

КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

*Li Wei, V. Zinchenko, V. Borisov,
N. Brovarska, V. Chymakov*

FORMING OF SOFTWARE ENVIRONMENTS FOR SUPPORT OF TASKS OF AUTOMATION OF SCIENTIFIC RESEARCHES

In the robot is considered the information-control system "CINDY", allowing the user to create a comfortable environment to give the application program to support solving automation research.

Key words: object, data, the team, the operator, system.

Розглядається інформаційно-керуюча система "CINDY", що дозволяє створювати користувачеві зручне середовище прикладних програм для підтримки рішення задач автоматизації наукових досліджень.

Ключові слова: об'єкт, дані, команда, оператор, система.

Рассматривается информационно-управляющая система "CINDY", позволяющая создавать пользователю удобную среду прикладных программ для поддержки решения задач автоматизации научных исследований.

Ключевые слова: объект, данные, команда, оператор, система.

© Ли Вэй, В.П. Зинченко,
В.В. Борисов, Н.И. Броварская,
В.Г. Чумаков, 2014

УДК 681.3.06

ЛИ ВЭЙ, В.П. ЗИНЧЕНКО, В.В. БОРИСОВ,
Н.И. БРОВАРСКАЯ, В.Г. ЧУМАКОВ

ФОРМИРОВАНИЕ ПРОГРАММНЫХ СРЕД ДЛЯ ПОДДЕРЖКИ ЗАДАЧ АВТОМАТИЗАЦИИ НАУЧНЫХ ИССЛЕДОВАНИЙ

Введение. В работе рассматривается информационно-управляющая система (ИУС) "CINDY", позволяющая создавать пользователю удобную среду прикладных программ (ПП) с развитым интерфейсом и возможностью запуска в любых приложениях Windows для поддержки решения задач автоматизации научных исследований [1, 2]. В системе имеется полный набор средств для объединения в прикладные пакеты любых программных модулей, средства автоматизации обмена данными внутри пакетов и средства для обеспечения максимального удобства пользователя.

Важной особенностью системы является возможность хранения информации в нескольких независимых базах данных (БД), называемых РАЗДЕЛАМИ [3, 4]. Разделы могут располагаться произвольно, обмен данными между ними производится путем простого копирования, подобно тому, как копируются файлы в операционной системе. Все это позволяет осуществлять обмен объектами между подсистемами, не прибегая к экспорту данных в файлы. Таким образом, на базе ИУС "CINDY" можно создавать эффективные и легко модернизируемые пакеты ПП, полностью избавляющие пользователей от необходимости указывать имена файлов, содержащих обрабатываемую информацию, а также заниматься поиском и размещением этих файлов на дисках.

Структура данных и принципы работы. ИУС "CINDY" поддерживает иерархическую структуру данных, состоящую из отдельных

ОБЪЕКТОВ. Каждый ОБЪЕКТ представляет собой логически цельный блок данных, хранящийся в отдельном файле. Генерация файлов под новые объекты производится автоматически, по мере необходимости. Наряду с наименованием каждому объекту присваивается ФАЙЛОВЫЙ КОД – целое пятизначное число, используемое системой при поиске файла, содержащего объект. По функциональному назначению объекты делятся на три типа: СПИСКИ, ПРОЦЕДУРЫ, СВОБОДНЫЕ объекты. Система может поддерживать до пяти независимых РАЗДЕЛОВ.

СПИСКИ предназначены для хранения наименований объектов и комментариев к ним. Кроме того, в них содержатся файловые коды объектов, что позволяет ссылаться на объекты напрямую, минуя индексные файлы. При этом списки могут содержать ссылки на нижестоящие списки, позволяя формировать многоуровневую иерархическую БД. Максимально возможное количество уровней – 11, т. е. списки десятого уровня могут ссылаться только на таблицы данных, введенных с клавиатуры, процедуры и свободные объекты. В одном списке может быть зарегистрировано до 400 объектов.

ПРОЦЕДУРЫ обеспечивают групповое выполнение команд ИУС "CINDY", позволяя формировать на основе EXE-модулей пакеты ПП, работающие в среде "CINDY", а также наращивать функциональные возможности самой системы.

СВОБОДНЫЕ объекты предназначены для хранения данных произвольного формата, формируемых вне ИУС "CINDY", и не подлежат просмотру и редактированию ее внутренними средствами. Максимальный размер файлов, содержащих свободные объекты, в принципе не ограничен и определяется размером самих объектов. По своей внутренней структуре объекты делятся на две группы: группу ФОРМАТНЫХ объектов, включающую Списки и Таблицы, которые можно просматривать и редактировать средствами ИУС "CINDY", и группу НЕФОРМАТНЫХ объектов, которые формируются с помощью внешних программ (например, текстовых или графических редакторов).

Для активизации внешних программ, организации обмена данными, автоматизации поиска и выборки данных, активизации процедур и "прямой" загрузки форматных объектов в ИУС "CINDY" предусмотрен набор команд, использующих два режима ввода команд, которые определяют соответствующие режимы работы системы – диалоговый или автоматический. В диалоговом режиме команды вводятся вручную, с отображением в командной строке, и активизируются нажатием клавиши [Ввести]. В автоматическом режиме производится последовательное выполнение команд и операторов, записанных в Процедурах.

Ввод и корректировка команд. Функции ИУС "CINDY" реализуется путем ввода специальных команд, которые вводятся с помощью клавиатуры, отображаются в командной строке и обрабатываются сразу после нажатия клавиши [Ввести], если в командной строке присутствуют какие-либо символы. Перед выполнением команды производится анализ содержимого командной строки. Если введенная команда содержит ошибки, система ее игнорирует, а на дисплей выводится сообщение о том, какая именно ошибка была допущена пользователем.

лем. ИУС "CINDY" хранит в своей памяти до 50 последних введенных команд. Выбор команды производится с помощью маркера, который следует переместить в соответствующую строку окна, или "мышь". Выбранная команда помещается в командную строку, после чего достаточно нажать клавишу [Ввести].

Выбор раздела. Система может одновременно поддерживать до пяти независимых РАЗДЕЛОВ, выбор которых начинается с вызова на дисплей списка разделов, описанных в файле конфигурации config.c1, для чего следует нажать комбинацию клавиш [Ctrl+F10] либо воспользоваться системным меню. После этого следует указать необходимый раздел и нажать клавишу [Выбрать] или дважды щелкнуть левой клавишей "мышь". В результате в окне "CINDY" появится корневой список выбранного раздела. Система позволяет непосредственно копировать объекты из одного раздела в другой. Локализуя один из разделов на переносном диске, можно переносить данные с одного ПК на другой в виде объектов, не прибегая к промежуточному копированию данных в файлы, даже при отсутствии связи через сеть.

Команды ИУС "CINDY". Команды ИУС "CINDY" предназначены для активизации внешних программ, организации обмена данными, автоматизации поиска и выборки данных, активизации процедур "CINDY" и "прямой" загрузки объектов.

<Command>: <arg> [<1-й ПР>]...[<4-й ПР>]/<опция>

где Command – наименование команды; arg – аргумент команды; ПР – параметр команды.

Наименование команды должно располагаться в начале командной строки и отделяться от остальных компонент. Порядок следования остальных компонент несущественен. Номера параметров определяются их взаимным расположением и увеличиваются слева направо. Параметры должны заключаться в скобки, максимальное количество параметров – 4. В случае необходимости, команду может сопровождать одна/несколько опций, обозначаемых косой чертой. Ошибочные опции игнорируются, о чем выводится соответствующее сообщение. Аргумент команды не должен содержать пробелов, так как считывается до первого пробела. Отметим, что малые и большие буквы латинского алфавита в командной строке система воспринимает одинаково. Максимальная длина аргумента/параметра не может превышать 80 символов (лишние символы отбрасываются).

Команда "RUN" предназначена для запуска в среде "CINDY" BAT- и EXE-модулей с передачей, если необходимо, данных из объекта, на который указывает маркер списка, или (автоматический режим) из первого помеченного объекта в списке.

RUN: <program/command> [1-й ПР]...[4-й ПР] /<опция>

где program/command – имя файла программного модуля (указывается обязательно); ПР – параметр команды.

Для передачи данных из объекта запускаемой программе необходимо в качестве первого параметра указать символ "*", а в качестве второго параметра

указать имя файла, в который будут копироваться данные из выбранного объекта. Например,

```
RUN: run.bat [*dan.dat] ,
RUN: program.exe [d:\proba\dan.dat] .
```

При запуске программы/команды указанное имя файла помещается в командную строку ОС после имени запускаемой программы/команды. Для ввода в командную строку дополнительной информации следует использовать 3-й и 4-й параметры. Например,

```
RUN:program.exe [d:\proba\dan.dat][G /F] ,
RUN: copy [dan1.dat][dan2.dat] .
```

Если в качестве первого параметра вместо символа "*" указано другое слово (например, имя файла), то оно помещается в командную строку ОС следом за именем запускаемой программы/команды без копирования данных из БД. Например,

```
RUN: program.exe [c:\proba\dan.dat /F] ,
RUN: copy [ dan1.dat dan2.dat] .
```

Отметим, что если в командной строке указано более одного параметра, то все указанные параметры будут помещены в командную строку.

Команда "PASS" предназначена для копирования (экспорта) содержимого объектов в файлы.

```
PASS: [*)<имя файла>] /<опция> .
```

Если в качестве второго параметра просто указано имя файла: pass:[*)<имя файла>], то объект, на который указывает маркер списка, или все помеченные объекты копируются в один указанный файл. Если в имени файла содержится символ "*" (например, pass:[*)<имя файла>], то помеченные объекты копируются в отдельные файлы с именами, соответствующими имени, указанному во втором параметре, а вместо символа "*" подставляются порядковые номера, под которыми копируются эти объекты. Например, [dan*.dat] = dan0.dat, dan1.dat

Команда "LOAD" предназначена для загрузки содержимого внешних файлов в БД ИУС "CINDY".

```
LOAD: [*)<имя файла>].
```

Файл копируется в объект, на который указывает маркер, либо в первый помеченный объект, если таковые имеются и система находится в автоматическом режиме. Если текущий/первый помеченный объект является списком, то загрузка данных не производится, и на дисплей выводится сообщение об ошибке. Если текущий/первый помеченный объект является таблицей, система проверяет загружаемый файл на соответствие формата файлов, получаемых при копировании табличных объектов в файлы без указания опции формата, и загружает данные только при соответствии форматов.

Если объект, в который копируется файл, уже содержит ранее введенную информацию, то до копирования на дисплей выводится меню с предложением

подтвердить необходимость обновления содержимого объекта. Для подтверждения следует нажать клавишу <Y>, а для отказа от обновления следует нажать клавишу <N> / <Esc>. Если нет необходимости подтверждать факт обновления данных, следует использовать опцию "/S". Например,

LOAD: [*][dan.dat] /s .

При загрузке текстов может возникнуть необходимость исключить из записываемой информации символы с кодом 26 (EOF). Например, если необходимо объединить в один файл содержимое заполняемого объекта с содержимым других текстовых объектов. Для исключения из текста символов EOF, следует использовать опцию "/ГХТ". Например,

LOAD: [*][dan.dat] /txt .

Команда "FIND" формирует временный список объектов текущего раздела, наименования и тип которых соответствуют наименованию и типу, указанным в параметрах команды. Кроме того, данная команда позволяет просмотреть список удаленных объектов, которые зарегистрированы в файле "DEL_A1.DAT" текущего раздела. Список, создаваемый командой "FIND", называется FIND-списком, и в первую очередь используется для поиска информации по нечетко сформулированным запросам пользователя.

FIND: [< наименование объекта >][< тип(ы) объекта >] .

При отсутствии второго параметра в FIND-списке регистрируются любые объекты, наименования которых соответствуют первому параметру команды. Например,

FIND: [Таблица N1] .

Если для конкретизации типа искомого объекта используется второй параметр команды, то тип объекта обозначается так: L – список, T – таблица, P – процедура, F – свободный объект. Отметим, что допускается задание несколько типов объектов для поиска. Например,

FIND: [Справочник][T,F] .

Можно уточнить имя искомого объекта, включив в него наименования вышестоящих списков. При этом не обязательно указывать имена всех вышестоящих списков. Наименования списков отделяются друг от друга и от наименования искомого объекта обратной косой чертой.

Например. Пусть в текущем разделе требуется найти таблицу с именем "Таблица 1", зарегистрированную в списке 5-го уровня "Новые таблицы", где соответствующий ей список 3-го уровня назван как "Справочник". Наименования вышестоящих списков 1-го, 2-го и 4-го уровней не требуются. Для поиска объекта, соответствующего указанным условиям, необходимо использовать формат:

FIND:[Справочник\ Новые таблицы\ Таблица 1][T] .

Если нет необходимости полностью указывать наименования искомого объекта и вышестоящих списков, или если неизвестно точное наименование искомого объекта, ИУС "CINDY" позволяет вести контекстный поиск объекта, т. е.

поиск на основании фрагмента(ов) наименования(й). Для активизации такого поиска каждому фрагменту наименования должны предшествовать не менее 3-х следующих подряд точек. При этом наименование искомого объекта и каждого из вышестоящих списков может задаваться не более, чем одним фрагментом. Например,

 FIND:[Справочник\Новые\ ... 1][Т] .

Отметим, что количество и расположение пробелов в фрагментах наименований должно соответствовать количеству и расположению пробелов в наименованиях соответствующих объектов. Например,

 FIND: [...ник\Новые\ ...1][Т] .

Если требуется выбрать все объекты какого-либо типа, то можно не указывать их наименования, достаточно указать их типы. Например,

 FIND: [[]Т] .

В команде FIND можно использовать такие опции: /first – ограничение поиска первым найденным объектом, наименование и тип которых соответствуют шаблону поиска; /U – активизация поиска объектов, удаленных из текущего списка, и формирование временного списка для просмотра и восстановления таких объектов; /G – употребляется совместно с опцией U, обеспечивая поиск всех удаленных объектов, зарегистрированных в файле "DEL_A1.DAT".

Если указана опция "/U", можно не указывать параметр команды. Например,

 FIND: /u /g .

Отметим, что если при активизации команды "FIND" использована опция "/U", то результирующий FIND-список будет содержать только удаленные объекты, и наоборот, если такой ключ отсутствует, FIND-список будет состоять только из существующих (неудаленных) объектов.

И наконец, если необходимо вернуться в список, который был загружен в соответствующее "окно" непосредственно перед формированием в нем FIND-списка, следует ввести команду:

 FIND: off .

Команда "CALL" предназначена для активизации автоматического режима работы ИУС "CINDY", где выполняется обработка ее процедур. Запуск процедуры, содержащейся в БД, в объекте соответствующего типа, осуществляется с помощью команды "CALL" следующего формата:

 Call:<имя процедуры>[I-й ПР]...[4-й ПР] ,

где параметры команды "Call" предназначены для передачи исходных данных активизируемой процедуре. Запуск процедуры, содержащейся во внешнем файле, производится с команд "Call" с опцией /ext:

 Call:<имя файла>[I-й ПР]...[4-й ПР]/ext .

Например,

 Call:Редактор_текстов [ре2][d:\tmp.txt] ,

Call:c:\ut_c1\editor.p[pe2][d:\tmp.txt] /ext .

Правила написания процедур. Для создания и редактирования процедур используется любой текстовый редактор, используемый для программирования. Язык "Procedure Manager" (PM), используемый для написания процедур, включает в себя полный набор команд ИУС "CINDY", а также набор операторов обеспечивающих определенный порядок выполнения команд и организацию диалога для работы пользователя [5]. По своему формату операторы аналогичны командам ИУС "CINDY", но, в отличие от команд, распознаются и выполняются только в автоматическом режиме. Наряду с командами и операторами язык PM содержит условный символ @, который позволяет вводить в текст процедуры комментарий. Последний может располагаться как на отдельной строке, так и на одной строке с командой/оператором (за исключением определителя метки перехода), но правее последнего значащего символа, относящегося к команде/оператору. Все символы, расположенные правее условного символа @, интерпретатором игнорируются.

Переменные. Каждая отдельная процедура оперирует с пятью беззнаковыми целочисленными переменными и пятью строковыми/символьными переменными. Все переменные имеют строго определенные обозначения, состоящие из ключевого символа и номера переменной (от 1 до 5), поэтому язык PM не имеет операторов описания переменных.

В качестве ключевого символа, обозначающего целочисленную переменную используется символ "&", например, &1, &2, &3, &4 и &5. Строковые переменные обозначаются ключевым символом "%", например, %1, %2, %3, %4 и %5. Номера переменных могут задаваться как непосредственно (%3, &2), так и параметрически с помощью целочисленных переменных (%&1, &&3).

Допустимые значения целочисленной переменной находятся в пределах от 0 до 65535. Каждая строковая переменная может включать в себя строку длиной до 80 символов. При попытке включения в символьную переменную строки длиной более 80 символов лишние символы отбрасываются. Все переменные являются локальными, т. е., если одна процедура вызывает другую процедуру, то значения переменных вызывающей процедуры не влияют на значения одноименных переменных вызываемой процедуры, и наоборот, значения переменных вызываемой процедуры не влияют на значения одноименных переменных вызывающей процедуры. По окончании работы вызываемой процедуры, значения ее переменных теряются.

Операторы языка PM. Язык PM содержит следующие операторы: операторы присвоения значений переменным – LET, LEN, SEARCH; арифметические операторы – ADD, SUB, MUL; операторы обмена данными с файлом – READ, WRITE; операторы передачи управления – CALL, GOTO, IF, ERROR, ELSE, RET, BYE; операторы работы с "окнами" и поддержки меню – WINDOW, MENU; специальные операторы – SOUND, CREATE, DELETE, BASE, STEP и оператор задания метки перехода – .:

Вместе с операторами передачи управления, в качестве параметров, употребляются следующие знаки операций сравнения: "=" – равно; "≠" – не равно; ">" – больше; "<" – меньше; ">=" – больше или равно (не меньше); "<=" – меньше или равно (не больше).

Формат операторов в основном аналогичен формату команд:

<NAME>: <arg> [<1-й ПР>]...[<4-й ПР>] ,

где NAME – наименование оператора; arg – аргумент оператора; ПР – параметр оператора.

Отличие формата операторов от формата команд заключается в том, что с операторами не употребляются опции.

Структурное программирование на языке РМ. Язык РМ имеет средства, позволяющие осуществлять структурное программирование, для чего используются конструкции, включающие в себя операторы условного перехода IF / ERROR, и фигурные скобки, "охватывающие" вложенный блок операторов и команд, выполняемых при реализации условия, описанного в предшествующем операторе условного перехода. Фигурные скобки рассматриваются системой как отдельные операторы и должны всегда располагаться на отдельных строках:

```
if: [&2][>=][3]
{ @ Пробный блок:
  sound: [200][5]
  run: %2 /L
  goto: final
}
error:
{
  Call: Обработка_ошибки [%2]
}
```

При наличии альтернативных вариантов передачи управления используется оператор "ELSE":

```
if: [&2][>=][3]
{
  @ Пробный блок:
  sound: [200][5]
  run: %2 /L
  goto: final
}
else
if: [&2][<][3]
{
  Call: %&3 [Запись N&1] /ext
  goto: final
}
else
{
```



```
goto: final  
}
```

На базе ИУС "CINDY" и расчетного пакета NASTRAN создана тестовая версия системы автоматизированного прочностного расчета крыла самолета. В рамках этой системы взаимодействие между объектами ИУС "CINDY" и пакетом NASTRAN осуществляется через временные файлы. При этом, кроме информационного обеспечения расчетов ИУС "CINDY" сформирован электронный банк технологий прочностного расчета [5].

Выводы. Предлагаемая программная оболочка для персональных компьютеров позволяет создавать удобную информационную поддержку пользователю при формировании и компоновке сред необходимых ПП, ориентированных на решение задач автоматизации научных и производственных экспериментов. Система имеет развитый интерфейс для обмена данными с прикладных программ и Windows-приложениями. Она также имеет полный набор средств, позволяющих объединять в ПП любые программные модули, полностью автоматизировать обмен данными внутри таких пакетов, а также обеспечить максимальные удобства для пользователей.

1. *Зінченко В.П., Зінченко Н.П.* Розробка методичної і програмної компонент інформаційної технології // Наукові вісті НТУУ "КПІ". – 2005. – № 2. – С. 98 – 109.
2. *Zinchenko V., Zinchenko S., Geraimchuk M. and els.* The analysis of tasks of search in difficult structures of the data // International scientific conference UNITECH'06. – Gabrovo: Technical university of Gabrovo, 2006. – P. I-362 – I-368.
3. *Борисов В., Зинченко В., Гераимчук И. и др.* Программная оболочка информационной поддержки пользовательского программного обеспечения // International Scientific Conference 23 – 24 November 2007, GABROVO, 2007.
4. *Зинченко В.П., Борисов В.В.* Исследование вопросов повышения качества разработки программного обеспечения PDM-систем за счет повышения независимости программных модулей // Открытые информационные и компьютерные интегрированные технологии. – Харьков: Гос. Аэроком. Ун-т "ХАИ", 2006. – Вып. № 10. – С. 125 – 131.
5. *Борисов В.В., Зинченко В.П.* Разработка прочностных моделей конструкций на основе метода последовательной детализации // VI Междунар. конф. "Гиротехнологии, навигация, управление движением и конструирование авиационно-космической техники". Сб. докл. в 2-х частях. – К.: НТУУ "КПИ", 2007. – Ч. II. – С. 55 – 61.

Получено 05.06.2014