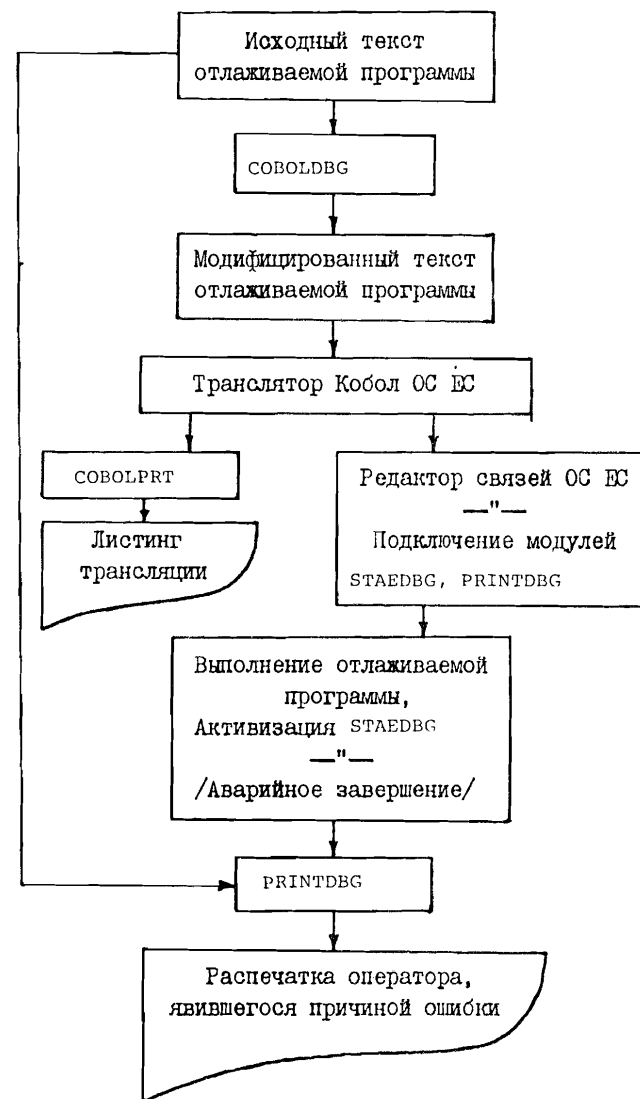


Рис. I. Схема работы средств трассировки отлаживаемой программы.

пользование при программировании на языке высокого уровня имеет ряд существенных недостатков. В частности, кроме обычного листинга трансляции программы требуется выводить на печать сгенерированный листинг на языке Ассемблера. Чтобы проанализировать содержимое программных переменных, необходимо также обеспечить расширенную распечатку их внутренних имен, смещений и т.п. Если при этом учесть тот факт, что, как правило, в каждой конкретной ситуации из получаемого дампа оперативной памяти для анализа требуется очень незначительная его часть, то вряд ли можно признать целесообразным и оправданным постоянное использование такой процедуры поиска ошибок<sup>/11/</sup>. Тем не менее объем и разнообразие содержащейся в дампе информации о задаче и причинах ее аварийного завершения очень велики. Из этого факта, а также сказанного выше напрашивается вывод о необходимости разработки средств более эффективной работы с дампом оперативной памяти. Примером реализации подобного подхода может служить включение подсистемы диалогового просмотра дампов в состав системы виртуальных машин ЕС ЭВМ<sup>/14/</sup>.

С целью автоматизации анализа дампа оперативной памяти, получаемого для программ на языке Кобол, была разработана специальная программа совдмп. Для ее применения создана каталогизированная процедура совдмп, по своим выполняемым функциям аналогичная стандартной процедуре совусг. Отличия в использовании этих процедур выражаются в дополнительных возможностях, предоставляемых программой совдмп. Если отлаживаемая программа завершается аварийно, то помимо информации о месте возникновения ошибки на печать также выводится содержимое всех переменных и полей, что может оказать большую помощь при анализе причин ошибки. Схема работы этого средства отладки приведена на рис.2.

В каталогизированной процедуре совдмп на шаге трансляции отлаживаемой программы задаются опции транслятора дмар и рмар, обеспечивающие выдачу внутреннего описания переменных и генерацию листинга на языке Ассемблер. Получаемый таким образом расширенный листинг трансляции на печать не выводится, а через оператор dd с именем sysprint помещается во временный рабочий набор данных. Если происходит аварийное завершение отлаживаемой программы, то сформированный дамп оперативной памяти через оператор dd с именем sysudmp помещается в другой рабочий набор данных. После этого управление передается программе совдмп, выполняющейся на последнем шаге процедуры и производящей параллельный однопроходный анализ обоих временных рабочих наборов данных.

В дампе оперативной памяти программой совдмп отыскивается адрес прерывания выполнения отлаживаемой программы (INTERRUPT AT),

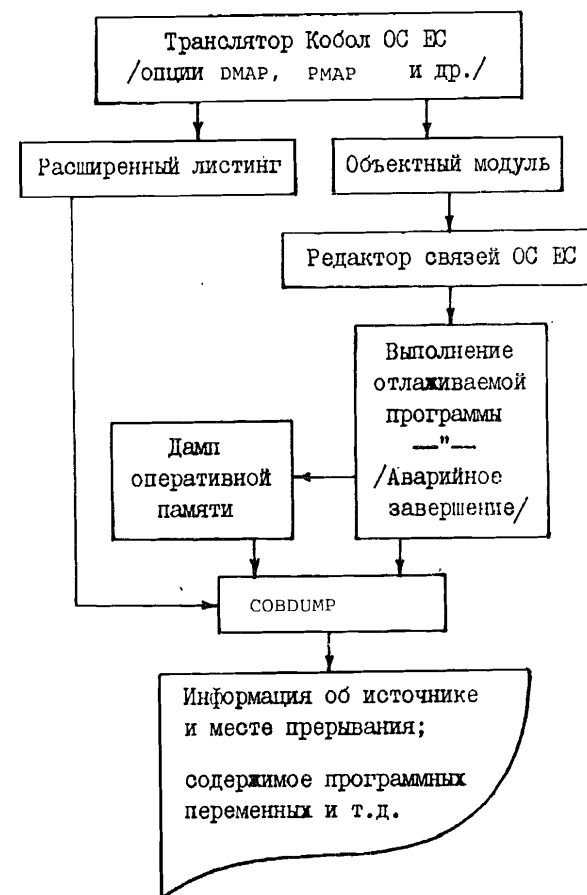


Рис. 2. Схема работы средства автоматизации анализа дампа оперативной памяти программы.

а также адрес ее точки входа EPA (ENTRY POINT ADDRESS ) в таблице оглавлений модулей CDE (CONTENTS DIRECTORY ENTRY ). На основе сопоставления этих адресов и информации из сгенерированного листинга на языке Ассемблер на печать выводится сообщение о месте возникновения ошибки. Если прерывание произошло внутри отлаживаемой программы, то указывается и оператор, явившийся причиной этой ошибки.

Важной особенностью программы совдупм является предоставление пользователю информации о содержимом всех переменных и полей в момент аварийного завершения отлаживаемой программы. Для этого из таблицы внутреннего описания переменных выбираются сведения об их длине, внутреннем представлении, идентификаторе базового регистра и смещении. С помощью таблицы назначения базовых регистров (REGISTER ASSIGNMENT ) программой определяются конкретные номера общих регистров, поставленные транслятором в соответствие каждому идентификатору базового регистра. На основе состояния общих регистров в дампе ( REGS AT ENTRY TO AVEEND ) находятся значения, содержащиеся в нужных базовых регистрах. По ним и соответствующим им смещениям вычисляются реальные адреса переменных и полей. По этим адресам программой совдупм содержимое переменных отыскивается в дампе и выводится на печать. Чтобы обеспечить распечатку переменных с любым внутренним представлением, вывод производится в шестнадцатиричной форме. Для удобства параллельно выдается таблица расшифровки в коде ДКОИ.

Программой совдупм обеспечивается печать листинга трансляции отлаживаемой программы в обычной стандартной форме, т.е. не выводятся сгенерированный листинг на языке Ассемблер и внутренние описания переменных. Не печатается также и дампы оперативной памяти. Структура программы совдупм дает возможность расширения списка выполняемых ею функций в необходимом направлении. Так, при прослеживании передач управления от программы к программе может оказаться полезным содержащийся в дампе список областей сохранения ( SAVE AREA TRACE ) и т.п.

Заключение. Рассмотренный в работе комплекс программных средств позволил практически полностью устранить один из самых серьезных недостатков транслятора с языка Кобол в ОС ЕС. Как показал опыт использования этих средств, предоставление разнообразной информации о месте и источнике возникновения ошибки в программе обеспечило значительное повышение эксплуатационных свойств транслятора.

Представляется весьма целесообразным дополнить листинг трансляции с языка Кобол ОС ЕС таблицей смещений операторов относительно начала программной секции. Затраты на выполнение программы статлос невелики - около 5% от времени трансляции программы, а практическое

ее использование показало, что это средство дает возможность легко определить причину более половины всех случаев аварийного завершения программ. Причем получаемая таблица смещений не является однократно используемым инструментом. Она может быть применена и при последующей эксплуатации программы, например при возникновении ошибки на других входных данных и т.п.

Отсутствие каких-либо дополнительных затрат со стороны пользователя и высокая эффективность средства трассировки обусловило его активное применение при отладке программ как начинающими, так и опытными пользователями. Для расширения сферы его использования был создан ряд каталогизированных процедур, обеспечивающих отладку программ, работающих под управлением других мониторинговых программ, в частности в среде системы управления базами данных ОКА.

В любой ситуации при анализе причин аварийного завершения программы большую помощь оказывает распечатка содержимого переменных и полей, выполняемая программой совдупм . Следует отметить, что наличие подобной функции является положительной и весьма желательной чертой современных трансляторов.

В целом применение описанных программных средств дало возможность облегчить и упростить разработку программ на языке Кобол ОС ЕС, особенно на стадии отладки, и позволило значительно сократить трудоемкость и сроки создания программного обеспечения.

#### Литература

1. Кобол ЕС ЭВМ / Быкова В.П., Загузова Л.К., Романовская Л.М. и др. - М., "Статистика", 1978, с.280.
2. Маджинис Дж. Программирование на стандартном КОБОЛе. - М., "Мир", 1979, с.487.
3. Майерс Г. Надежность программного обеспечения. - М., "Мир", 1980, 360 с.
4. Баррон Д. Введение в языки программирования. - М., "Мир", 1980, 190 с.
5. Балодис Р.П. Средства языка ПЛ-I, используемые в задачах обработки данных. - УСИМ, 1981, № 1, с.77-81.
6. Бублик В.В., Гороховский С.С. О применении языка ПЛ-I для реализации алгоритмов над структурированной памятью, - Кибернетика, 1983, № 1, с. 26-29.
7. Катков В.Л., Пилоцкий И.И. Подмножество языка ПЛ-I для системного программирования. - УСИМ, 1983, № 3, с.69-63.

8. Бабенко Л.П. Проблемно-ориентированные средства в языке Кобол. - М., "Статистика", 1979, 192 с.
9. Бозм Б., Браун Дж., Каспар Х. и др. Характеристики качества программного обеспечения. - М., "Мир", 1981, 208 с.
10. Кавалерчик Б.Я. О повышении эксплуатационной надежности программного обеспечения. - УСИИ, 1982, № 5, с.71-76.
11. Риндфлайш Д. Отладка программ в системах 360/370 на основе дампов памяти операционной системы. - М., "Машиностроение", 1982, 144 с.
12. Ерофеев В.И., Меркулов Ю.П., Першиков В.И. и др. Средства отладки программ в ОС ЕС ЭВМ. - М., "Статистика", 1979, 245 с.
13. Данилочкин В.П., Одинцов Б.В., Леледов Г.В. Справочник системного программиста по операционной системе ОС ЕС. - М., "Финансы и статистика", 1982, 288 с.
14. Система виртуальных машин для ЕС ЭВМ. Справочник / Булко И.М., Дорожко Н.Н., Дудкин Л.И. и др. - М., "Финансы и статистика", 1985, 360 с.

Рукопись поступила в издательский отдел  
27 мая 1986 года.

Ершов А.М.

P10-86-334

Средства, расширяющие возможности отладки программ на языке Кобол ОС ЕС

Рассмотрена проблема упрощения и облегчения процесса отладки программ на языке Кобол ОС ЕС. Описан комплекс программных средств, позволяющий свести до минимума затраты на поиск оператора, вызвавшего аварийное завершение или прерывание выполнения программы, на выявление причин возникновения ошибки. Рассмотрены способ формирования таблицы смещения операторов относительно начала программной секции, средство трассировки отлаживаемой программы, программа автоматизации анализа дампа оперативной памяти. Отмечено существенное повышение эксплуатационных свойств транслятора с языка Кобол ОС ЕС при использовании указанных средств расширения возможностей отладки программ.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1986

Перевод автора

Ershov A.M.

P10-86-334

Facilities for the Extending OS ES COBOL Language Programs Debugging Means

The problem of simplifying and making easy OS ES COBOL language programs debugging process is considered. The program means complex intended to minimize expenditures for a search for the statement which has caused an abnormal termination or interruption of program, to reveal reasons of the error appearance are described. The method for construction of statement displacement table relatively to program section beginning, the debugging program trace facility, the program for the main storage dump analysis automation are considered. The essential improvement of OS ES COBOL compiler usage characteristics as a result of using specified facilities for the extending the program debugging means is noted.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR

Preprint of the Joint Institute for Nuclear Research. Dubna 1986