



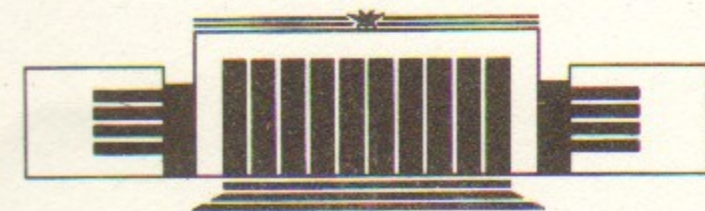
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ СО АН СССР

Ю.И. Мерзляков

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ПРОЦЕССОРА АП-32.**

4. Консольная подсистема. Библиотекарь

ПРЕПРИНТ 90-29



НОВОСИБИРСК

4. Консольная подсистема. Библиотекарь

Ю.И. Мерзляков

Институт ядерной физики
630090, Новосибирск 90, СССР

АННОТАЦИЯ

Описаны программы для работы с процессором АП-32 — мониторная программа CONSOL, реализующая операторскую консоль и отладчик фортрановских программ АП-32, а также библиотекарь объектных модулей.

7. КОНСОЛЬНАЯ ПОДСИСТЕМА АП-32

Арифметический процессор АП-32 предназначен для выполнения счетных алгоритмов и не имеет собственного канала ввода/вывода и пульта управления. Для полноценного использования процессора к нему подключается серийная ЭВМ семейства «Электроника», называемая дальше ЭВМ накачки (ЭВМН) [1]. Образованный таким образом многомашинный комплекс позволяет решать любую задачу, поставленную для универсальной ЭВМ.

Важными задачами, решаемыми при настройке, тестировании и эксплуатации любой ЭВМ, являются

- 1) обеспечение доступа оператора к содержимому запоминающих устройств ЭВМ (регистры и память) по считыванию и записи;
- 2) управление состоянием ЭВМ: пуск, останов, шаговый (командный) режим исполнения программы.

Эти задачи нужно решать и в АП-32; существуют разные способы их реализации.

В ранних ЭВМ использовался, как правило, пульт управления, представлявший собой множество индикаторов для отображения содержимого памяти и регистров ЭВМ и клавиш для ввода данных и управления, ввод и вывод в двоичном виде.

В более поздних ЭВМ пульт управления заменен консольным терминалом, работающим с микропрограммой этой ЭВМ. Ввод данных и управление осуществляется с клавиатуры терминала, а выводятся данные на его экран. В отличие от пульта, параллель-

ный доступ к набору данных здесь невозможен, регистры или адреса памяти нужно выбирать последовательно, но работать с терминалом быстрее и удобнее, чем с пультом.

В современных ЭВМ, организованных как многомашинные комплексы, могут использоваться консольные подсистемы, реализованные на одной из входящих в комплекс ЭВМ. Примером такого комплекса может служить VAX 11/780 с консольной ЭВМ LSI-11. Возможности, предоставляемые такой ЭВМ, значительно больше, чем в предыдущем случае, и к перечисленным задачам добавляется, например, тестирование главного процессора от консольной ЭВМ.

Очевидно, есть все условия и для реализации консольной подсистемы для АП-32. Так, ЭВМН, кроме функций обслуживания ввода/вывода и управления АП-32, приобретает также функцию консольной ЭВМ. В общем виде ее назначение — обеспечить доступ к АП-32 как при настройке и регламентных инженерных работах, так и при отладке программ, сочетая достоинства перечисленных выше способов. Кроме поставленных задач, здесь решались некоторые новые.

1. Создать программный «гибрид» пульта управления и консольного терминала, используя экран терминала
 - а) для отображения максимального количества полезной информации о содержимом памяти и регистров АП-32;
 - б) для изменения содержимого при помощи клавиатуры терминала путем модификации информации на экране.
2. Позволить выбирать нужные регистры и адреса памяти для одновременного отображения.
3. Обеспечить ввод и отображение этой информации в любом необходимом формате с возможностью выбора этого формата.
4. Дать возможность программировать АП-32 на языке ассемблера в режиме интерпретатора (ввод текста программы по одной команде, немедленная трансляция и загрузка в память) со сбросом по желанию полученной программы из памяти АП-32 на диск в виде образа задачи (для написания небольших тестовых или настройных программ).
5. Реализовать режимы загрузки и пуска программ в АП-32.
6. Предоставить средства отладки программ АП-32 (останов по заданному адресу, реализованный через механизм точек останова).
7. Применить экранное редактирование как основной метод модификации данных АП-32.

Консольная подсистема реализована программными средствами ЭВМН в двух вариантах:

- 1) программа CONSOL;
- 2) часть CONSOL, используемая в качестве отладчика в OTSAP — программе обслуживания фортрановских задач АП-32 [2].

CONSOL предназначена для инженерной деятельности и отладки программ АП-32, не обращающихся к ЭВМН с требованием обслуживания (ввод/вывод и т. п.). Отладчик, сохраняя большинство возможностей CONSOL, позволяет работать с программами, взаимодействующими с ЭВМН по протоколу (как правило, фортрановскими). Консольная подсистема написана на языках ФОРТРАН-IV и MACRO-11.

7.1. Возможности CONSOL

CONSOL — интерактивная программа, по характеру взаимодействия с оператором напоминающая экранный редактор текста.

Деятельность оператора можно свести к вводу текстовой информации, которая будет впоследствии преобразована в двоичный вид и записана в АП-32, и вводу служебной информации для считывания данных из АП-32 и вывода их на экран, переключения режимов CONSOL или управления АП-32. Объект модификации на экране, соответствующий регистру или ячейке памяти АП-32, выбирается посредством клавиш перемещения курсора. Текстовая информация вводится нажатием клавиш основной клавиатуры; текст можно редактировать. Как и в текстовых редакторах, служебные действия оператора подразделяются на простые, называемые функциями, и сложные, называемые командами. Функции реализуются нажатием клавиш на дополнительной клавиатуре терминала, а команды — путем перехода в специальный командный режим и ввода дополнительных параметров.

Функции CONSOL сводятся к следующим:

1. Выбор текущего объекта просмотра/модификации (регистр, память).
2. Выбор формата представления данных выбранного объекта или группы объектов. Возможны следующие варианты представления для ввода и вывода информации:
 - 1) восьмеричное слово/полуслово/байт;
 - 2) десятичное слово/полуслово;
 - 3) код RADIX-50;

- 4) текстовый код ASCII (КОИ-7);
 - 5) E-формат чисел с плавающей запятой;
 - 6) мнемонический код команд на языке ассемблера АП-32;
 - 7) двоичный код (только для вывода).
3. Физическая запись или физическое чтение информации АП-32 для выбранного объекта.
 4. Управление режимами исполнения функций, включение/выключение:
 - 1) чтения и отображения данных после записи (верификации);
 - 2) режима покомандного исполнения программы АП-32;
 - 3) режима трассировки данных во время покомандного исполнения программы (отображение текущего значения).
 5. Пуск/останов программы АП-32.
 6. Вставка/удаление точек останова в программе АП-32.
 7. Вывод подсказки — описания функций и команд CONSOL.
С помощью команд реализуются:
 - 1) загрузка файла программы АП-32 типа .JOB в память АП-32;
 - 2) создание файла образа задания типа .JOB с содержанием — образом указанных областей памяти АП-32;
 - 3) даммирование областей памяти АП-32 на устройство печати или в файл в указанном виде;
 - 4) заполнение указанных областей памяти АП-32 заданной величиной;
 - 5) вывод основной формы (образа экрана терминала) CONSOL на устройство печати или в файл;
 - 6) сравнение файла образа задания .JOB с содержимым соответствующих областей памяти.

7.2. Организация CONSOL

Программа CONSOL по структуре сходна с текстовым редактором. Она состоит из подпрограммы редактирования текстовой информации и ряда процедур, исполняющих команды и функции, задаваемые оператором.

Подпрограмма редактирования текста принимает всю вводимую оператором информацию. Алфавитно-цифровые данные вводятся в текстовый буфер, их можно редактировать в экранном режиме. Нажатие управляющих клавиш или клавиш дополнительной клавиатуры приводит к выходу из этой подпрограммы в диспетчер процедур исполнения функций/команд и запуску соответ-

ствующей процедуры. По завершении процедуры управление возвращается в редактор; в ряде случаев обновляется содержимое экрана.

Все процедуры, за исключением обработки функции пуска, исполняются определенное предсказуемое время, при пуске АП-32 это время зависит от длительности его программы, работающей до программного останова или ввода оператором функции останова. После пуска АП-32 блокируются все функции и команды, кроме останова и выбора текущего поля перемещением курсора. Это гарантирует целостность программы и данных АП-32 во время его работы. При работе АП-32 на экран выводится его процессорное время при помощи асинхронной подпрограммы, запускаемой по таймеру.

Взаимно однозначное соответствие содержимого памяти/регистров АП-32 и экранной информации устанавливается путем обмена данными между программными буферами CONSOL и аппаратурой АП-32, сопровождаемого преобразованием данных из двоичного вида в текстовый и обратно. CONSOL имеет два таких буфера:

- 1) текстовый буфер-образ экрана терминала;
- 2) буфер двоичных данных.

При записи данных в АП-32 информация, введенная с клавиатуры, фиксируется в текстовом буфере и одновременно выводится на экран. Затем текст преобразуется в двоичный вид, помещается в двоичный буфер, а оттуда копируется в АП-32.

При чтении данных из АП-32 они попадают в двоичный буфер, оттуда после преобразования — в текстовый, затем выводятся на экран.

Соответствие данных устанавливается дискретно, в следующие моменты:

- 1) при выполнении функции записи данных в регистр или память АП-32, верификация (контрольное считывание) гарантирует результат;
- 2) при выполнении функции считывания регистра/памяти или группы таких объектов АП-32;
- 3) на каждом шаге исполнения программы АП-32 в пошаговом режиме с трассировкой;
- 4) при самостоятельном или принудительном его останове из рабочего состояния.

В случаях 1 и 2 операция считывания обеспечивает соответствие данных. В случае 3 последняя исполненная команда АП-32

подвергается интерпретации, в ходе которой выясняется, на какие объекты, выводимые в данный момент на экран, она оказала воздействие. Например, если это команда загрузки в регистр, которая изменила его содержимое, то произойдет считывание этого регистра и вывод содержимого на экран. Этот прием позволил не обновлять всю информацию экрана и экономить время. В случае 4 происходит считывание и отображение всех объектов АП-32, выбранных на экране.

При исполнении некоторых функций экран терминала модифицируется сразу в нескольких местах. При частом повторном их исполнении объем выводимой информации становится значительным, что может привести к перегрузке операционной системы и потере данных. Для устранения перегрузки применена оптимизация: вся вводимая информация обрабатывается, результаты помещаются в буфер экрана, но вывод на экран откладывается до момента прекращения ввода данных оператором. Данные на экране не будут соответствовать результату всех исполненных функций.

Для преобразования данных в CONSOL реализована подсистема преобразования — набор подпрограмм, вызываемых процедурами обработки функций/команд. В нее, кроме подпрограмм форматного преобразования, входят строчный ассемблер и реассемблер команд АП-32. Последние транслируют операторы языка ассемблера АП-32 в машинные коды и восстанавливают текст оператора по машинному коду. В ассемблере использован табличный метод разбора, описанный в [3]. Реассемблер использует таблицы машинного кода и мнемоник ассемблера.

Для операций ввода/вывода и управления АП-32 используется пакет подпрограмм IOAP32, работающий с АП-32 через драйвер [1].

Загрузчик программ АП-32 из файла образа задачи и формирователь такого файла из содержимого заданных областей памяти содержатся в соответствующих процедурах обработки команд.

7.3. Вызов и прекращение работы CONSOL

Программа CONSOL работает с терминалом, совместимым по кодировке с DECscope VT52 и использует все 80 позиций строки, поэтому терминал нужно установить в режим без автоматического перевода строки из 80-й позиции, если это не режим по умолчанию. Программа также использует команды $\langle \text{ESC} \rangle \text{T}$ и $\langle \text{ESC} \rangle \text{U}$ для включения/выключения подчеркивания (подсвечивания) объ-

екта, что работает не на всех терминалах. Нестандартный выход из программы (перезагрузка машины, например) может привести к тому, что терминал останется в режиме HOLD SCREEN, из которого нужно выйти сбросом терминала или его выключением. Нормальный выход из CONSOL переводит терминал в исходное состояние.

Программа запускается мониторной командой R с устройства SY: R CONSOL. При запуске загружается драйвер АП-32 и включаются режимы терминала ALTERNATE KEYPAD и HOLD SCREEN. Курсор устанавливается на первый объект, и программа готова к работе.

При запуске программы возможны отказы. Наиболее типичные причины — отсутствие драйвера процессора АП-32 AP.SYS или AP.TSX на устройстве SY: или недостаточный объем памяти (чаще в RT-11). Сообщения о невозможности запуска выводятся на терминал в виде сообщения об ошибке и номера этой ошибки.

Для прекращения работы программы и выхода в монитор операционной системы необходимо дважды нажать $\langle \text{CTRL/C} \rangle$, после чего экран терминала очищается, и выводится сообщение о завершении работы программы. Прекратить работу программы можно только в режиме редактирования.

7.4. Отладчик фортрановских программ АП-32

При работе с фортрановскими программами АП-32 можно использовать отладчик, представляющий собой подмножество программы CONSOL. Отладчик реализован как часть программы OTSAP, участвующей в создании операционной среды фортрановских программ АП-32. Переход в состояние отладчика происходит при пуске программы АП-32, в результате ошибки ее исполнения или прохода через точку останова этой программы. В отличие от CONSOL отладчик не имеет командного режима.

В [2] описаны два способа пуска программ — с отладчиком и без него. При пуске программ без отладчика (нормальный режим) точек останова в программе АП-32 нет. Ошибки исполнения программы перехватываются исполнительной системой, которая формирует и выводит сообщения о них. Ряд же ошибок, не обрабатываемых исполнительной системой (например, непредусмотренный останов программы по команде HALT), приводят к переходу в состояние отладчика. При этом на экране терминала возникает основная

форма программы CONSOL, и оператор может заняться выяснением причины ошибки.

При пуске программы с отладчиком любая ошибка исполнения, включая те, которые нормально обрабатываются исполнительной системой (переполнения и т. п.) переводят в состояние отладчика. Исходная позиция при таком пуске также отличается. Если при пуске без отладчика программа АП-32 немедленно запускается, то при пуске с отладчиком произойдет переход в его состояние, после чего можно расставить необходимые точки останова. Следует заметить, что программа будет запущена со стартового адреса независимо от занесения данных в программный счетчик.

Существует также возможность прервать исполнение программы АП-32 и перейти в состояние отладчика. Для этого необходимо ввести CTRL/↑, после чего АП-32 остановится, и на экран выведется форма отладчика.

Для выхода из состояния отладчика необходимо воспользоваться функцией <ПУСК>. Адрес пуска — стартовый адрес программы (первый раз) или адрес останова + 1. Изменить пусковой адрес невозможно, это сделано специально, так как фортрановская программа привязана к исполнительной системе, и изменение хода программы может привести к ее разрушению. При пуске экран терминала очищается, и на него могут быть выведены сообщения программы АП-32.

8. LIBRAB — БИБЛИОТЕКАРЬ АП-32

Библиотекарь процессора АП-32 — программа для создания библиотек объектных модулей АП-32, используемых позднее при сборке его рабочих программ, а также макроблиотек, используемых макроассемблером при трансляции ассемблерных программ.

Библиотечный файл — это файл с прямым доступом (файл, который имеет каталог), который содержит один или более модулей одного и того же типа. Библиотекарь организует библиотечный файл таким образом, что редактор связей и макроассемблер могут быстро извлекать из них модули.

Библиотека представляет собой файл, состоящий из заголовка, таблицы входных точек, одного или нескольких объектных модулей. Объектными модулями в библиотеке могут быть подпрограммы, многократно используемые в программе, подпрограммы,

используемые более чем в одной программе, или подпрограммы, связанные между собой и собранные в одно место для удобства. Нормально, если в библиотеке нет главных программ.

Примером объектного библиотечного файла является системный файл APLIB.OBT, который используется редактором связей. Примером библиотечного файла макрокоманд является ARMAC.SML, который используется макроассемблером для обработки системных макрокоманд.

Можно извлекать объектные модули из библиотечного файла путем создания в программе ссылок к глобальным символам библиотеки; затем эти объектные модули подшиваются к программе в файле образа задачи.

При помощи библиотекаря над библиотечным файлом можно выполнять следующие функции:

- 1) создание библиотечного файла;
- 2) вставка и замена объектных модулей;
- 3) удаление объектного модуля;
- 4) извлечение объектного модуля;
- 5) получение листинга содержимого объектного библиотечного файла;
- 6) манипуляции каталогом библиотеки.

Библиотекарь организован как набор процедур для исполнения этих функций. В начале работы принимается и разбирается командная строка, открываются входные и выходные файлы, распределяется память и вызывается процедура, соответствующая заданию в командной строке. По ее завершении повторяется прием командной строки.

При создании или обновлении библиотеки совершается два прохода, т. е. два просмотра входных файлов. В первом проходе из всех входных объектных модулей извлекаются глобальные символы, а также имена модулей и программных секций (если требовалось) и в оперативной памяти строится каталог библиотеки. Размер каталога и, соответственно, емкость таблицы определяется только буфером, отводимым в оперативной памяти. Во втором проходе каталог записывается в выходной файл, а следом за ним — конкатенация объектных модулей из входных файлов. Каталог содержит имя глобальных символов и их смещения относительно начала библиотечного файла.

При извлечении модулей, стирании их или работе с каталогом только каталог может подвергнуться модификации, объектные модули в библиотеке не изменяются.

Работа с макробibliothеками в целом аналогична работе с объектными библиотеками, за исключением того, что обрабатывается текстовая информация и отсутствуют операции извлечения модулей и распечатки каталога.

8.1. Вызов и прекращение работы библиотекаря

Программа LIBRAP вызывается с системного устройства с помощью команды R LIBRAP. Командная строка библиотекаря имеет следующий вид:

```
DEV:FLIB,DEV:FLIST=DEV:FINP1,DEV:FINP2,...,FINPN/S
```

где DEV—имя устройства; FLIB—имя создаваемого библиотечного файла; FLIST—имя файла для листинга библиотеки; FINP1, ..., FINPN—имена входных файлов, модули которых должны войти в библиотечный файл; /S—один из ключей команд.

Каждый входной файл хранится на указанном устройстве под заданным именем и типом. Существуют следующие типы файлов по умолчанию:

Файл	Тип файла
Библиотечный файл	.OBT
Входной файл	.OBT
Справочный файл	.LST

Входной файл может состоять из одного или нескольких объектных модулей. После того, как входной файл помещается в библиотеку, к любому из составляющих его модулей можно обратиться только по имени, указанному в директиве ассемблера .TITLE, а не по имени файла.

Библиотекарь позволяет создавать также библиотеки макроопределений. LIBRAP не создает файл листинга каталога для макробibliothек.

Типом входного и выходного файла по умолчанию является .MAC. Библиотечный файл не должен иметь то же имя, что и входной файл.

ЛИТЕРАТУРА

1. Аксенов Г.А. и др. Универсальный арифметический процессор АП-32. 3. Канал управления и ввода/вывода, драйвер операционной системы.—Препринт ИЯФ 90-3. Новосибирск, 1990.
2. Аксенов Г.А. и др. Программное обеспечение процессора АП-32. Комплекс

3. Мерзляков Ю.И. и др. Универсальный арифметический процессор АП-32. 2. Ассемблер. Редактор связей.—Препринт ИЯФ 90-30. Новосибирск, 1990.

Приложение 1

ФУНКЦИИ И КОМАНДЫ CONSOL

Объекты на экране терминала

При запуске программы на экран терминала выводится основная картинка (форма), содержащая поля, подвергаемые вводу/выводу, и пояснительные надписи. Эта форма моделирует операторский пульт АП-32.

Поля соответствуют конкретным физическим объектам (I-, F-регистры и ячейки памяти АП-32) или служат для их выбора или других служебных целей. По способу обращения с полями их можно разделить на

- 1) RO—поля только для чтения (вывода на экран);
- 2) WO—поля только для записи (ввода с клавиатуры);
- 3) RW—поля для чтения и записи;
- 4) MS—поля типа сообщений; то же, что RO, но их нельзя выбрать установкой курсора.

Такое разделение зависит от типа объекта, соответствующего этому полю. Невозможно, например, записать что-либо в выходной регистр сумматора АП-32 или бессмысленно считывать номер I-регистра, используемый для его выбора. Поля описываются ниже в порядке их расположения в форме, строка за строкой. Строки нумеруются сверху вниз, 1—24. Сама форма в документ не помещается.

Строка 1—служебные поля

В строке 1 находится надпись—имя программы и ее версия. В следующих полях типа сообщений выводится текущая дата, текущее время (раз в 5 секунд) и астрономическое время работы программы АП-32 с момента ее пуска с точностью до тика (1/50 секунды).

Строка 3—регистры АП-32

В строке 3 несколько полей разного назначения, перечисление слева направо:

1. Поле сообщений программы. Содержит текстовое сообщение о последней ошибке аппаратуры или оператора или информационное сообщение о последней исполненной функции драйвера АП-32.
2. Поле кода последней исполненной или исполняемой операции драйвера АП-32, три буквы из имени функции драйвера (см. его описание).
3. Поле, обозначенное «\$\$», типа RW, для установки режима работы драйвера, см. описание драйвера. По умолчанию установлен режим ввода/вывода из памяти АП-32 по каналу прямого доступа, для TSX-Plus установлен нормальный режим работы драйвера, а для RT-11 — режим драйвера как библиотеки подпрограмм.
4. Поле счетчика команд АП-32, обозначено «РС». Служит для чтения/записи в счетчик команд.
5. Поле регистра связи АП-32, обозначено «LR», также типа RW.
6. Поле регистра состояний интерфейса АП-32, только для сообщений.
7. Поле регистра флагов интерфейса АП-32, только для сообщений.
Все поля этой строки рассчитаны на ввод/вывод восьмеричных чисел.

Строки 5—8 — I-регистры ПЦЧ

В этих строках содержатся 16 групп из двух полей — номера I-регистра и его содержимого. Номер I-регистра — поле типа WO, служит для его выбора. Слева от него поле содержимого, типа RW, для чтения/записи информации I-регистра. По умолчанию номера I-регистров установлены 0—15, допустимы номера до 31. Содержимое I-регистров представлено по умолчанию в восьмеричном виде.

Строки 10—13 — F-регистры ППЗ

В этих строках помещены поля входных и выходных регистров арифметики чисел с плавающей запятой. Сюда входят:

1. Строка 10 — сумматор, регистры:
 - 1) AD — слагаемое, типа WO, по умолчанию — E-формат;
 - 2) AR — слагаемое, типа WO, по умолчанию — E-формат;
 - 3) AF — сумма, типа RO, по умолчанию — E-формат;
 - 4) AM — абсолютная величина суммы, типа RO, по умолчанию — E-формат.

2. Строка 11 — вычитатель, регистры:
 - 1) SB — уменьшаемое, типа WO, по умолчанию — E-формат;
 - 2) SR — вычитаемое, типа WO, по умолчанию — E-формат;
 - 3) SF — разность, типа RO, по умолчанию — E-формат;
 - 4) SM — абсолютная величина разности, типа RO, по умолчанию — E-формат.
3. Строка 12 — умножитель
 - 1) MP — множимое, типа WO, по умолчанию — E-формат;
 - 2) MR — множитель, типа WO, по умолчанию — E-формат;
 - 3) MF — произведение, типа RO, по умолчанию — E-формат.
 Последние два поля в этой строке — номер F-регистра (можно задать любой, по умолчанию 12₈, регистр графического дисплея) и значение этого регистра (восьмеричное по умолчанию).
4. Строка 13 — блок специальных функций. Четыре поля этой строки — содержимое регистров этого блока (восьмеричное по умолчанию).

Строки 15—18 — память данных

Первое поле строки 15 — адрес памяти данных, поле типа WO, восьмеричное значение. Остальные четыре поля, как и четверки полей в строках 16—18, — содержимое ячеек памяти, расположенных по смежным адресам, начиная с задаваемого адреса памяти данных. Эти поля имеют атрибут RW и могут быть представлены в любом виде, кроме мнемонических кодов.

Строка 20 — двоичное представление

Двоичное представление любого выбранного в текущий момент времени объекта выводится в этой строке. Это поле типа сообщения.

Строка 22 — точки останова

В строке расположено 8 полей восьмеричных значений точек останова типа WO, применяемых при отладке программ. Нулевое значение поля говорит об отсутствии адреса останова (это значит, что нулевое значение адреса программной памяти зарезервировано, на адресе 0 ставить точку останова нельзя). При ненулевом значении поля программа АП-32, работающего в автономном режиме, будет остановлена по этому адресу.

Строка 24 — объекты программной памяти

Первое поле этой строки — адрес программной памяти, поле типа WO, восьмеричное значение. Следующие два поля — дублирующие друг друга, значение ячейки программной памяти с заданным адресом. Первое из них — восьмеричное значение, а второе — мнемонический эквивалент на языке ассемблера, оба поля типа RW. Последние 4 поля в строке — однобитовые значения флагов процессора чисел с фиксированной запятой, поля типа сообщений.

Надписи

Пропущенные в описании строки содержат пояснительные надписи, которые при работе программы не изменяются.

Текущее поле

Текущим считается поле, на котором в настоящий момент находится курсор. Такое поле на экране помещается в угловые скобки и подчеркивается (подсвечивается), если в терминале реализовано подчеркивание. Поле становится текущим, когда оно выбирается функциями перемещения курсора. При пуске программы текущим становится поле выбора режима драйвера АП-32, обозначенное «§§».

Группа объектов

Некоторые функции касаются групп объектов (полей). К таким группам относятся:

- 1) содержимое I-регистров (16 объектов);
- 2) содержимое F-регистров (сумматор, вычитатель, умножитель, 11 объектов);
- 3) содержимое памяти данных (16 объектов).

Функции CONSOL

Для вызова функций использована дополнительная клавиатура терминала типа VT52, каждая клавиша которой, как правило, управляет двумя функциями. Прямая функция запускается нажатием на клавишу, а для выполнения альтернативной функции предварительно нажимают специальную клавишу-префикс, а затем нужную клавишу. Возможности, реализованные при помощи функций, описаны выше.

Ниже описывается каждая функция, в порядке расположения клавиш на клавиатуре. Так как клавиатуры у терминалов, совместимых с VT52, могут отличаться по расположению клавиш и над-

писям на них, за основу при описании берется «классический» вариант дополнительной клавиатуры VT52. Обозначения функций совпадают с надписями на клавишах в подсказке программы, в тексте они взяты в угловые скобки. В описании первая функция прямая, вторая — альтернативная (если есть).

Клавиши основной клавиатуры

При нажатии любой клавиши основной клавиатуры происходит занесение знака в текущее поле, в позицию над курсором. Курсор затем смещается в соседнюю позицию правее. Если текущая позиция — последняя в поле, перемещение курсора не происходит. Новое вводимое значение замещает старое в текущей позиции (а не вставляется). Горизонтальная табуляция текста работает только в поле мнемонических кодов (команд на языке ассемблера) для разделения операндов.

GOLD — левая клавиша в верхнем ряду (PF1)

Префиксная клавиша, используется для вызова альтернативных функций. Сама никакие функции не вызывает.

Средняя клавиша в верхнем ряду (PF2)

⟨HELP⟩ — вывод на экран терминала краткого описания функций и команд CONSOL аналогично редактору текста K52 и т. п. По первому нажатию экран очищается и выводится схематическое изображение дополнительной клавиатуры VT52 с краткими надписями на каждой клавише, обозначающими функции. По второму нажатию на экран выводится краткое описание команд. При последующих нажатиях все повторяется. Возврат в режим редактирования — при нажатии любой клавиши.

⟨0 → АКК⟩ — очистка (занесение нуля) всех 32 I-регистров.

Правая клавиша в верхнем ряду (PF3)

⟨ПУСК⟩ — пуск программы АП-32 на один шаг, если установлен режим покомандного исполнения программы (см. ниже), или начало работы АП-32 в автономном режиме. Пуск происходит с адреса, занесенного в поле «РС» (программный счетчик). В автономном режиме запускается счетчик процессорного времени. В шаговом режиме объекты, связанные с исполнением текущей команды, после исполнения модифицируются на экране (если они там есть), например, содержимое регистров или памяти, если

включен режим трассировки (см. ниже). В нижней строке появляется код и мнемоника следующей команды.

〈ОСТАНОВ〉 — останов АП-32, если он находился в автономном режиме, иначе ничего не происходит. По останову информация на экране терминала полностью обновляется и соответствует последней исполненной команде АП-32.

Клавиша «0»

〈СБРОС ТАЙМЕРА〉 — сбрасывает таймер-счетчик процессорного времени АП-32 в верхнем правом углу экрана.

〈ПОЛУСЛОВА〉 — переводит поле выбранного 32-разрядного объекта (регистр, память) в полусловное представление. Первым выводится старшее полуслово, вторым младшее, они разделяются пробелом.

Клавиша «1»

〈СЛЕДУЮЩИЙ〉 — переход на следующее поле, которое можно редактировать. Если действует режим перемещения вперед, следующее поле — правее в той же строке или первое в следующей. Если действует режим перемещения назад, следующее поле — левее в той же строке или последнее в предыдущей. Поля только для вывода и надписи пропускаются.

〈СЛОВО〉 — для 32-разрядных объектов (содержимое регистров, памяти) происходит переход к словному представлению информации.

Клавиша «2»

〈16 АКК〉 — выбор набора 16 I-регистров. При поочередном нажатии клавиши высвечиваются на экране I-регистры 0—16 или 17—31 в соответствующих полях.

〈БАЙТЫ〉 — для 32-разрядных объектов (содержимое регистров, памяти) происходит переход к байтовому представлению информации. Байты выводятся, в отличие от полуслов, в порядке старшинства слева направо.

Клавиша «3»

〈Е-ФОРМАТ〉 — для 32-разрядных объектов (содержимое регистров, памяти) переход к представлению информации в формате с плавающей запятой. Формат соответствует фортрановскому формату E15.7 (ширина поля 15 знаков, 7 знаков после точки).

〈ГРУППА〉 — команда 〈Е-ФОРМАТ_i〉, исполняемая для всей группы, к которой принадлежит объект.

Клавиша «4»

〈ВПЕРЕД〉 — выбор направления движения для функции 〈СЛЕДУЮЩИЙ〉, от начала экрана к концу.

〈В КОНЕЦ〉 — выбор в качестве текущего объекта последнего поля — мнемоническое изображение команды.

Клавиша «5»

〈НАЗАД〉 — выбор направления движения для функции 〈СЛЕДУЮЩИЙ〉, от конца экрана к началу.

〈В НАЧАЛО〉 — выбор в качестве текущего объекта поля значение счетчика команд, поле, обозначенное «РС».

Клавиша «6»

〈ДЕСЯТИЧ〉 — переход к десятичному представлению значения текущего поля.

〈ГРУППА〉 — команда 〈ДЕСЯТИЧ〉, исполняемая для всей группы, к которой принадлежит объект.

Клавиша «7»

〈ОБНОВИТЬ〉 — считать значения физических объектов, представленных на экране, из АП-32 и вывести их на экран в текущем формате.

〈КОМАНДА〉 — прием команды. Экран очищается, в левом верхнем углу выводится надпись «Команда: », и программа ждет от оператора ввода команды.

Клавиша «8»

〈ШАГОВЫЙ〉 — объявление шагового (покомандного) режима исполнения программы АП-32. Следующая команда 〈ПУСК〉 приведет к исполнению одной команды.

〈ОТМЕНА〉 — отмена шагового режима. Следующая команда 〈ПУСК〉 приведет к пуску программы в АП-32.

Клавиша «9»

〈ВОСЬМЕР〉 — переход к восьмеричному представлению значения текущего поля.

〈ГРУППА〉 — команда 〈ВОСЬМЕР〉, исполняемая для всей группы, к которой принадлежит объект.

Клавиша «.»

〈ЧТЕНИЕ〉 — для физических объектов на экране (регистры, память АП-32) — физическое чтение текущего объекта с выводом содержимого в соответствующее поле.

〈ВЕРИФИЦИР〉 — включение режима верификации записи по функции 〈ЗАПИСЬ〉 чтением с выводом результата чтения в соответствующее поле.

Клавиша «ENTER»

〈ЗАПИСЬ〉 — для физических объектов на экране (регистры, память АП-32) — физическая запись информации в текущий объект из соответствующего поля. Для полей, предназначенных для выбора номера (адреса) физического объекта, 〈ЗАПИСЬ〉 означает считывание содержимого этого объекта (для памяти данных — значений 16 слов) и вывод на экран в текущем формате. Для остальных полей эта функция означает фиксацию набранного в данном поле.

〈НЕ ВЕРИФ〉 — отмена режима верификации записи чтением для текущего физического объекта.

Клавиша «DELETE»

Стирание набранного значения выбранного поля и восстановление исходной величины.

Клавиша «RETURN»

Выбор следующего адреса. Эта функция действует только при выборе поля адреса памяти данных или программ или любого поля памяти данных или программ. Адрес памяти программ увеличивается на 1 для памяти программ и на 10_8 для памяти данных. Затем считывается одно слово памяти программ или 16 слов памяти данных и выводится на экран в соответствующих форматах.

Клавиша «LINE FEED»

Выбор предыдущего адреса. Все аналогично предыдущей функции, но адреса не увеличиваются, а уменьшаются.

Клавиша «CTRL/W»

Экран очищается и перерисовывается заново. Полезно при работе в мультипрограммной системе или при переключении терминала.

Клавиша «CTRL/R»

Действие эквивалентно действиям функций 〈ОБНОВИТЬ〉 и 〈CTRL/W〉.

Клавиша «TAB»

Переход к представлению 32-разрядных объектов в виде четырех знаков ASCII (КОИ-7). Знаки выводятся, как и байты, по старшинству слева направо.

Клавиша «FORM FEED»

Переход к представлению 32-разрядных объектов в виде шести знаков кода RADIX-50.

Клавиша «Курсор вверх»

Переход к полю, расположенному над текущим.
〈ТРАССИР〉 — включение режима трассировки в шаговом режиме исполнения программы. Под трассировкой понимается изменение на экране значений физических объектов по мере исполнения команд, которые их касаются. Например, исполнение команды загрузки I-регистра приведет к высвечиванию загруженного значения в поле данного I-регистра. Трассируемые объекты не имеют отношения к выбранному в данный момент, но должны присутствовать на экране.

Клавиша «Курсор вниз»

Переход к полю, расположенному под текущим.
〈НЕ ТРАСС〉 — выключение описанного выше режима трассировки.

Клавиша «Курсор вправо»

Переход к следующему в выбранном поле знаку при редактировании. Если поле кончилось, курсор переходит в начало следующего поля, а величина текущего поля восстанавливается в старую (новая не фиксируется, если и была введена). Если поле последнее, курсор остается на месте.

Клавиша «Курсор влево»

Переход к предыдущему в выбранном поле знаку при редактировании. Если поле кончилось, курсор переходит в начало предыдущего поля, а величина текущего поля восстанавливается в старую (новая не фиксируется, если и была введена). Если поле первое, курсор остается на месте.

Команды CONSOL

Команды предназначены для исполнения сложных операций, связанных с файловыми операциями и требующих ввода дополнительных параметров. После ввода команды при помощи функции `<GOLD>` `<КОМАНДА>` экран очищается, и выводится специальная форма для ввода параметров команды. Форма содержит от одного до семи полей, которые нужно заполнить, количество полей зависит от конкретной команды. Форма занимает одну строку и может содержать следующие поля:

1. ИМЯ ФАЙЛА — спецификация файла в терминах RT-11, т. е. имя устройства, имя файла, расширение и необязательный размер.
2. D-ПАМЯТЬ — два поля, обозначающие начальный и конечный адреса области памяти данных, над которой производится операция (восьмеричные величины).
3. I-ПАМЯТЬ — два поля, обозначающие начальный и конечный адреса области памяти программ, над которой производится операция (восьмеричные величины).
4. Поле, принимающее разные величины в зависимости от команды:
 - 1) СТАРТ — стартовый адрес программы (восьмеричная величина);
 - 2) ВЕЛИЧИНА — 32-битовое число, которое заносится в память АП-32 (восьмеричная величина);
 - 3) O/D/E/A — поле для выбора вида представления содержимого памяти, вводится одна буква из перечисленных:
 - О — восьмеричная величина;
 - D — десятичная величина;
 - E — E-формат чисел с плавающей запятой;
 - A — команда на языке ассемблера (только для памяти программ).
5. I/D — однобуквенное поле для выбора типа памяти АП-32, над которым производится операция, I соответствует памяти программ, D — памяти данных.

При выводе формы в поля выводятся значения по умолчанию, которые могут быть исправлены или оставлены в неизменном виде.

При редактировании полей допустимы следующие функции:

- 1) `<КУРСОР ВПРАВО>` — перемещение курсора в поле и на следующее поле;

- 2) `<КУРСОР ВЛЕВО>` — перемещение курсора в поле и на предыдущее поле;
- 3) `<СЛЕДУЮЩИЙ>` — перемещение курсора на следующее поле;
- 4) `<ЗАПИСЬ>` — фиксация содержимого текущего поля;
- 5) `<ПУСК>` — исполнение команды;
- 6) `<DELETE>` — отмена командного режима, возврат в режим редактирования.

Если при исполнении команды возникает ошибка, в нижней строке экрана выводится сообщение об ошибке, а в середине экрана надпись PAUSE — НАЖМИ `<RETURN>`. После того, как оператор прочитает сообщение об ошибке, он нажимает `<RETURN>` и переходит в режим редактирования. При нормальном завершении исполнения команды возврат в режим редактирования происходит без дополнительных сообщений. Ниже приводится описание команд. В квадратных скобках находится часть имени команды, которую можно не вводить.

F[ILL] — заполнение памяти АП-32 заданной величиной.

Команда предназначена для заполнения указанных областей памяти введенной величиной. Одна команда работает с каким-либо одним типом памяти. Поля D-ПАМЯТЬ или I-ПАМЯТЬ должны содержать начальный и конечный адреса заполнения, поле ВЕЛИЧИНА — численную константу заполнения (по умолчанию 0), а поле I/D — тип заполняемой памяти.

L[OAD] — загрузка программы в АП-32.

Команда загружает программу из файла типа .JOB (образ задачи АП-32) в память программ и память данных АП-32. Требуется заполнить только поле ИМЯ ФАЙЛА именем файла образа задачи; поля D-ПАМЯТЬ, I-ПАМЯТЬ и СТАРТ служат для вывода считанных из файла образа задачи, границ занимаемой памяти и стартового адреса задачи.

S[AVE] — сброс содержимого памяти АП-32 в файл.

Команда служит для сброса в указанный в поле ИМЯ ФАЙЛА файл областей памяти АП-32, границы которых заданы в полях D-ПАМЯТЬ и I-ПАМЯТЬ. Создаваемый таким образом файл по структуре соответствует файлу образа задачи типа .JOB.

D[UMP] — вывод содержимого памяти АП-32 в читаемом виде.

Команда выводит содержимое памяти заданного в поле I/D

типа в пределах, указанных в полях D-ПАМЯТЬ или I-ПАМЯТЬ, на устройство или в файл, указанный в поле ИМЯ ФАЙЛА. Вид выводимой информации определяется полем O/D/E/A. При выводе на терминал любое нажатие клавиши прекращает вывод.

P[PRINT] — сброс содержимого экрана CONSOL в файл.

В файл или на устройство, указанное в поле ИМЯ ФАЙЛА, выводится содержание экрана (основная форма) CONSOL со всеми текущими значениями. Применяется для протоколирования текущего состояния процессора или программы.

S[OMP] — сравнение файла программы и содержимого памяти.

Содержимое файла образа задачи, указанного в поле ИМЯ ФАЙЛА, сравнивается с содержимым соответствующих областей памяти АП-32. Границы областей извлекаются из файла образа задачи. Это способ проверки целостности памяти.

Сообщения об ошибках

Сообщения об ошибках выводятся тремя способами:

1. Если при пуске CONSOL не выполнены какие-либо условия или произошла аппаратная ошибка, экран терминала очищается, выводится фатальное сообщение, и исполнение программы CONSOL прекращается.
2. При работе в режиме редактирования сообщения об ошибках выводятся в поле сообщений в начале строки 3. В отличие от информационных сообщений сообщения об ошибках сопровождаются звуковым сигналом терминала и начинаются с вопросительного знака.
3. В командном режиме сообщения об ошибках выводятся в последней строке экрана терминала. Затем программа ожидает реакции оператора в виде ввода <RETURN>, после чего происходит переход в режим редактирования.

Сообщения об ошибках выводятся по-русски и не требуют дополнительных пояснений.

КЛЮЧИ БИБЛИОТЕКАРЯ LIBRAP

Для обслуживания библиотечного файла используются ключи команд. Если ключ не указан, то библиотекарь автоматически вводит модули в библиотечный файл. Ниже описаны ключи, задающие режимы работы LIBRAP.

Ключи продолжения команды (/C и //). Ключ продолжения используется, когда команда не помещается в одной строке. Максимальное число входных файлов на одной строке равно шести; ключ /C или ключ // используется для указания большего количества файлов. Следует вводить /C в конце текущей строки и повторять его в конце последующих строк столько раз, сколько необходимо. Если произойдет переполнение памяти, то печатается сообщение об ошибке. Каждая строка продолжения после первой командной строки может содержать только спецификацию входного файла. Не следует указывать /C в последней строке команды. Если используется ключ //, то он печатается в конце первой и последней входной строки.

Ключ удаления (/D). Ключ /D исключает модули и все их глобальные символы из библиотеки. Так как символы удаляются только из каталога библиотеки, они восстанавливаются при очередной модификации библиотеки. Когда употребляется ключ /D, библиотекарь печатает:

MODULE NAME? (имя модуля?)

Следует ввести имя модуля, законченное возвратом каретки. Вопрос будет повторяться, пока не будут введены имена всех модулей. Чтобы окончить ввод и инициировать выполнение командной строки, надо ввести возврат каретки сразу после сообщения MODULE NAME?.

Ключ извлечения (/E). Ключ /E позволяет извлекать объектный модуль из библиотечного файла и помещать его в файл типа .OBJ. При указании ключа /E библиотекарь печатает:

GLOBAL?

Если указано глобальное имя, то библиотекарь извлекает весь модуль, частью которого является это имя.

Нельзя использовать ключ /E в одной и той же командной строке с любым другим ключом.

Ключ удаления глобальных символов (/G). Ключ /G позволяет стирать спецификацию глобальных символов из каталога библиотечного файла. При использовании ключа /G библиотекарь печатает:

GLOBAL?

В ответ нужно напечатать имя глобального символа, подлежащего удалению, за которым следует возврат каретки, и продолжать так до тех пор, пока не будут введены все глобальные символы, подлежащие стиранию. После сообщения GLOBAL? необходимо напечатать возврат каретки, чтобы окончить ввод и начать выполнение командной строки.

Библиотекарь стирает глобальные символы только из каталога (а не из самой библиотеки). Как только обновлен библиотечный файл, все глобальные символы, ранее стертые, восстанавливаются, если не был указан ключ /G.

Ключ включения имен модулей (/N). Если не использован ключ /N в первой строке команды, то библиотекарь не включает имена модулей в каталог. Редактор связей загружает модули из библиотек, основываясь на неопределенных глобальных символах, а не на именах модулей.

Если имена модулей не включаются в каталог, то столбец «MODULE» листинга каталога не будет заполнен до тех пор, пока модуль не потребует продолжения строки для печати всех глобальных символов. Знак (+) в столбце «MODULE» указывает на продолжение строки. Ключ /N используется для создания временной библиотеки, чтобы получить листинг каталога. Если библиотека не имеет в своем каталоге имен модулей, необходимо создать новую библиотеку с включением имен модулей.

Ключ включения имен P-секций (/P). Библиотекарь не включает имена программных секций в каталог, если не использован ключ /P в первой строке команды. Редактор связей не использует имена секций для загрузки программ из библиотек; включение имен может снизить производительность редактора связей.

Ключ замещения (/R). Ключ /R используется для замещения модулей в библиотечном файле. /R должен следовать за именами тех входных файлов, в которых содержатся модули для замены. Операция замещения заключается в том, что все модули, содержащиеся во входных файлах с ключом /R, заменяют существующие

модули с теми же именами, что и в указанном библиотечном файле.

Если прежний модуль не имеет того же имени, что и входной модуль, или, если указан ключ /R в библиотечном файле, то библиотекарь печатает сообщение об ошибке, и замена модулей в этом случае не происходит.

Ключ корректирования (/U). Операция корректирования выполняется при использовании ключа /U и совмещает две операции: замещения и вставки объектных модулей. Если объектные модули входного файла командной строки уже существуют в библиотечном файле, то при использовании ключа /U выполнится операция замещения, если подобных модулей нет, модули входного файла с ключом /U будут включены в библиотечный файл.

Ключ широкого формата (/W). Ключ /W включает широкую распечатку, если требуется файл листинга. Листинг широкого формата имеет шесть столбцов «GLOBAL» вместо трех в обычном листинге. Ключ /W полезен для распечатки каталога на устройстве печати или терминале, который имеет ширину 132 колонки.

Объединение библиотечных файлов

Можно объединить два и более библиотечных файлов под одним именем путем указания в командной строке всех объединяемых библиотечных файлов. Библиотекарь не стирает индивидуальные библиотечные файлы после объединения, если имя выходного файла не идентично одному из входных имен файлов.

Синтаксис команды:

FLIB = FINP1,FINP2,...,FINPN

где FLIB — библиотечный файл, который вмещает все объединенные файлы. Если под этим именем уже существует библиотечный файл, необходимо указать его во вводимой командной строке, FINP — объединяемые библиотечные файлы.

Сочетание ключей в командной строке

Можно запросить две или более библиотечных функций в одной командной строке, за исключением ключа /E. Библиотекарь выполняет действия (и выдает соответствующий запрос на подтверждение) в следующем порядке:

- 1) /C или //;
- 2) /D;
- 3) /G;
- 4) /U;
- 5) /R;
- 6) включение;
- 7) распечатка.

Ключ макро (M[:N])

Ключ /M[:N] создает файл макробιβлиотеки из входного файла, содержащего директивы макроассемблера. Аргумент N определяет область для размещения каталога имен макроопределений. Число N восьмеричное. Каждые 64 макрокоманды занимают один блок области каталога библиотеки; по умолчанию N=128.

Синтаксис команды:

FLIB = FINP/M[:N]

где FLIB — создаваемая макробιβлиотека, FINP — входной файл, содержащий макроопределения.

Ю.И. Мерзляков

Программное обеспечение процессора АП-32.

4. Консольная подсистема. Библиотекарь

Ответственный за выпуск С.Г.Попов

Работа поступила 28 февраля 1990 г.
Подписано в печать 5.03 1990 г. МН 02148
Формат бумаги 60×90 1/16 Объем 2,2 печ.л., 1,8 уч.-изд.л.
Тираж 250 экз. Бесплатно. Заказ № 29

Набрано в автоматизированной системе на базе фото-наборного автомата ФА1000 и ЭВМ «Электроника» и отпечатано на ротапинтере Института ядерной физики СО АН СССР,
Новосибирск, 630090, пр. академика Лаврентьева, 11.