

Стандартная система редукции астрономических данных MIDAS

Лекция I. Введение. Краткий обзор возможностей

А.Ю. Князев

Специальная астрофизическая обсерватория РАН, п.Нижний Архыз, 357147, Россия

Слушай первую премудрость, и большей не было в мире никогда со времен Адама... Запомни мои слова и думай над ними неотступно днем и ночью — и тогда ты постигнешь заключающуюся в них премудрость.

Леонид Соловьев. "Повесть о Ходже Насреддине".

Цель любого астрономического эксперимента — получение максимально достоверной информации. Процесс получения такой информации в наблюдательной астрономии формально можно разбить на следующие шаги:

1. Получение астрофизических данных (наблюдения);
2. Буферизация и архивизация полученных данных;
3. Обработка полученных данных;
4. Анализ и интерпретация полученных результатов.

Каждый шаг предполагает наличие и использование целого ряда различных средств и методов, составляющих технологию наблюдательного процесса: телескопов, их систем управления, астрономических приборов, систем управления этими приборами, систем сбора данных, систем буферизации и архивизации данных и т.д. Данные, проходя через все эти шаги, связывают их между собой и определяют технологию работы с экспериментальным материалом.

В последнее время наблюдательная астрономия практически имеет дело только с данными, получаемыми в цифровом виде, и это определяет ее зависимость от состояния и развития вычислительных средств и программного обеспечения. Каждый из вышеперечисленных шагов предполагает наличие определенного типа этих компонент.

Оптимизация параметров цепочки получения информации с точки зрения вычислительных средств и программного обеспечения является достаточно важной задачей, ибо после развития телескопов, астрономических приборов и астрономических приемников, именно она определяет временные затраты астронома на получение конечного результата его работы.

Специальная астрофизическая обсерватория как астрономический центр имеет все звенья цепочки работы с экспериментальными данными, и процесс получения информации связан с профессиональной деятельностью астрономов САО: разрабатываются новые приборы и методы наблюдений, планируются и реализуются наблюдательные программы, постоянно проводится обработка полученного материала. Как универсальный инструмент, 6-метровый телескоп всегда имел большой набор разнообразных методов и инструментов, а значит, и достаточно большое число систем сбора, систем редукции данных и проблем, связанных с созданием и развитием тех и других.

Скачкообразное развитие вычислительных средств в конце 80-х годов, появление большого количества новых типов компьютеров привело к осознанию необходимости выработки концепции развития цепочки

Сбор —> Архивизация —> Обработка —> Анализ ,

для обсерватории на ближайшее будущее и с учетом перспектив развития вычислительных и программных средств, методов и приборов. Такая концепция была сформулирована в 1990 году и состояла в следующем:

1. Сбор, обработка и архивизация данных представляют однородный, связанный процесс и должны осуществляться в одной среде. Требование многозадачности при наблюдениях и при обработке, естественность последующего развития, подразумевающего создание сетей и развитие моды удаленных наблюдений, а также тенденция развития мирового астрономического сообщества указывают на то, что этой средой должна являться операционная система UNIX.
2. Необходимо использование стандартной системы редукции астрономических данных (одной или нескольких), что позволит избежать проблем, связанных с ее развитием и поддержкой, и увеличит совместимость и переносимость программных продуктов, созданных внутри такой системы.
3. Необходима совместимость с мировым астрономическим сообществом по типам выходных данных.
4. Необходимо наличие архива наблюдательных данных.

В этом цикле лекций я коснусь только одного пункта этой концепции — стандартной системы редукции астрономических данных. Я почти ничего не буду рассказывать об операционной системе UNIX и о том, как в ней работать: для этого необходим еще больший цикл подобных лекций. Однако, уверен, что каждый из Вас, кто попробует работать с MIDAS (или с любой другой стандартной системой редукции астрономических данных), тут же поймет, что знать UNIX надо, ибо очень многое в процессе редукции приходится делать вне системы редукции (копировать файлы, переименовывать их, задавать недоуменные вопросы по электронной почте людям, знающим больше Вас на текущий момент обработки, и т.д. и т.п.). Но я могу сказать точно, о чем буду говорить. Любая стандартная система редукции — это безбрежный океан возможностей: начинающему в нем легко заблудиться и утонуть. Так вот, я расскажу о том, как начать работать в MIDAS. Я попытаюсь показать на наборе примеров и систематизации всего материала, как ориентироваться в MIDAS. Вместе с тем, я попытаюсь обозреть максимальное число возможностей в этой системе редукции, чтобы любой из Вас на основе

этих лекций мог предполагать, в какой части этого океана лежит тот островок, который Вас интересует. Я расскажу о том, что из себя представляет язык для написания командных файлов в MIDAS, а также опишу, как встраивать в него свои программы, уже написанные ранее на С или FORTRAN и каких правил при этом надо придерживаться. В конце я также коснусь вопроса об установке MIDAS на собственный компьютер. Я работаю с этой системой уже более 10 -ти лет, считаю себя экспертом в MIDAS не только потому, что устанавливал его на 5 компьютеров различных типов и 9 разновидностей UNIX, но и потому, что обрабатывал в нем совершенно различные типы данных. За все это время мне задавали гигантское количество разнообразных вопросов, и я видел наиболее часто встречающиеся ошибки начинающих (и не только) пользователей. Поэтому постоянно в тексте я буду выделять предупреждающие конструкции **Внимание** и **Замечание**.

ВНИМАНИЕ : *Это будет выглядеть так*

Замечание : *Это будет выглядеть так*

MIDAS команды в тексте будут выделяться так: MIDAS/COMMAND.

1. Стандартные системы обработки. MIDAS — почему именно он?

Под стандартной системой редукции всегда подразумевается некий набор программ, предназначенных для обработки астрономических данных и работающих в некоторой программе-мониторе, которая обеспечивает передачу различных данных между программами и обеспечивает связь с операционной системой.

Проблема обработки данных в астрономии стара, как и сама астрономия, и имеет, с моей точки зрения, два принципиально разных подхода:

Первый — построение специализированной системы, оптимальной для данного приемника (или набора данных, или человека...).

Второй — установка и использование любой стандартной системы обработки астрономических данных.

Оба подхода, с моей точки зрения, абсолютно равны и могут выбираться в зависимости от конкретных условий, обстоятельств и пристрастий.

Первый подход обеспечивает наиболее быстрое решение проблемы и максимальное использование возможностей компьютера. Его недостаток — ориентированность на данный тип компьютера и/или операционной системы, а также приемника. Смена любой из этих компонент очень часто приводит не просто к необходимости перетранслировать программы — проблема становится куда более серьезной: требуется существенная переработка программ для подстраивания их под это изменение (это — плата за очень короткое время создания системы). Другой недостаток — необходимость сопровождения системы (по последним данным, на это уходит 75% времени у человека, который ее создал), так как речь идет не о нескольких программах, которыми пользуется один человек, написавший их для себя. Речь идет о документированной, развивающейся системе обработки, в которой должны быть исправлены ошибки.

Второй подход решает часть из этих проблем, но создает новые. Его преимущество состоит в том, что поддержка системы осуществляется ее создателями, которые систему и развиваются. Система, как правило, содержит возможность обработки большого набора разнообразных данных, получаемых на различных приемниках, и, что является на мой взгляд самым главным, имеет общую концепцию развития с начала ее создания. Однако решение устанавливать стандартную систему обработки накладывает сразу же довольно жесткие ограничения на параметры используемых для этой цели компьютера и операционной системы, а также ... "увеличивает головную боль у того, кто устанавливает ее первый раз" и требует от этого человека достаточно большого времени для ее изучения.

Тем не менее, я лично (в силу различных причин и обстоятельств) придерживаюсь второго подхода. С моей точки зрения, гораздо проще изучить в деталях **ЛЮБУЮ** стандартную систему астрономических данных, а также механизмы встраивания в эту систему пользовательских программ (о чем, для MIDAS, я буду детально рассказывать в лекции 5 этого цикла) и после этого заниматься созданием дополнительного, необходимого только этому астрономическому прибору, приемнику или астроному, программного обеспечения, чем каждый раз пытаться делать практически все заново и, один-на-один сшибаясь в смертном бою с проблемой переносимости программ.

Однако не всегда все было так понятно и очевидно. Буквально десять лет назад, когда стало очевидно, что будущее компьютерного парка в России — это дешевые PC, еще не было ни одной стандартной системы редукции астрономических данных на этом самом PC (тогда и UNIX-то на PC встречался очень редко). Вот тогда в САО РАН по инициативе талантливого астронома Валентина Александровича Липовецкого, была создана рабочая группа из программистов и астрономов для переноса на PC одной из существующих стандартных систем обработки астрономических данных.

Для этого был сформулирован ряд требований по выбору оптимальной для САО РАН системы обработки астрономических данных:

1. Уже существующая возможность установки на максимальном количестве различных типов машин.
2. Наличие системы в исходных текстах, написанных на стандартных, широкоиспользуемых языках.
3. Наличие готовых программ и пакетов для обработки данных, имеющихся в САО.
4. Возможность встраивания новых программ и пакетов.
5. Возможность тесного и постоянного контакта с ее разработчиками.

Исходя из этих принципов, и была выбрана система MIDAS (разработанная и поддерживаемая Южноевропейской обсерваторией), обладающая на 1989 год рядом преимуществ относительно других известных систем. Работоспособность MIDAS-portable была проверена на многих типах компьютеров и в разных версиях OS, были пробы также и на персональных компьютерах (PS/2, OS AIX). Авторы считали своей главной задачей реализацию рабочей версии MIDAS в целом без существенных изменений с учетом ограниченных ресурсов персональных компьютеров, полностью сохраняя при этом концепцию переносимости.

Конец у этой сказки счастливый — уже в 1990 г. первая, перенесенная на РС, версия MIDAS заработала (хотя и не в полном объеме), а в 1992 году этот результат уже докладывался международному сообществу. С тех пор наша лаборатория (лаборатория "Структура"), а значит и САО РАН, в MIDAS и работает. При этом необходимо отметить, что стандартная система астрономических данных — IRAF (разработана и поддерживается в Национальной Оптической Астрономической Обсерватории, США) начала работать на РС с UNIX только с 1997 года!

Другими же известными стандартными системами редукции астрономических данных являются:

- AIPS — разработана в США для обработки радиоданных, имеет версию,ирующую на РС с UNIX;
- STARLINK — разработана в Англии. Вот-вот должна появиться на РС с UNIX;
- VISTA — разработана в США для обработки оптических данных. Мне известен только очень урезанный, сделанный, видимо, студентом в качестве дипломной работы, пакет PC-VISTA.
- IDL — коммерческий пакет с векторным языком программирования. Не специализирован для обработки именно астрономических данных, но имеет огромный набор дополнительных пакетов, написанных именно астрономами и именно для обработки астрономических данных.

Вообще говоря, я не отдаю никакого предпочтения любой из всех перечисленных стандартных систем редукции данных: каждая из них имеет свои как положительные, так и отрицательные стороны. Мой опыт показывает, что во многих ситуациях они дополнительны друг к другу: поэтому в последнее время я использую как MIDAS, так и IRAF при обработке. Однако, с моей точки зрения, есть несколько аргументов в пользу того, чтобы прочитать лекции именно по MIDAS:

(1) разработанный в Германии, MIDAS несет в себе структуированность подхода, что сильно облегчает систематизацию знаний о нем. В частности, это относится к системе **HELP**.

(2) человеку, изучающему новую (первую?) стандартную систему редукции, постоянно хочется задавать вопросы более знающему товарищу. Так вот, людей знающих MIDAS, очень хорошо и достаточно хорошо, в САО РАН существенно больше, чем людей, знающих, скажем, IRAF, IDL или AIPS.

(3) пункт (2) также относится к установке и настройке.

2. Что из себя представляет MIDAS сейчас

MIDAS (Munich Image Data Analysis System) — это система, используемая для анализа различного рода астрономических данных, созданная и поддерживаемая в ESO. Систему начала развивать с 1979 года Image Processing Group из ESO с целью создать "открытую систему обработки данных", в которую можно легко встраивать новые программы и целые пакеты для обработки данных, полученных на новых астрономических приборах. С 1982 года MIDAS используется для обработки астрономических данных в

ESO, а также свободно распространяется в других астрономических заведениях. С 1996 года система доступна любому человеку в текстах через INTERNET.

В настоящее время MIDAS работает под управлением операционных систем: VAX/VMS и UNIX и реализован на машинах различных типов: SUN, DEC, IBM, VAX, IBM, PCS, HP, Convex, Silicon Graphics и PC. Это возможно благодаря тому, что MIDAS распространяется в текстах, опирается на стандартные компиляторы (FORTRAN-77 и C) и на стандартные библиотеки. Последние или поставляются вместе с MIDAS, или присутствуют на большинстве компьютеров. В 1995 году было оценено, что на создание и отладку этой системы ушло примерно 300 человеко-лет (300 программистов трудилось один год, или один программист работал 300 лет — как Вам больше нравится). Тексты MIDAS, необходимые для инсталляции системы, занимают около 50 мегабайт дискового пространства, а полностью сгенерированная система занимает около 100 мегабайт. Система имеет три тома документации, поставляемой в виде PostScript и HTML файлов. Можно копировать уже сгенерированную систему для Вашего компьютера с Вашей версией UNIX, используя INTERNET адрес:

<http://www.eso.org/projects/esomidas/>

В настоящее время MIDAS содержит около 400 базовых команд, в нем реализовано большое количество различных пакетов (около 30) для обработки и анализа различных данных, полученных на астрономических приборах (что составляет еще дополнительное около 300 команд), а также содержит 8 GUI пакетов (графические пользовательские интерфейсы), позволяющие проводить обработку и анализ данных по технологии: Click-click-click-click... Более детально о составе пакетов в MIDAS я буду рассказывать в лекции 4 этого курса.

Если у пользователя появляются вопросы или он обнаруживает ошибку, то он немедленно может написать обычное электронное письмо по адресу midas@eso.org или воспользоваться для этого специализированной формой в GUI-интерфейсе CREATE/GUI help.

MIDAS стоит на 4-х основных "китах":

- Графическая библиотека AGL (Astronet Graphical Library) — разработана в Italian ASTRONET и используется для вывода всех графиков (не изображений) на стандартный набор внешних устройств. Поставляется с MIDAS.
- Математическая библиотека NAG (Numerical Algorithms Group) — используется в некоторых пакетах, в том числе и для приближения методами максимального правдоподобия. NAG — это коммерческий пакет, поставляемый для [C,F77,ADA,Pascal,Algol68] и содержащий около 2000 подпрограмм, используемых при математических расчетах. В настоящее время разработчики MIDAS практически полностью ушли от использования этого пакета из-за разработки собственных программ.
- Библиотека использования изображений (IDI-интерфейс) — написана для эксплуатации ее под управлением X Window, что является основным вариантом работы в UNIX. Поставляется с MIDAS. Если бы ее не было, вся одномерная графика и визуализация изображений не могли бы существовать.

- Motif — библиотека, используемая для создания графических пользовательских интерфейсов. В настоящее время имеется ее публично доступный собрат.

MIDAS состоит из двух основных программ:

- MIDAS-монитора — он вызывается при старте MIDAS-сессии. Является интерпретатором Ваших команд и осуществляет разбор и анализ командной строки, набранной пользователем, связь с операционной системой, обмен данными между различными программами и между внешними устройствами. А также пытается объяснить Вам Ваши ошибки.
- IDI-сервера — он вызывается при запуске первой программы, работающей с графикой или с вводом/выводом изображений на image-дисплей и отвечает за обмен данными между монитором и графическим и image дисплеями. Поскольку IDI-сервер работает в X Window стандарте, это автоматически позволяет использовать для визуализации данных и результатов удаленные сетевые мониторы, работающие в X Window.

2.1. Особенности системы MIDAS

MIDAS обладает следующими особенностями:

- Система приспособлена для интерактивной работы;
- MIDAS команды моделируют командный язык DEC, работающий на операционной системе VAX/VMS;
- Команды можно объединять, создавая исполняемые командные файлы;
- Командный язык MIDAS является развитым языком программирования, так как обладает широким набором возможностей для написания программ: операторами цикла, операторами переходов, локальными/глобальными переменными, возможностью вызывать подпрограммы;
- Наличием развитой системы подсказок, снабженной детальным описанием всех команд и квалифицированных.

MIDAS поддерживает большой набор структур данных: Изображения (Images), Таблицы (Tables), Описатели (Descriptors), Ключевые слова (Keywords), Каталоги (Catalogs), Fit-files, что обеспечивает широчайший спектр возможностей. Об этом я буду рассказывать в дальнейшем.

Система обеспечивает ввод-вывод данных в FITS-формате, что облегчает перенос данных. Система поддерживает работу с огромным набором внешних устройств для вывода на них графики и изображений.

2.2. Базовые возможности

Существующий MIDAS обладает базовым набором функций для работы со своими структурами данных:

1. Чтение в image-дисплей (на VAX/VMS это специальный видеодисплей (типа DeAnza IP 8500), а в X Window — специально заведенное окно) изображений, масок, таблиц раскраски (look-up tables — LUT), таблиц преобразования интенсивности (intensity transfer tables — ITT) и запись их из image-дисплея на диск.
2. Интерактивная модификация LUT, позволяющая выделить некоторые особенности на изображении.
3. Увеличение(уменьшение) изображений, загруженных в image-дисплей, интерактивно или с фиксированными величинами.
4. Просмотр изображений в монохромном или в псевдоцветном режиме.
5. Извлечение участков изображений интерактивно, пользуясь курсором или используя заранее выбранные координаты.
6. Рисование спектров, контуров, гистограмм, перспектив.
7. Вращение, переворачивание и другие преобразования изображений (например, линеаризация).
8. Выполнение арифметических операций и других обычных FORTRAN-функций над изображениями и таблицами.
9. Различная фильтрация и Фурье-преобразования изображений и таблиц.

3. Пособие для самых нетерпеливых

Существует категория людей, которым хочется и нравиться разбираться во всем новом самим. Так вот, специально для них сообщаю, что MIDAS вызывается командой **inmidas**, выход из него — команда **BYE**, а самая полезная и необходимая команда — **HELP**. Также добавлю, что абсолютно необходимо хоть какое-то знание английского языка. Вышеперечисленного достаточно, чтобы начать разбираться во всем самим, а не читать всю эту рутину дальше.

4. Что такое MIDAS-команда

Одно из базовых понятий для работе с MIDAS — MIDAS-команда. MIDAS-команда выглядит вот так:

COMMAND/QUALIFIER par1 ... par8 !Комментарий

Она состоит из команды (COMMAND) и квалификатора (QUALIFIER), разделенных прямым слэшем (/) и набором входных параметров (par1 ... par8), отделенных друг от друга пробелами. В одной команде не может быть более 8-ми параметров. Через пробел за последним параметром может следовать восклицательный знак, обозначающий начало комментария, а дальше, собственно, комментарий. А вот комментарий может быть любой.

ВНИМАНИЕ : Помните о пробеле между последним параметром и знаком комментария. Его отсутствие может привести к странностям при выполнении командных файлов, создание которых является естественным продолжением интерактивной моды работы в MIDAS.

Несколько странный вид команды, доставшийся MIDAS от системы IHAP и операционной системы VAX/VMS, является очень удобным, с моей точки зрения, ибо позволяет дополнительно систематизировать команды. По внутреннему MIDAS-стандарту принято, что команда обозначает некое действие, а квалификатор — то, с каким типом данных это действие выполняется. Например:

STATISTICS/IMAGE — расчет статистических характеристик изображения или его фрагмента;

STATISTICS/TABLE — расчет статистических характеристик колонки таблицы;

EDIT(TABLE — редактирование таблицы;

READ/KEYWORD — чтение содержания ключевого слова.

Существуют, конечно, и исключения, когда квалификатор специфицирует тип действия. Например:

FILTER/GAUSS — фильтрация с использованием GAUSS-функции;

FILTER/SMOOTH — фильтрация при помощи скользящего среднего (осреднение).

Исключение также составляют дополнительные команды из пакетов (контекстов — о них я подробнее буду рассказывать в четвертой лекции). В них команда обозначает действие, а квалификатор специфицирует, из какого пакета эта команда. Например:

PLOT/ECHELLE — рисование спектров в пакете ECHELLE-редукции;

PLOT/LONG — рисование спектров в пакете редукции двумерных спектров, полученных в mode "длинная щель".

Как я уже говорил, параметры разделены пробелом и могут быть числами (в общем случае — несколько чисел, разделенных запятыми) или именами изображений/таблиц/описателей/ключевых слов. Например:

STATISTICS/IMAGE ccdimage CURSOR — имя входного изображения и выбор моды работы с ним (курсор);

FILTER/GAUSS ccdininput ccdoutput 2,2 3.,1.,3.,1. — имена входного и выходного изображений, а также наборы числовых параметров фильтрации;

EDIT/TABLE mytab — имя входной таблицы;

READ/KEYWORD inputi — имя ключевого слова.

Параметры в MIDAS-команде позиционные. Это значит, что первым должен стоять первый параметр, вторым — второй и так далее. При этом многие параметры имеют значения "по умолчанию" (о чём будет сказано в третьей Лекции). При написании команды параметры, значения которых "по умолчанию" нас удовлетворяют, обозначаются просто знаком "?". Если все последующие параметры со значениями "по умолчанию" нас удовлетворяют, они просто не пишутся. Например:

STATISTICS/IMAGE ? CURSOR — статистика текущего (визуализированного в image-дисплее) изображения. Область выбирается курсором;

FILTER/GAUSS ccdinput ccdoutput — имена входного и выходного изображений. Параметры фильтрации — "по умолчанию".

Однако есть способы написания параметров в произвольном порядке. Для этого пишется выражение Рномер_параметра=значение. При такой записи параметры могут следовать в произвольном порядке. Например:

STATISTICS/IMAGE P2=CURSOR — статистика текущего (визуализированного в image-дисплее) изображения. Область выбирается курсором;

FILTER/GAUSS P2=ccdoutput P1=ccdinput — имена входного и выходного изображений. Параметры фильтрации — "по умолчанию".

Большинство команд имеют квалификатор, но некоторые — нет. Их очень мало и их надо знать "в лицо":

BYE — команда "прощания" с MIDAS (выход из сессии обработки);

HELP — самая простая команда запроса о помощи у системы;

RUN — исполнение монитором MIDAS внешней программы, написанной на С или FORTRAN.

Написание команд и квалификаторов может быть укорочено до некоего минимума, когда монитор еще различает их между собой. Например:

STATISTICS/IMAGE сокращается до **STAT/IMA** и даже до **ST/I**;

FILTER/GAUSS сокращается до **FILT/GAUSS** и даже до **FI/G**;

EDIT/TABLE сокращается до **EDIT/TAB** и даже до **ED/T**;

READ/KEYWORD сокращается до **READ/KEY** и даже до **R/K**.

ВНИМАНИЕ : Учитите, что этот минимум зависит от версии MIDAS и от загруженных пакетов (появились новые команды). Я настоятельно не рекомендую укорачивать команды и квалификаторы при написании командных файлов. Зв это можно поплатиться тем, что они перестают работать в следующих версиях системы (она развивается и появляются новые команды).

Советую использовать клавиши *Tab* при вводе командной строки. После нажатия этой клавиши монитор попытается вставить всю команду целиком, если Вы ввели уже минимум, позволяющий монитору сделать однозначный выбор.

5. Сага о формате FITS и основные типы данных системы MIDAS

Когда-то очень-очень давно (лет двадцать-тридцать назад), когда началась вторая волна компьютеризации в астрономии, проблема переносимости данных предстала перед астрономами-наблюдателями во всей своей красе: данные стали такими большими по объему, что их уже стало невозможно передать в виде тетрадки, записанной от руки, распечатки или колоды перфокарт. А компьютеров стало много и очень разных с сильно отличающимся матобеспечением. Поэтому астроном **A**, передавая астроному **B** полученные им для него данные (в те времена это происходило, в основном, при помощи магнитных лент), должен был, во-первых, очень детально описать в прилагающемся письме, каким образом и что записано на ленте (сколько байт, порядок бит, число записей), а во-вторых, детально описать сам процесс наблюдения — где, что и как. Вот тогда-то и был выработан стандарт формата — FITS (Flexible Image Transport System). Идею его можно выразить несколькими словами — это самоописанный тип данных. Сначала идет заголовок, в котором в соответствии с жесткими правилами описано, какие данные идут за ним: их размерность, число точек по каждой из осей, когда и кем данные были получены и так далее и тому подобное. Считается стандартом, что результаты наблюдений автоматически получаются в формате FITS. Все стандартные системы редукции могут читать/записывать файлы данных в стандарте FITS. Однако при работе практически все стандартные системы редукции используют не FITS, а некоторые свои специфические внутренние форматы. Это связано как с историей самой системы редукции, так и с ее основными типами данных.

Вообще типы данных, используемые системой редукции, имеют огромное значение. Если Вы разработчик, то внимательно проработайте этот вопрос, и получится красивое, стройное дерево системы редукции, к которому совершенно естественным образом пристает все то новое, что появляется в астрономии. Выберите неправильный стандарт или недостаточное количество разных типов данных, и уже вскоре Вы почувствуете, как ограничены в своих возможностях в рамках созданного матобеспечения. Например, Вы как разработчик забыли о таблицах — а ведь многочисленные конечные результаты обработки — это именно таблицы! Или, если Вы ограничитесь двумерными изображениями, то немедленно обнаружите, что многие астрономические приемники или системы сбора в настоящее время выдают трехмерные массивы!

Так вот, с моей точки зрения, MIDAS был разработан очень грамотными астрономами и программистами. Основные типы данных, используемые внутри MIDAS:

- **Images (Frames)** — массивы однородных данных (размерностью до 3). Частный случай изображений — это битовые или байтовые маски, которые могут быть использованы для выбора интересующей области в изображениях. Файлы стандартно имеют расширения: ".bdf".
- **Tables** — содержат данные в строках и колонках, и эти данные могут не иметь одинакового физического значения (неоднородные). Двумерные (трехмерные) массивы, организованные в колонки и строки, хранятся в виде бинарных файлов. Файлы стандартно имеют расширения: ".tbl".
- **Descriptors** — содержат информацию, которая физически ассоциируется с изображением, таблицей и fit-файлом (описывает их содержимое). Например, это имя

или число пикселей на каждой оси и т.д. Набор дескрипторов встроен в них в виде отдельных "шапок".

- **Keywords** — это глобальные переменные MIDAS-монитора, используемые для связи между различными MIDAS программами.
- **Catalogues** — используются для работы с группами однородных файлов данных: таблицами, изображениями и ASCII-файлами. Это просто ASCII-файлы, содержащие списки таблиц, изображений или других ASCII-файлов. Имеют расширение ".cat".
- **Fit-files** — это файлы, используемые в FITTING-программах. Содержат описание математической функции и начальных значений параметров. Имеют расширение ".fit".

ВНИМАНИЕ : *Внутренние типы данных в MIDAS являются переносимыми только в рамках одной и той же операционной системы и сильно связаны на hardware. Кроме того, существует несовместимость данных между некоторыми версиями MIDAS. Для портабельности бинарные данные в MIDAS необходимо преобразовывать в FITS или ASCII форматы.*

Замечание : Хочу поделиться опытом: внутренние MIDAS-данные совместимы между PC-компьютерами и Alpha-станциями, и несовместимы между PC-компьютерами и компьютерами фирмы SUN.

6. Первый шаг: старт и выход

Итак, вернемся к тому, что я уже коротко описывал ранее. Если на компьютере установлен MIDAS, то Вы можете вызвать его, просто набрав команду **inmidas**. Эту работу в конкретном MIDAS я буду называть MIDAS-сессией или просто сессией. Как только Вы почувствуете, что Вы хотите выйти из сессии, наберите команду **BYE**. Вернувшись к неоконченному после заслуженного отдыха, наберите команду **gomidas**, и Вы окажетесь в том месте, на котором остановились.

ВНИМАНИЕ : *Если Вы выполнили команду **inmidas** — стартовали MIDAS-сессию, то ни в коем случае не исполняйте эту команду параллельно второй раз — ничего хорошего не получится. Чтобы научиться работать в нескольких MIDAS одновременно — смотрите лекцию 2 этого курса.*

Уже после первого вызова команды **inmidas** Вы обнаружите в Вашей домашней директории новую, имеющую стандартное название **/midwork**. Не стирайте ее — она все равно будет создаваться каждый раз, когда Вы будете вызывать MIDAS. В ней MIDAS создает различные служебные файлы для текущей сессии.

Замечание : Запись **/midwork** стандартно для UNIX значит, что данная директория находится в Вашей домашней, которая иногда также обозначается как **`\${HOME}`**. В этом обозначении высказанные выше записалось бы в виде **`\${HOME}/midwork`**.

Эта его особенность и определяет тактику работы в MIDAS: пользователь в любом, ему удобном месте создает новую директорию, заходит в нее копирует туда файлы данных и стартует MIDAS-сессию. Для обработки других данных он может создать другую директорию в другом месте (главное не запутаться!). В директории же `/midwork` пользователь держит написанные им командные файлы и программы, которые он использует при обработке данных, находящихся в разных директориях.

Рассмотрим теперь в качестве примера некую MIDAS-сессию, чтобы наглядно показать, насколько логически простыми являются основные команды системы MIDAS. Ее может повторить каждый при правильно и полностью установленном MIDAS.

ВНИМАНИЕ : *По умолчанию считается, что Вы работаете в X Window.*

Замечание : Знак "`>`" в приведенном ниже примере выдает MIDAS-монитор в качестве приглашения к работе.

`inmidas` — старт MIDAS-сессии.

`> -COPY MID_TEST:sombrero.bdf testima.bdf` — это самая непонятная строка во всем примере, обозначающая копирование изображения галактики Sombrero в MIDAS-формате из некоторой стандартной директории в текущую. Скопированное изображение будет иметь имя `testima.bdf`.

`> STAT/IMAGE testima` — считаем статистику по всему изображению.

`> CREATE/DISPLAY` — создаем image-дисплей.

`> LOAD/IMAGE testima` — визуализируем изображение в image-дисплее.

`> DISPLAY/LUT` — показываем текущую LUT-таблицу.

`> LOAD/LUT rainbow` — "загружаем" в дисплей новую LUT-таблицу.

`> GET/CURSOR` — Создаем курсор в image-дисплее.

`> EXTRACT/IMAGE pice = testima[100,120:200,220]` — извлекаем часть изображения `testima.bdf` и переносим в другое изображение с именем `pice.bdf`.

`> CREATE/GPGRAPH` — создаем графический дисплей.

`> PLOT/ROW testima 100` — рисуем строку номер 100 из изображения.

`> PLOT/COLUMN testima 100` — рисуем колонку номер 100 из изображения.

`> $ rm testima.bdf pice.bdf` — удаляем изображения с диска.

`> BYE` — завершаем MIDAS-сессию.

Таблица 1. Возможности команды HELP

Команда	Ее назначение	Пример
HELP	Краткий обзор всех команд	
pattern?	Список всех команд, начинающихся с pattern	RE?
HELP command	Список всех команд, содержащих command, с кратким описанием назначения и параметров	help read
HELP command/qualif comnd/qualif ??	Подробный help по команде Краткое описание назначения и параметров команды	HELP READ/KEY READ/KEY ??
HELP/QUALIF qualif	Список всех команд, имеющих квалификатор qualif	HELP/QUALIF TABLE
HELP/SUBJECT	Полный список тем, по которым есть информация	
HELP/SUBJECT subject	Подробная информация по данной теме	HELP/SUBJ image
HELP/CL	Краткий обзор всех команд внутреннего языка MIDAS-монитора	
HELP/CL comnd	подробный help по команде языка MIDAS-монитора	HELP/CL IF
HELP/KEY keyword	описание назначения и типа ключевого слова	HELP/KEY inputi
HELP [Topic]	Детальное описание всех новостей и новых возможностей	HELP [News]
HELP/APPLIC	Полный список дополнительных команд	
HELP/APPLIC APPLIC	Подробное описание дополнительной команды applic	HELP/APPLIC autocuts

7. On-Line Help

Как я уже указывал ранее, в MIDAS хорошо развита система HELP. Она позволяет делать обзор всех команд или выделять все команды, имеющие данный квалификатор. Я настоятельно рекомендую начать самостоятельное изучение MIDAS именно с этой команды и полностью освоить все ее возможности. После освоения HELP регулярно используйте ее. Некоторые ее возможности приведены в Таблице 1 и проиллюстрированы примерами:

ВНИМАНИЕ : Внимательно читайте подробный help по команде. Параметры в квадратных скобках "[parametr]" являются необязательными.

Очень удобным, с моей точки зрения, является GUI-интерфейс Help (вызов: CREATE/GUI HELP).

7.1. Разделы Tutorials

MIDAS обладает не только развитой системой help, но и многочисленным набором примеров. Это команды TUTORIAL. Они служат для демонстрации работы некоторых базовых команд, работы с различными типами данных, работы с различными контекстами и т.д. Напишите команду HELP TUTORIAL и посмотрите, какие команды имеются. Исполните их и посмотрите на результат.

Замечание : Есть еще один замечательный способ познакомиться со многими возможностями MIDAS. Дело в том, что после инсталляции системы, создается специальный пакет, тестирующий большую часть ее возможностей. По мере выполнения своих тестов, исполняемые команды появляются на терминале, а в графический и image-дисплеи загружаются результаты тестирования. Если пакет прервал свою работу — значит что-то с MIDAS на Вашем компьютере не в порядке — зовите системщика или начинайте разбираться сами. Итак, чтобы выполнить этот пакет, сначала наберите команду @ vericopy, а после того как она отработает, — команду @@ veriall и ... сидите и смотрите.

ВНИМАНИЕ : Самая распространенная ошибка во время исполнения этого пакета — использование терминала с разрешением меньшим, чем 1024×768 точек. В один прекрасный момент исполнение прервется только потому, что графический и image-дисплеи не умещаются на Вашем экране. Системщик не поможет. Надо переделывать программы.

И это конец первой сказки о MIDAS.

Список литературы

Князев А.Ю., Диссертация, 1997, с.1–248 (http://precise.sao.ru/Laboratory/Dis_agn/index.html)
 Kniazev A.Y., Shergin V.S., Lipovetsky V.A. (1992) MIDAS on small computers: reduction of low order echelle spectra. 4-th ESO/ST-ECF Data Analysis Workshop: 169–176
 MIDAS Users Guide, 1995a, Volume A
 MIDAS Users Guide, 1995b, Volume B

Но эта премудрость — ничто в сравнении со второй премудростью, которую я тебе поведаю...
Леонид Соловьев. "Повесть о Ходже Насреддине"

Стандартная система редукции астрономических данных MIDAS

Лекция II. MIDAS-монитор. Ввод-вывод данных.

Взаимодействие с операционной системой

А.Ю.Князев

Специальная астрофизическая обсерватория РАН, п.Нижний Архыз, 357147, Россия

Открой свои уши и внимай, ибо вторая премудрость... Постарайся же, о сын мой, понять эту премудрость и радуйся, что получил ее бесплатно.
Леонид Соловьев. "Повесть о Ходже Насреддине".

В предыдущей лекции мы обсудили общие сведения о стандартной системе астрономических данных MIDAS. Настало время более детально рассмотреть различные возможности и особенности этой системы. И начну я с нескольких способов запуска MIDAS-сессии.

1. Особенности запуска MIDAS-сессии

Я уже рассказывал, что команда запуска сессии называется **inmidas**. Очень полезно знать хотя бы некоторые ее возможности. Для ознакомления достаточно выполнить одну из двух команд: **man inmidas**, если MIDAS установлен полностью, то после этой UNIX-команды пользователь получает детальное описание сервиса или, если установка выполнена "спустя рукава", необходимо выполнить команду **inmidas -help** и получить следующую информацию на экране терминала:

```
Usage: inmidas [unit] [-h midashome] [-r midvers] [-d display] [-p/-P/-nop]
          [-m mid_work] [-noh] [-j "midas-command-line"] [-help]
Options:
-h midashome      ...
-r midvers        ...
-d display         ...
-p/-P/-nop        ...
unit              ...
-m mid_work       ...
-noh              ...
-j "midas_command_line" ...
-help             ...
```

Я коротко поясню эти опции:

unit Каждая MIDAS-сессия ассоциируется с двумя цифрами или буквами. Если это цифры (диапазон от 00 до 99), то работа идет в X Window, если буквы (диапазон от ха до zz), то Вы запускаете MIDAS на ASCII-терминале (некий неграфический терминал). "По умолчанию" (если ничего не указывать) используется 00. Эти две цифры в сочетании с опцией **-р** позволяют работать в нескольких MIDAS одновременно.

Пример :

inmidas 00 или **inmidas 10**

inmidas xx (вы предполагаете работу без графики)

-р/-Р/-nop При запуске команды **inmidas** или **inmidas 00** в директории **/midwork** стираются ВСЕ временные файлы, относящиеся к любым другим MIDAS-сессиям. Чтобы это не произошло, необходимо указать **-р** или **-Р**, что просто приведет к появлению новых файлов для новой MIDAS-сессии (конечно, их номера НЕ ДОЛЖНЫ совпадать с уже запущенными сессиями). Это включение так называемого PARALLEL (параллельного) режима. Опция **-nop** — это явное запрещение такого режима запуска.

Пример :

inmidas 00 — запуск первой сессии. Все стирается.

inmidas 10 -р — запуск следующей сессии с сохранением предыдущей.

-r midvers На одном компьютере могут одновременно существовать несколько версий MIDAS. Разные пользователи могут предпочитать разные версии (я сам предпо- читаю версию 96NOV всем остальным), поскольку есть неисправленные ошибки, пристрастия и так далее. "По умолчанию" при вызове **inmidas** вызывается некая (обычно последняя) версия MIDAS, но любой пользователь при помощи этой опции может загрузить и любую другую версию, существующую на данном компьютере.

Замечание : Новая версия MIDAS выпускается один раз в году и называется соответственно 95NOV, ... 99NOV (NOV от NOVEMBER, поскольку до 1994 года выпускались две версии в год, которые назывались 93MAY и 93NOV соответственно)

Пример :

inmidas 00 — запуск первой сессии. Все стирается.

inmidas 10 -р — запуск следующей сессии. Предыдущая сохраняется. Release of MIDAS. Sets variable MIDVERS to midvers and it must exist as a subdirectory under MIDASHOME.

Замечание : В случае попытки одновременного запуска двух сессий с одинаковыми номерами (допустим сессии с номером 10), MIDAS предупредит Вас об этом:

```
Unit 10 is locked by another MIDAS session.
To unlock just continue.
Do you want to continue [yn]? (n):
```

Внимательно читайте все сообщения. Если Вы не хотите проблем, нажмите "n" в этом случае. Однако бывает ситуация, когда надо сказать "у": Вы работали в этой сессии, и вдруг все "рухнуло" — задумавшийся системщик нажал кнопку "Reset". После перезагрузки системы, при попытке зайти и поработать в MIDAS-сессии под номером 10, Вам также выдастся подобное сообщение.

-h midashome Бывают "странные" администраторы, которые помещают разные версии MIDAS в разные места на диске. Упрямый пользователь может найти эти версии и все-таки запустить их, используя еще и эту опцию.

-m mid_work Поменять можно многое — в том числе и назначить новую директорию для временных файлов вместо стандартной `/midwork`. Разные бывают ситуации в жизни обработчика данных.

-d display Поскольку работа (в основном) происходит в среде X Window, то можно работать на одном терминале, а графический вывод переназначить на другой терминал (предположим терминал Вашего лучшего друга Васи — чтобы ему жизнь медом не казалась). Легко можно придумать и другую ситуацию — Вы сидите и работаете в MIDAS, в X Window, за маленьким (14") терминалом, а вся графика и изображения загружаются на большой (17") терминал, стоящий рядом.

Замечание : Интересующихся отсылаю к любому пособию по UNIX, чтобы узнать детали. А неинтересующимся могу сообщить, что им все равно придется разобраться, как разрешать/запрещать/переназначать вывод на свой/другой терминал в среде X Window, а иначе они рискуют оказаться в роли лучшего друга Васи у интересующихся.

Пример :

inmidas 00 -d serv.sao.ru:0.0 — запуск сессии с явным обозначением имени терминала (учтите, что для успешной работы этого примера необходимо еще, чтобы Вам было разрешено выводить данные на этот терминал).

-j "midas_command_line" Выполнение команды "midas_command_line" как первой команды Вашей MIDAS-сессии.

Очень важный ключ. Основной режим работы в MIDAS-мониторе — интерактивный. А что делать, если Вам надо профильтровать изображение (нет — сто изображений!), и время работы программы над одним — сутки? А сзади у Вас очередь из пяти человек, которым тоже страшно хочется поработать в X Window. Что делать? Отстреливаться и не пускать? И так в течение ста дней. Не спешите! Для этого есть опция **-j**. Вместе с опцией **-noh** и механизмом выполнения отложенных заданий, существующем в любом UNIX, Вы можете устроить такое, что

этим пяти (а вместе с ними и всем остальным пользователям данного компьютера-сервера) просто придется дожидаться, когда Ваши программы закончат работать. Как это сделать? Дождитесь пятой лекции. Только учтите, что во всем хороша мера — а не то могут и лицо попортить.

-noh Указание: не очищать терминал при старте и не выводить никаких дополнительных сообщений.

-help После использования этого ключа выдается некий английский текст, коротко описывающий то, что я сейчас объяснил.

Итак, перефразируя слова известного героя кинофильма, "я коротенько, минуток эдак за 45", изложил то, какими способами можно MIDAS вызывать. Полезно сделать еще одно замечание. При старте MIDAS-сессии 00 в директории **/midwork** создаются два файла FORGR00.LOG и FORGR00.KEY. Следует знать, что файл FORGR00.LOG содержит весь протокол Вашей работы с MIDAS в этой сессии. Скопируйте его в файл с другим именем после окончания Вашей работы, и у Вас навсегда останется нужная информация (Ваши команды и MIDAS-ответы).

Замечание : Советую заходить в MIDAS всегда в параллельном режиме и под разными номерами (*MIDAS units*). Это поможет избежать проблем в случае, если несколько человек одновременно работают в MIDAS на одной и той же машине под одним и тем же пользователем.

2. Работа в MIDAS-мониторе

Так как основная мода работы с MIDAS интерактивная, то одно из самых важных умений — это грамотное управление MIDAS-монитором. Кратко суммирую то, что с моей точки зрения, должен знать каждый более-менее грамотный пользователь:

- MIDAS-монитор не различает большие и малые буквы: ему все-равно, напишите ли Вы **stat/ima**, **StAt/iMa** или **STAT/IMA**. В своих лекциях я пишу все команды большими буквами только ради красоты и однородности подачи материала.

ВНИМАНИЕ : *Прошу помнить, что имя команды и имя изображения — это разные вещи. Команду обрабатывает MIDAS-монитор, а имя изображения он отдает операционной системе. Так, если у Вас есть изображение на диске с именем **qubodub.bdf**, то написав **READ/DESCRIPTOR QUBODUB**, Вы получите ошибку — на диске нет изображения **QUBODUB.bdf***

- Максимальная строка ввода равна 256 символов.

- Символ " в конце строки обозначает, что ввод не окончен, а будет продолжаться со следующей строки.

Пример :

```
> EXTRACT/IMA pice = testima[100,120:200,220] — команда извлечения части из изображения testima.bdf
> EXTRACT/IMA -
> pice = testima-
> [100,120:200,220] — та же команда, но записанная в режиме перехода на новую строку.
```

- Можно записывать несколько команд в одной строке. Разделителем является символ ";".

Пример :

```
> DISPLAY/LUT ; LOAD/LUT rainbow
```

- Комментарием является символ "!"

Пример :

```
LOAD/LUT rainbow ! это коментарий
```

- Иногда текстовый параметр содержит пробелы. Можно указать монитору, что это один параметр — для этого его надо заключить в апострофы.

Пример :

```
> WRITE/OUT "I know it" — вывести на экран эту текстовую строку.
```

- Существуют различные способы задания MIDAS-параметров, которые в общем случае являются позиционными. Эти способы я приводил в лекции 1. Необходимо отметить следующее: кроме имен P1...P8 каждому параметру можно присвоить и собственное имя, которым и пользоваться далее для вызова данного параметра. Эти имена придумываются авторами, командных файлов, а авторы знают о существовании команды CROSSREF. Имена приводятся в "help" по данной команде (посмотрите, например, внимательно HELP STAT/IMA).

Пример :

```
> STAT/IMA testima [<,<:>,>] ? ? ? test P
> STAT/IMA testima P6=test P7=P — то же
> STAT/IMA FRAME=testima OUTTAB=test PL0T=P — то же
```

- Прервать выполнение любой команды можно, нажав одновременно клавиши Ctrl и C (обозначается как <Ctrl/C>). Правда, в некоторых версиях MIDAS Вы можете

при этом также прервать и всю MIDAS-сессию. Но что только не случается в нашей жизни.

- Последние 15 команд остаются всегда в командном буфере, который вызывается простым нажатием клавиши <Enter>. Размер буфера меняется командой SET/BUFFER.
- Любую уже исполненную команду, находящуюся в этом буфере, можно перезаписывать, пользуясь ее условным номером в буфере, написанном слева.

Пример :

> 3;4 — вызвать для исполнения команды номер 3 и 4 из буфера.

То же самое можно сделать, написав ":" и указав образец для поиска в буфере.

Пример :

> :WRITE — вызвать на исполнение первую команду из буфера, начинающуюся с WRITE.

ВНИМАНИЕ : Здесь MIDAS прозревает и начинает различать большие и малые буквы.

- Любую уже исполненную команду, находящуюся в командном буфере, можно перезаписывать для редактирования, пользуясь ее условным номером в буфере, написанном слева, добавив к нему ":".

Пример :

> 10. — вызвать на редактирование 10-ую строку из буфера.

То же можно сделать, написав ":@" и указав образец для поиска в буфере.

Пример :

> :.WRITE — вызвать на редактирование первую команду из буфера, начинающуюся с WRITE. > .:WRITE — то же.

- Параллельно существует и другой буфер команд, просматривать который можно, пользуясь стрелками ↑ и ↓ на клавиатуре. При вызове команды на исполнение из этого буфера, первый буфер будет тоже меняться.
- Любая команда операционной системы может быть выполнена из-под монитора с использованием символа "\$".

Пример :

> \$ ls — список всех файлов в текущей директории.

- Монитор всегда помнит о последнем значении каждого параметра. Оно обозначается как "." Этим можно пользоваться.

Пример :

```
> SHOW/TABLE mytable — показать заголовок таблицы mytable.tbl
> EDIT/TABLE . — исполнится команда EDIT/TABLE mytable.
> . yourtable — исполнится команда EDIT/TABLE yourtable
```

ВНИМАНИЕ : Из-за этой особенности MIDAS-монитора я не раз видел, как пользователи пытались нанести увечья компьютеру себе или окружающим, пытаясь выполнить в мониторе команду копирования данных в другой файл и забыв только что сказанное мной. Скажем, бедолага пишет:

```
> SHOW/TABLE mytable.tbl
> $cp ..//testima.bdf . — это абсолютно правильная команда в UNIX,
обозначающая копирование файла testima.bdf из верхней директории в текущую. Но выполнение ее в MIDAS-мониторе приводит к ужасающим последствиям! Пользователь перезаписывает файл mytable.tbl, в котором у него хранилась, несомненно, единственная копия всех последних двухнедельных расчетов, быстро убеждается в том, что последствия действительны ужасающи и впадает в прострацию/тоску/гнев (в зависимости от темперамента). Чтобы этого не произошло с Вами, приучитесь писать подобные команды так: $cp ..//testima.bdf ./ — это абсолютно верная запись как для MIDAS, так и для UNIX.
```

3. Внешние устройства

Доступное количество внешних устройств и их определение в MIDAS меняется в зависимости от того, какую версию Вы используете. К внешним устройствам относятся:

Магнитофоны. При правильной установке MIDAS в сети выбирается один компьютер, к которому подсоединяются все внешние устройства. На нем устанавливается и настраивается MIDAS tape-сервер, к которому может обратиться любая сессия с любого компьютера для записи/чтения данных с ленты.

Терминалы. Поскольку работа одновременно ведется с текстовым, графическими и image-дисплеями, приняты следующие обозначения:

- T[erminal] — текстовое окно;
- G,n — окно для вывода графики. Одновременно можно работать с десятью такими окнами — n принимает значения [0-9];
- D,n — image-дисплей. Одновременно можно работать с десятью такими окнами — n принимает значения [0-9];
- D[ISPLAY] — синоним D,0;
- POSTSCRIPT — файл в формате PostScript;

- **NULL** — мусорная корзина.

Принтеры. — MIDAS настраивается таким образом, что одновременно можно работать как с принтером Вашего компьютера, так и с сетевыми принтерами: матричными, лазерными, цветными и т.д. Существуют стандартные имена, а также локальные, настраиваемые Вами системщиками. Вся информация о локальных именах настроенных принтеров должна выдаваться по команде **HELP [Printers]** (это приходится "вынимать" из системщиков путем длительного и упорного общения с ними). К стандартным именам относятся

LPRINT — линейный(матричный) принтер;

LASER — лазерный принтер;

FILE — вывод в файл на диск.

Плоттеры. Можно подсоединять и плоттеры, хоть это и устарело. Стандартное имя — **PENPLOT**.

Переназначение вывода на другое внешнее устройство делается семейством команд **ASSIGN: ASSIGN/DEFAULT, ASSIGN/DISPLAY, ASSIGN/GRAFICS, ASSIGN/INPUT, ASSIGN/PRINT**

4. Ввод-вывод данных в MIDAS

Процесс обработки данных всегда начинается с ввода данных в MIDAS, а заканчивается выводом данных или результатов обработки. Коротко рассмотрим эти вопросы. Для получения более детальной информации по данному разделу смотрите команды **HELP [DataInput], HELP [TapeDevices], HELP [Printers], HELP [ImageDisplay]**.

4.1. Ввод

Данные в MIDAS могут быть введены из

FITS-файлов. Существуют как базовый FITS-формат, так и его расширения. Полезно знать, в какие внутренние структуры данных преобразуются эти форматы.

Базовый FITS	→	MIDAS-image
FITS-tables	→	MIDAS-table
BINTABLE	→	MIDAS-table
FITS random groups	→	MIDAS-image + MIDAS-table

Команды для чтения FITS-файлов и преобразования их во внутренний формат: **INDISK/TAPE** — чтение файлов с ленты или одиночных файлов с диска.

INDISK/FITS — чтение как одиночных файлов с диска, так и списков.

Пример :

(Предположим, что работаем с изображениями)

INTAPE/FITS 1 st star01.fits — чтение одного файла. Входное имя star01.fits, а выходное будет st0001.bdf

INDISK/FITS star01.fits star01 — чтение одного файла. Даны входное и выходное имена.

INDISK/FITS in.cat output.cat — чтение списка изображений. Входные имена записаны в файле in.cat, а выходные — в файле output.cat в формате, содержащем одно имя в строке.

INTAPE/FITS 1,5-7 st /dev/nrmt1 — чтение 1-го, 5-го, 6-го, 7-го изображений с ленты /dev/nrmt1. Выходные изображения будут именоваться st0001.bdf ...

RESTORE/NAME st0001 — восстановление старого имени файла (имени, которое было до того, как файл записали на ленту).

ВНИМАНИЕ : Помните, что имена списков для команды **INDISK/FITS** должны **ОБЯЗАТЕЛЬНО** заканчиваться на ".cat".

Замечание : Начиная с версии *MIDAS 94NOV*, можно работать с компрессированными данными. Если они есть, то не раскомпрессируйте их, а давайте полные имена. Например:

INDISK/FITS star01.mt.gz star01. Но имена должны заканчиваться стандартно ".Z, .z, .gz". Предполагается компрессия стандартными программами **compress** и **gzip**. Однако системщик может настроить на любые имена и практически любые компрессоры (например, **bzip2**) — "трясите" их (системщиков).

ASCII-файлов. В зависимости, от того какой внутренний формат Вы хотите получить, используйте разные команды:

MIDAS-изображение — команда **CREATE/IMAGE**:

CREATE/IMAGE data 1,512 1.,1. ASCII_FILE

MIDAS-таблица — команда **CREATE/TABLE**:

CREATE/TABLE catalog 20 1000 data.dat data(fmt

Ключевое слово — команды **OPEN/FILE** и **READ/FILE**:

OPEN/FILE data.dat read fctrl

READ/FILE fctrl(1) charbuf 20

TIFF-изображений. Это потенциальная возможность. Она нуждается в дополнительной настройке со стороны Вашего системщика, после чего можно использовать команды @a **intiff** и @a **sc anima**.

5. Вывод

Данные из MIDAS могут быть выведены в следующие форматы (я не учитываю прямого вывода на принтер командами **PRINT**):

FITS-файл Внутренние структуры данных MIDAS преобразуются в следующие FITS-форматы (основной и расширения):

MIDAS-image → Базовый FITS
 MIDAS-table → FITS-tables

Команды для записи FITS-файлов:

OUTTAPE/FITS — запись файлов на ленту или файлов на диск.

OUTDISK/FITS — запись файлов на диск по списку.

Пример :

(Предположим, что работаем с изображениями)

OUTTAPE/FITS testima.bdf testima.mt — запись одного файла на диск.

CREATE/ICAT icat t*.bdf — создание каталога изображений

OUTTAPE/FITS icat /dev/nrmt1 ASN — запись всех изображений из каталога на ленту

OUTDISK/FITS in.cat out.cat — запись файлов на диск по списку.

ASCII-файл Используются наборы команд ASSIGN и PRINT.

Пример :

ASSIGN/PRINT file data.tmp — назначаем имя файла вывода.

PRINT/TABLE catalogue #1,:Name,#10 — Печатаем в файл несколько колонок таблицы, явно их указывая.

PRINT/IMAGE testima <,@20,10 — распечатываем участок изображения.

PRINT/KEYWORD outputr,outputi — распечатываем содержание переменных.

Можно пользоваться и командами READ. Распечатывая интересующие Вас значения данных на экран, Вы всегда потом можете выделить этот кусок вывода из файла протокола (смотрите начало этой лекции).

Можно использовать возможность перенаправления ввода-вывода для записи интересующей Вас информации с экрана в файл (смотрите более подробно раздел о взаимодействии с операционной системой).

Если Вам нравятся трудности, то Вы, несомненно, захотите освоить возможность вывода с использованием команд OPEN/FILE и WRITE/FILE. Овладение этими командами важно, если Вы начали программировать на внутреннем языке MIDAS-монитора.

PostScript Вывод в формате PostScript является очень важным для представления Ваших результатов. Все современные редакции астрономических журналов принимают картинки в этом формате. Система $\text{\TeX}/\text{\LaTeX}$, с использованием которой пишет статьи подавляющее большинство астрономов, позволяет легко включать такие картинки в текст.

ВНИМАНИЕ : Всегда существует 5-10% любителей трудностей, которые не используют \TeX и не будут это делать по принципиальным соображениям. Однако я таки советую Вам выучить его сейчас.

Сомневающиеся могут почитать правила приема статей в журналы "Astrophysical Journal" или "Astronomy and Astrophysics" или "Бюллетень CAO".

Можно выделить два момента:

1. Получение PostScript для графических рисунков. Нарисуйте картинку в графическом окне (как это сделать я расскажу ниже), а дальше воспользуйтесь командой: COPY/GRAFH postscript. В текущей директории появится файл с именем postscript.ps. Переименуйте его, чтобы его имя было Вам понятно. Это и есть искомый PostScriptn с Вашим рисунком.

Могу описать и другой способ получения такого же результата, а Вы его разберите на досуге для понимания MIDAS-логики:

```
ASSIGN/GRAFH postscript
PLOT/TABLE ...
ASSIGN/GRAFH g,0
```

2. Получение PostScript-картинки для двумерного изображения. Визуализируйте изображение в image-дисплее, а потом воспользуйтесь командой: COPY/DISPLAY p5=noprint. В текущей директории появится файл с именем image.ps. Переименуйте его по своему усмотрению.

ВНИМАНИЕ : Следите, чтобы Ваш image-дисплей не выходил за рамки экрана — в противном случае будет плохо. Программа честно об этом предупреждает, но кто же читает эти сообщения...

Замечание : Команда содержит существенно большее количество возможностей. Надеюсь, Вы их изучите.

Вывод текста на терминал Вывод текста на терминал осуществляется с помощью команды

```
WRITE/OUT "text-string".
```

5.1. Немного подробнее о цепочке MIDAS → PostScript → LaTeX

Продемонстрируем возможности формата PostScript для представления результатов и встраивания картинок в LaTeX-файл двумя примерами:

Первый пример показывает, как вставить PostScript файл, полученный в MIDAS, с именем HS.ps с использованием окружения "psfig". Результат показан на Рисунке 1. Учтите, что существует еще 38 способов, связанных с использованием других окружений.

```
\documentclass{article}
\usepackage{psfig}

\begin{document}
```

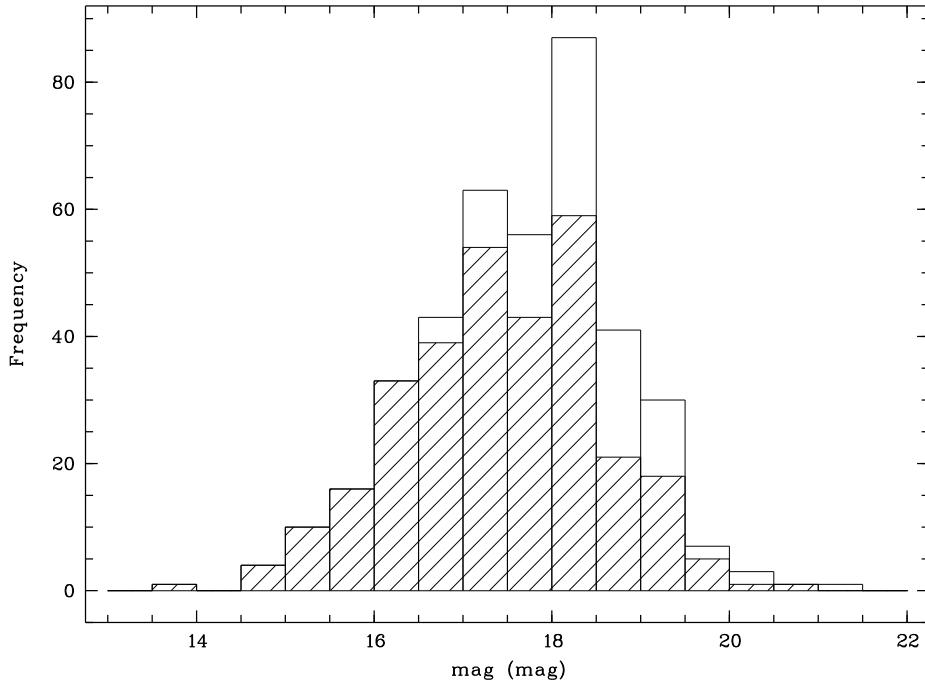


Рис. 1. Эффективность обзора Гамбург/САО по поиску эмиссионных галактик

```
\begin{figure}[htbp]
\centering{
\hspace*{-1.cm}
\vbox{\psfig{figure=HSS.ps, width=14cm, angle=-90}}\par
}
\caption{Эффективность обзора Гамбург/САО по поиску эмиссионных галактик}
\label{Velo}
\end{figure}

\end{document}
```

Полезно знать, что окружение `psfig` имеет следующие ключи, которые записываются в формате, приведенном в примере:

figure= — имя файла;

file= — синоним для `figure`;

height=, **width=** — высота и ширина картинки (если дана одна из них, то вторая будет масштабироваться автоматически и пропорционально);

bbllx=, **bbly=**, **bburx=**, **bbury=** — контролирует размер Postscript BoundingBox (какой кусочек картинки извлечь и нарисовать);

clip= — если эта опция присутствует, то остальная часть картинки (вне указанного BoundingBox) не рисуется;

angle= — задает угол поворота перед выводом на экран;

silent= — работа без вывода.

Второй пример для более сложного случая. Ни одно из известных мне окружений не позволяет правильно и красиво помещать несколько картинок на один рисунок. Но всегда есть способ это сделать, адресуясь напрямую к языку PostScript. Ниже я привожу пример, как можно скомпоновать три картинки в один рисунок. Результат виден на Рисунке 2.

```
\documentclass{article}
\usepackage{psfig}

\begin{document}

\begin{figure}[htbp]
\centering{
  \vspace*{-0.0cm}
  \hspace*{4.0cm}
  \vbox{
    \special{psfile=V432_2d.ps
      hoffset=-350 voffset=-300 hscale=45 vscale=45 angle=0}
    \special{psfile=VV432rot.ps
      hoffset=-120 voffset=75 hscale=38 vscale=50 angle=-90}
    \special{psfile=VV432_2d.ps
      hoffset=-380 voffset=-90 hscale=74 vscale=70 angle=-90}
    \special{psfile=VV432PA1a.ps
      hoffset=-380 voffset=-330 hscale=74 vscale=70 angle=-90}
  }\par
  \vspace*{24.5cm}
}
\caption{Сводная информация о галактике VV432}
\label{VV432_fig}
\end{figure}
\end{document}
```

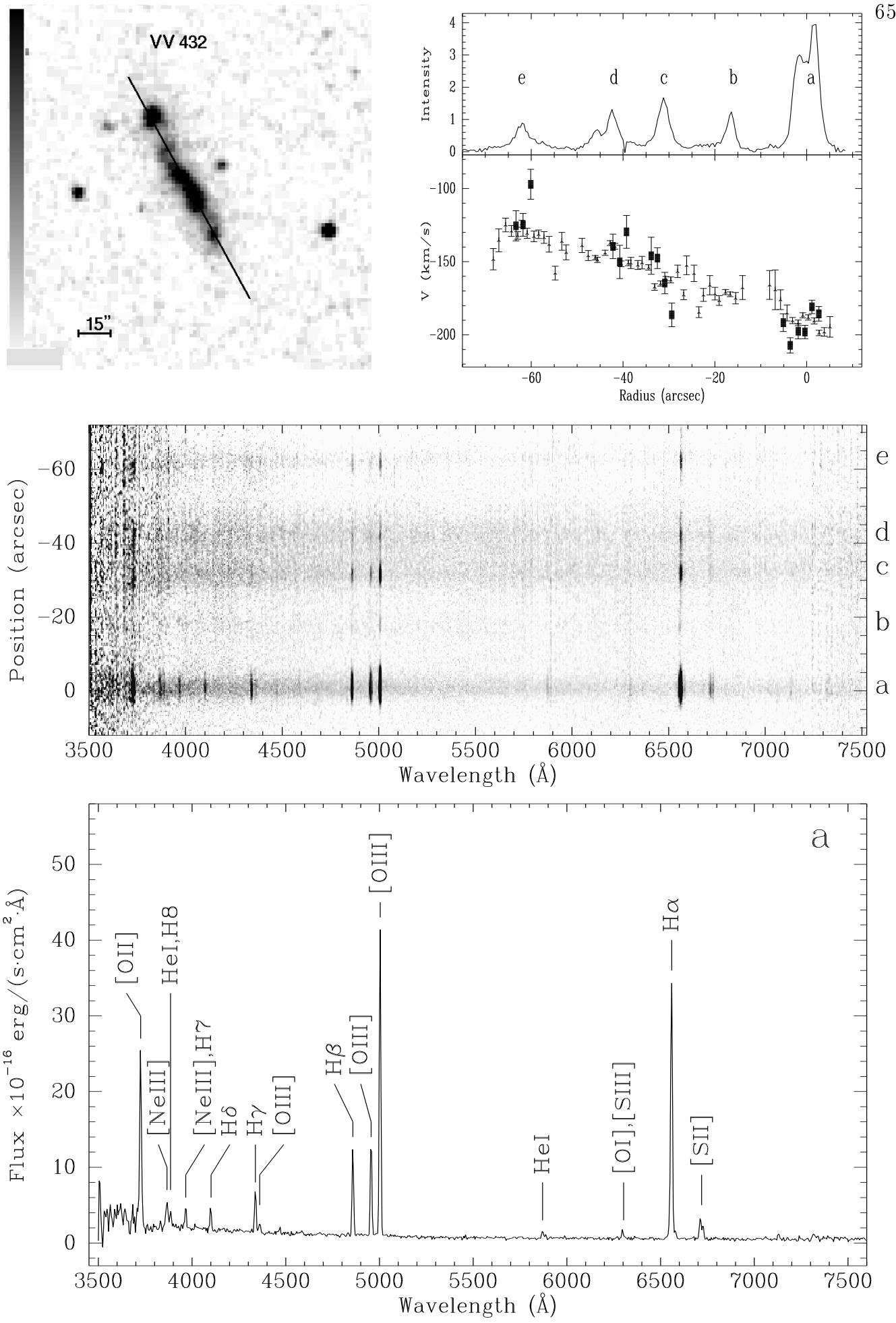


Рис. 2. Сводная информация о галактике VV432

6. MIDAS и операционная система — их взаимодействие

MIDAS - это хорошо! (Без MIDAS - плохо...), А MIDAS+UNIX - лучше!!!

Я думаю, Вы уже убедились, что знание операционной системы намного упростит Вам задачу освоения MIDAS. Если нет, то уверен, что первая же потеря важных данных из-за незнания UNIX подтолкнет Вас в этом направлении. Я же хотел теперь поподробнее остановиться на взаимодействии MIDAS+UNIX.

Итак, мы уже знаем, что любая команда операционной системы может быть выполнена из-под монитора с использованием символа "\$". Но поверьте это еще не все существующие полезные возможности! Можно переключить монитор в режим исполнения только команд UNIX при помощи команды `SET/MIDAS_SYSTEM ENVIRONMENT=host`. После этого работают все возможности редактирования командной строки, о которых я говорил в предыдущей секции, а знак "\$" писать не надо. Одно плохо — MIDAS-команды не исполняются. Но можно вернуться обратно, исполнив команду `SET/MIDAS_SYSTEM ENVIRONMENT=MIDAS`. И все было бы хорошо, но снова надо писать знак "\$" для UNIX-команд. И вот тогда можно установить `SET/MIDAS_SYSTEM ENVIRONMENT=MidHost` и иметь все разом. Тогда монитор будет пробовать исполнять любую, написанную Вами команду, а если он ее не найдет, то отдаст ее UNIX-shell.

Но и это еще не все!

6.1. Перенаправление ввода-вывода

С версии 94NOV в MIDAS начало работать перенаправление ввода-вывода. Это обозначает, что любая информация, выдаваемая на терминал программой, может быть перенаправлена в файл или подана на вход другой программе.

Например, если Вы хотите скопировать содержание MIDAS-таблицы в ASCII-файл, надо выполнить две команды:

```
> ASSIGN/PRINT file mytable.dat — назначение вывода в файл.  
> PRINT/TABLE mytable — вывод.
```

Используя возможность переназначения вывода, можно теперь написать всего одну команду:

```
> WRITE/TABLE mytable >mytable.dat — эквивалентна по результату двум вышеописанным. Файл 'mytable.dat' будет создан в текущей директории.  
> WRITE/TABLE mytable »mytable.dat — дописать в уже существующий файл.
```

ВНИМАНИЕ : Между перенаправлением ввода-вывода для UNIX и MIDAS существует разница. Если shell в UNIX строку типа `cat file > aaaa` поймет правильно при любом количестве пробелов (например, `cat file>aaaa`), то в MIDAS Вы ДОЛЖНЫ писать знаки ">" "<" всегда после пробела, а имя файла пробелом не отделять.

Если Вы хотите записать данные в файл и одновременно вывести на терминал, то это делается вот так:

```
> WRITE/TABLE mytable >mytable.dat+terminal
```

Если Вы не хотите вывода на терминал вообще, то существует специальная форма перенаправления вывода для этого случая (нуль-устройство):

> WRITE/TABLE mytable >Null

Ввод данных со стандартного входа имеет гораздо меньше приложений и туманен для MIDAS, с моей точки зрения (скажем, я до сих пор не понимаю, какой параметр при этом работает, ведь в MIDAS-процедуре их может быть 8). Однако вот, например, полезное ему применение:

`$ls a*.bdf >mydata.dat` — запись всех имен изображений, начинающихся с "a", в файл.

> READ/DESCR <mydata.dat — чтение из этого файла имен и выдача содержания стандартных дескрипторов на терминал. Например, то же действие при помощи стандартных MIDAS-команд выглядит вот так: > CREATE/ICAT OBJ a*.bdf — создание каталога изображений;

> EXECUTE/COMMAND READ/DESCR OBJ.cat — выдача стандартных дескрипторов для всех изображений из каталога.

И последний шик для ввода-вывода это, конечно же, использование программного канала ("трубы" на UNIX жаргоне). Это обозначает, что вывод MIDAS-программы можно перенаправить на вход UNIX-программе. И наоборот.

Пример :

> READ/DESCRIPTOR mama | \$grep CUNIT — выдача содержания стандартных дескрипторов изображения mama.bdf и поиск среди них содержания нужного дескриптора CUNIT.

> \$pwd | WRITE/KEYWORD INPUTC — запись имени текущей директории, в которой ведется обработка, в переменную INPUTC.

Замечание : Замечу, что при использовании программного канала между несколькими UNIX-командами знак \$ ставится только один раз — в начале.

Таким образом, команда > \$ls | \$grep it — неправильная, а команда > \$ls | grep it — правильная.

6.2. Проблема отцов и детей и ее приложение к команде cd

Детального объяснения, с моей точки зрения, требует часто задаваемый пользователями вопрос: "Почему не работает UNIX-команда **cd**, выполненная из-под MIDAS как, например, > \$ cd ..??" Она работает — просто Вы этого не видите. Дело в том, что этот вопрос касается известной проблемы отцов и детей, перекочевавшей из жизни в операционную систему. Исполнение любой команды в UNIX — это порождение процесса-потомка процессом-отцом (вообще, вся система как совокупность одновременно работающих программ порождается отцом всех процессов). После того как процесс-сын выполняет свое предназначение, он завершает работу, извещая отца только о том, что он выполнил/не выполнил свое задание. Исполняя любую UNIX-команду из-под MIDAS, Вы порождаете процесса-сына, имея MIDAS-монитор как процесса-отца. Сын благополучно выполняет команду перехода, о чём и сообщает отцу. И завершается. А отец за сына не в ответе — он остался в старом состоянии. Так вот, чтобы встрихнуть отца и перейти в другую директорию, существует специальная MIDAS-команда **CHANGE/DIR** — пользуйтесь ей, и все будет работать.

Замечание : Достаточно продвинутые пользователи *MIDAS* начинают через некоторое время эмулировать все *UNIX*-команды в *MIDAS*-окружении. Скажем, они создают новую *MIDAS*-команду *cd*:

> **DEFINE/COM cd CHANGE/DIR**

После этого команда > *cd .. /midwork* сработает. Но обратите внимание, что это не команда > *\$cd .. /midwork!*

6.3. Системные команды, существующие всегда

Очень полезно знать, что в любом *MIDAS* (даже если Вы работаете в *VMS*) всегда существует общий набор команд, а следовательно, их можно всегда использовать как при работе, так и при написании командных файлов. Эти команды, перечисленные ниже, начинаются со знака —, и вызывают, на самом деле, следующие *UNIX*-команды:

—DIR	\$ls
—COPY	\$cp
—@	\$sh
—DELCNF	\$rm -i
—DELETE	\$rm -f
—RENAME	\$mv
—TYPE	\$cat
—MORE	\$page
—PRINT	\$lpr

Пример :

> —COPY MID_WORK:tmp.tbl ./ — копирование чего-то в текущую директорию.

Еще несколько замечаний о взаимодействии *MIDAS* и операционной системы я сделаю в дальнейшем, объясняя запуск *MIDAS* в отсоединенном режиме. Здесь же хочу напомнить то, что уже говорил: несмотря на то, что монитор не знает разницы между малыми и большими буквами, операционная система их различает, а значит, пишите имена файлов правильно.

И это конец второй сказки о *MIDAS*.

Список литературы

MIDAS Users Guide, 1995a, Volume A
MIDAS Users Guide, 1995b, Volume B

Но даже и эта вторая премудрость — ничто рядом с третьей...

Леонид Соловьев. "Повесть о Ходже Насреддине".

Стандартная система редукции астрономических данных MIDAS

Лекция III. Возможности графики и визуализации

А.Ю.Князев

Специальная астрофизическая обсерватория РАН, п.Нижний Архыз, 357147, Россия

...ничто рядом с третьей премудростью, сияние которой можно сравнить только с ослепительным блеском солнца и глубину которой можно сравнить только с глубиной океана.

Леонид Соловьев. "Повесть о Ходже Насреддине"

Пришло время более детально познакомиться с возможностями MIDAS-графики и image-дисплея — в дальнейшем я буду называть это пакетом визуализации. При интерактивной работе в мониторе необходимо достаточно подробно представлять их свойства и возможности, а при создании рисунков для статей при помощи MIDAS эти знания просто жизненно важны.

1. Что такое image и графические дисплеи. Их возможности.

Для визуализации изображений используется механизм дисплейных окон. MIDAS запускает свой дисплей-сервер (IDI), который взаимодействует с X11 и MIDAS-монитором. Все команды в MIDAS, использующие дисплей, работают через этот сервер.

1.1. Image-дисплей

Image-дисплей используется в MIDAS для визуализации изображений. Самая необходимая команда при работе с image-дисплеем — это LOAD/IMAGE.

ВНИМАНИЕ : Для корректной работы image-дисплея необходимо иметь работающий X Window в режиме 256 цветов. Правда, в последних версиях MIDAS существует возможность работы и в том случае, когда сервер X Window работает в режиме 24-bit True-color. Однако хочу заметить, что такая работа еще не является гарантированной. Установка режима работы делается в этом варианте командой INITIALIZE/DISPLAY. Для ознакомления с особенностями работы в таком режиме читайте HELP [News].

Существует возможность открытия до 10 image-дисплеев (от 0-го до 9-го) одновременно. Команда создания image-дисплея — CREATE/DISPLAY. Каждый дисплей может иметь до 12 каналов — это значит, что Вы можете загрузить 12 изображений одновременно в 12-ти разных каналах и рассматривать или работать с ними, указывая номер канала. Число каналов создаваемого дисплея задается в команде CREATE/DISPLAY (вместе с размерами дисплея), а номер текущего канала устанавливается командой DISPLAY/CHANNEL. Эти каналы могут быть такого же самого размера, как и размер дисплейного окна, но могут быть и больше, и тогда весьма полезна команда SCROLL/CHANNEL. Все команды работы с каналами имеют квалификатор CHANNEL: BLINK/CHANNEL, CLEAR/CHANNEL, DISPLAY/CHANNEL, SCROLL/CHANNEL, SHOW/CHANNEL, ZOOM/CHANNEL.

Дополнительный канал созданного image-дисплея всегда используется как оверлейный (графический) канал. Таким образом, поверх визуализированных изображений можно также рисовать графические картинки, перенаправив вывод с помощью команды ASSIGN/GRAFICS). Для работы с каналом в оверлейном режиме существует многочисленный набор команд, которые я охарактеризую ниже.

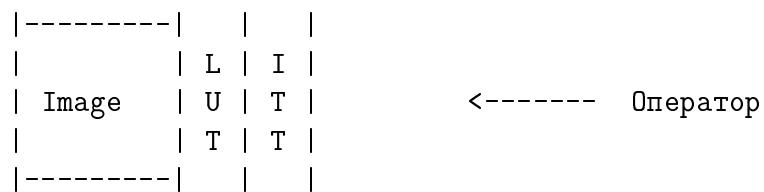
ВНИМАНИЕ : Помните, что все команды переназначения начинают работать ТОЛЬКО после создания соответствующих дисплеев и окон.

ВНИМАНИЕ : Хотя разработчики MIDAS утверждают, что оверлейный канал создается всегда, я обращаю Ваше внимание на то, что число каналов в созданном image-дисплее должно быть больше 1 для гарантированного использования возможности оверлея. В противном случае MIDAS может отрабатывать задаваемые Вами команды, не выводя никаких результатов и ничего не сообщая.

Также для каждого image-дисплея можно создать:

- zoom-окно (команда создания — CREATE/ZOOM) — в него выводится увеличенный в несколько раз участок визуализированного изображения. Существуют команды, которые автоматически создают это окно (например, VIEW/IMAGE);
- shadow-дисплей (параметр команды CREATE/DISPLAY) — параллельный вывод результатов визуализации на любой компьютер в сети, имеющий X Window.

Рассмотрим общую схему image-дисплея, которую я показал на приведенном ниже рисунке. Оператор-астроном смотрит на изображение справа. Image-дисплей позволяет раскрашивать визуализированные изображения в псевдо-цвета (256 цветов максимум), менять их интенсивность. Но вариантов цветовых гамм очень много. За конкретный вариант цветовой гаммы отвечает LUT-таблица (Look-Up Table), которая адаптирует человеческое восприятие цветов к изображенным на экране. Однако есть еще один слой, накладывающий дополнительный закон на LUT-таблицу. Этот слой называется ITT-таблица (Intensity Transfer Table), и его также можно менять.



Рассмотрим теперь подробнее различные наборы команд для работ с image-дисплеем.

ВНИМАНИЕ : Иногда (а для некоторых пользователей, у которых пальцы двигаются по клавиатуре быстрее, чем думает голова — очень часто) пользователь совершает набор движений, после которых связь между MIDAS-монитором и IDI-сервером разрушается. Это выражается в том, что графика перестает работать вообще либо на экране возникает нечто неожиданное, изображения не визуализируются и так далее. В этом случае необходимо помнить о могучей команде **RESET/DISPLAY**, которая восстанавливает эту связь. Только не удивляйтесь, что после выполнения данной команды все графические и image-дисплеи исчезнут — их необходимо создать заново.

1.1.1. LUT — Look-Up Tables

Принцип показа визуализированных интенсивностей различными цветами очень прост: весь их перепад (а его границы определяются параметром **CUTS** команды **LOAD/IMAGE**) разбивается на 256 интервалов, каждый из которых обозначается своим цветом. Поскольку любой цвет определяется сочетанием трех основных: R — красного, G — зеленого и B — голубого, — то LUT-таблица просто определяет для каждого из 265 интервалов интенсивность этих трех. Она изменяется от 0 до 1. Например, если ее изменение одинаково линейно от 0 до 1 для RGB, то цвет будет изменяться от белого к черному, а если наоборот — от 1 до 0, то от черного к белому.

В MIDAS существует большой набор стандартных LUT-таблиц. Их можно модифицировать в интерактивном режиме. LUT-таблицы помогают выделить на визуализированном изображении его особенности различным распределением цвета. Чтобы посмотреть, как выглядит одно и то же изображение при использовании различных LUT-таблиц, выполните команду **TUTORIAL/LUT**. Она же покажет Вам и основные принципы работы с этими таблицами. Все команды работы с ними имеют квалификатор **LUT**. Это: **CLEAR/LUT**, **CREATE/LUT**, **DISPLA/LUT**, **GET/LUT**, **LOAD/LUT**, **MODIFY/LUT**, **SET/LUT**, **TUTORIAL/LUT**.

Замечание : По умолчанию всегда загружается LUT-таблица, соответствующая равномерному распределению цвета от черного до белого. Ее имя — "ramp".

Некоторые из существующих стандартных LUT-таблиц имеют следующие имена: **backgr**, **color**, **heat**, **light**, **pastel**, **pseudo1**, **pseudo2**, **rainbow**, **rainbow1 ... rainbow4**, **random**, **random1 ... random4**, **smooth**, **staircase**, **stairs8**.

1.1.2. ITTs — Intensity Transfer Tables

Как я уже говорил выше, при помощи ITT-таблиц делается преобразование текущих значений загруженной LUT-таблицы в соответствии со значениями ITT (накладывается некоторая функция преобразования). Например, можно наложить на черно-белое распределение "ramp" дополнительно логарифмический закон (команда **LOAD/ITT log**), и тогда Вы получите такое же распределение интенсивностей, как на фотопластинке.

В MIDAS существует большой набор стандартных ITT-таблиц. Их можно модифицировать в интерактивном режиме. Чтобы посмотреть, как выглядит одно и то же изображение при использовании различных ITT-таблиц, выполните команду TUTORIAL/ITT. Она же покажет Вам и основные принципы работы с этими таблицами. Все команды работы с ними имеют квалификатор ITT. Это: CLEAR/ITT, GET/ITT, LOAD/ITT, MODIFY/ITT, SET/ITT, TUTORIAL/ITT.

Замечание : По умолчанию всегда загружается ITT-таблица, соответствующая нулевому дополнительному закону. Ее имя — "ramp".

Некоторые из существующих стандартных ITT-таблиц имеют следующие имена: neg, expo, log, neglog, jigsaw, staircase.

1.1.3. Работа с курсором

Очень часто для работы с визуализированными изображениями необходимо использовать курсор. Каждый image-дисплей может работать с двумя курсорами одновременно (например, EXTRACT/TRACE): один будет управляться "мышью", а второй — клавиатурой. Если курсор управляется "мышью", то левая кнопка ВСЕГДА служит для чтения текущего положения курсора, а правая и средняя — для выхода из программы.

ВНИМАНИЕ : Для чтения текущего положения курсор должен быть неподвижен, что иногда очень трудно сделать при наличии "мыши" плохого качества.

Если курсор управляется с клавиатуры, то направление движения определяется стрелками на клавиатуре, размер шага — клавишами от 1 до 9, а для чтения текущего положения служит клавиша <Enter>.

Очень многие команды используют курсор прямоугольной или круглой формы (например, VIEW/IMAGE). В этом случае каждому курсору соответствует так называемая "область интереса" (ROI — a region of interest). Размер этой области может меняться либо при помощи клавиш (от 0 до 9 на клавиатуре), либо при помощи других, специально оговоренных, клавиш.

Все команды работы с курсором имеют квалификатор CURSOR: GET/CURSOR, CLEAR/CURSOR, SET/CURSOR.

1.1.4. Графика

Как я уже говорил, у image-дисплея существует оверлейный канал, в котором пользователь также может рисовать все то, что он рисует на графическом дисплее (мы будем рассматривать все эти команды далее). Кроме того, существует набор дополнительных команд, позволяющих управлять рисованием в оверлейном канале (разрешать, запрещать, очищать): SET/OVERLAY, CLEAR/OVER, CLEAR/CHAN OVER — и набор команд, которые могут рисовать ТОЛЬКО в оверлейном канале: DRAW/... (CIRCLE, RECTANGLE ...), LABEL/DISPLAY.

ВНИМАНИЕ : Очень часто пользователь путает команду CLEAR/OVERLAY — запрещающую вообще вывод в оверлейный канал, с командой CLEAR/CHANNEL OVERLAY, очищающей оверлейную память.

1.1.5. Алфавитно-цифровая память

Каждое дисплейное окно может иметь (а может и не иметь) так называемую алфавитно-цифровую память, которая создается командой CREATE/DISPLAY. Эта память содержит 3 строки и позволяет выводить из нее различные характеристики визуализированного изображения. Команды работы с памятью: LABEL/DISPLAY, CLEAR/ALPHA.

1.2. Графический дисплей и графический пакет

Существует возможность открыть до 10 графических дисплеев (от 0-го до 9-го) одновременно. Команда создания графического дисплея — CREATE/GRAph. Работа с графическим дисплеем является частью работы с графическим пакетом MIDAS в целом.

Все команды для работы с графическим пакетом MIDAS можно разбить на три основные группы:

Общие команды — это команды создания/уничтожения дисплейных графических окон, а также команды настройки, просмотра состояния и команды ввода-вывода на различные внешние устройства.

- CREATE/GRAph, DELETE/GRAph, CLEAR/GRAph — команды создания, уничтожения и очистки;
- SET/GRAph, SHOW/GRAph — команды настройки и просмотра текущего состояния параметров графического пакета. Команда SET/GRAph является, с моей точки зрения, самой важной для графического пакета, так как позволяет производить все настройки. В Таблице 1 я кратко перечислил все возможные параметры этой команды и их назначение;
- ASSIGN/GRAph, COPY/GRAph — команды перенаправления вывода. Обычно используются для получения копии созданного рисунка;
- PLOT/AXES, OVERPLOT/AXES, LABEL/GRAph, OVERPLOT/LINE, OVERPLOT/SYMBOL, OVERPLOT/GRID — рисование осей, символов, линий, формул и т.д. Необходимо знать, что, как и в L^AT_EX, в графическом пакете MIDAS можно использовать специальные команды для рисования специальных символов, подъема и опускания текста. Я суммировал это в приведенных Таблице 2 и Таблице 3.

В качестве примера использования этих знаний, хочу привести следующий:

Пример: Если написать в MIDAS-мониторе следующую строку

```
LABEL/GRA "e\{\!\u(x\{\!\u2\}+y\{\!\u2\})\}= -\alpha+\beta \sin\{\!\u2\}\theta
```

то результат на графическом дисплее будет выглядеть так:

$$e^{(x^2+y^2)} = (\alpha + \beta) \sin^2 \theta,$$

Команды рисования — это команды для создания новых рисунков (.../PLOT) на основе различных структур данных (спектров, таблиц ...) или добавления изображения к уже существующему (.../OVERPLOT).

Таблица 1. Параметры команды SET/GRAFH

Параметр	Возможные значения и значение "по умолчанию"
DEFAULT	Устанавливает значения всех параметров "по умолчанию"
XAXIS=	пределы и разметка по X-оси. AUTO или xstart,xend,xbig_tick,xsmall_tick в мировых координатах; когда xsmall_tick < 0 рисуется в логарифмической шкале; значение "по умолчанию" — AUTO
YAXIS=	пределы и разметка по Y-оси. AUTO или ystart,yend,ybig_tick,ysmall_tick в мировых координатах; когда ysmall_tick < 0 рисуется в логарифмической шкале; значение "по умолчанию" — AUTO
ZAXIS=	тоже, что для YAXIS
FRAME=	RECT или SQUA; значение "по умолчанию" — RECT
XSCALE=	масштабирование. AUTO, масштаб в мировых units/per mm или размер рисунка
YSCALE=	масштабирование. AUTO, масштаб в мировых units/per mm или размер рисунка
XOFFSET=	NONE или сдвиг от левой границы окна
YOFFSET=	NONE или сдвиг от нижней границы окна
XFORMAT=	Формат подписи по X-оси. NONE, AUTO или описание формата (смотри HELP)
YFORMAT=	Формат подписи по Y-оси.
ZFORMAT=	тоже, что для YFORMAT
PMODE=	-1 — рисование без осей и подписей, 0 — рисование с подписанными осями, 1 — рисование с подписанными осями и некоторой дополнительной информацией, 2 — рисование с подписанными осями и полной информацией ("по умолчанию")
FONT=	назначение фонта, который будет использован при написании текста на рисунке; значение "по умолчанию" — 1; (see below)
LTYPE=	тип линии. От 1 (непрерывная) до 6 (прерывная длинная-короткая); значение "по умолчанию" — 1; 0 соответствует отсутствию соединения точек линиями (для таблиц)
STYPE=	тип символа: от 1 (точка) до 21 (заполненный ромб); значение "по умолчанию" — 4 (крестик); 0 соответствует отсутствию символа
LWIDTH=	установка ширины рисуемой линии; 0 или 1 это нормальная ширина; 2, 3 и 4 для увеличения толщины (ширины)
SIZE=	число, устанавливающее масштабный фактор для рисуемого символа; значение "по умолчанию" — 1
TSIZE=	число, устанавливающее масштабный фактор для текста; значение "по умолчанию" — 1
TWIDTH=	установка толщины текста; 0 или 1 это нормальная ширина; 2, 3 и 4 для увеличения толщины; значение "по умолчанию" — 1; Результат будет виден ТОЛЬКО на PostScript-картинке!
BINMODE=	OFF или ON; "по умолчанию" — OFF
COLOUR=	цвет. От 0 до 7; "по умолчанию" — черный (1);
BCOLOUR=	цвет фона. От 0 до 7; "по умолчанию" — белый (0);
CLEARGRA=	ON или OFF. при OFF графический дисплей не очищается даже при использовании следующей команды PLOT; значение "по умолчанию" — ON

Таблица 2. Использование специальных символов в графическом пакете MIDAS

Metacharacter	Meaning
\{	начало группы
\}	конец группы
\^	поднять на половину высоты символа
\!u	поднять на половину высоты символа
_	опустить на половину высоты символа
\!d	опустить на половину высоты символа
\<	сдвинуть влево на ширину символа
\+	увеличить размер текста на 20%
\-	уменьшить размер текста на 20%
\!	рассматривать следующую за этим последовательность как последовательность метасимволов (это разрешает использовать метапоследовательности, начинающиеся с 'n', не рассматривая их как переход на новую строку)
\0	установить фонт 0 ("по умолчанию")
\1	установить фонт 1 (Quality roman font)
\2	установить фонт 2 (Greek font)
\3	установить фонт 3 (Script font)
\4	установить фонт 4 (Old English)
\5	установить фонт 5 (Tiny roman font)
\[увеличить толщину линий (bolding; optional)
\]	уменьшить толщину (bolding; optional)
\n	переход на новую строку
\~	Нарисовать символ '~,
\`	Нарисовать символ '~,
\`\\	Нарисовать символ '\\
\\\`	Нарисовать символ '\\'

Таблица 3. Использование ТЕХ-подобных символов в графическом пакете MIDAS

\AA	\Alpha	\Aquarius	\Aries
\Beta	\Cancer	\Capricorn	\Chi
\Delta	\Earth	\Epsilon	\Eta
\Gamma	\Gemini	\Iota	\Jupiter
\Kappa	\Lambda	\Leo	\Libra
\Mars	\Mercury	\Moon	\Mu
\Neptune	\Nu	\Omega	\Omicron
\Pi	\Phi	\Pisces	\Pluto
\Psi	\Rho	\Sagittarius	\Saturn
\Scorpio	\Sigma	\Sqrt	\Tau
\Taurus	\Theta	\Upsilon	\Uranus
\Venus	\Virgo	\Xi	\Zeta
\aleph	\alpha	\asteroid	\beta
\bigcirc	\black	\blue	\cent
\chi	\circ	\cyan	\clover
\clubsuit	\comet	\dag	\ddag
\default	\delta	\diamond	\div
\downarrow	\epsilon	\equinox	\equiv
\eta	\firtree	\gamma	\ge
\greek	\green	\hbar	\heart
\infty	\int	\iota	\italic
\kappa	\lambda	\larrow	\le
\magenta	\mp	\mu	\nabla
\neq	\nu	\odot	\oint
\old	\omega	\omicron	\oplus
\otimes	\palm	\paragraph	\parallel
\partial	\perp	\phi	\pi
\pm	\propto	\psi	\red
\rho	\rightarrow	\roman	\script
\shield	\sigma	\snow	\spade
\sqrt	\sum	\tau	\theta
\times	\tiny	\uparrow	\upsilon
\varepsilon	\varphi	\vartheta	\white
\xi	\yellow	\zeta	

PLOT/CONTOUR, PLOT/COLUMN, PLOT/ROW, PLOT/DESCRIPTOR, PLOT/GRAY, PLOT/HISTOGRAM, PLOT/KEYWORD, PLOT/PERSPECTIVE, PLOT/TABLE OVERPLOT/ERROR, PLOT/VECTOR . . .

Команды для работы с графическим курсором — команды для снятия текущих координат, установления типа курсора и так далее.

GET/GCURSOR, CENTER/GAUSS, MODIFY/GCURSOR, INTEGRATE/LINE, INTEGRATE/STAR, INTEGRATE/STAR . . .

1.3. Пакет визуализации и создание рисунков

Хорошее знание пакета визуализации в MIDAS (работа с image и графическим дисплеями) избавляет пользователя-астронома практически полностью от необходимости работать с другими внешними пакетами при подготовке рисунков для статей. Вы как бы погружены в единую среду, в который производится как обработка данных (или расчет моделей), так и подготовка к выдаче Ваших результатов "наружу". Конечно, существует ряд рисунков, которые очень трудно создать в MIDAS, и поэтому я не берусь утверждать, что этот пакет есть панацея от всех бед. Но грамотный пользователь очень легко расширяет его возможности путем написания командных файлов.

ВНИМАНИЕ : *При создании рисунков для статей ОБЯЗАТЕЛЬНО оформляйте их в виде командных файлов: один рисунок — один командный файл. Храните этот командный файл и данные для него в одном месте. При необходимости переделать/доделать рисунок Вы просто слегка модифицируете имеющийся командный файл.*

Я хотел бы привести в пример два командных файла с результатами их работы, конечно. Как будет говориться в дальнейшем, командный файл в MIDAS обычно имеет расширение ".prg" и, если он находится в текущей директории, вызывается на исполнение MIDAS-монитора командой **@@ ИМЯ_ФАЙЛА**. При разборе приведенных примеров учтите, что графический и image-дисплей я уже создал до исполнения командных файлов.

Пример :

Есть изображение **h038Rbffd.bdf** с прямым снимком галактики. Необходимо его нарисовать для статьи, наложив контуры на центральную часть галактики. На рисунке надо показать масштаб. Я написал командный файл, который назвал **2D.prg** и вызвал его командой **@@ 2D**. Ниже даны краткие пояснения некоторым командам, а полной разбор строки со всеми параметрами оставлен интересующимся. Результат работы данной программы приведен на Рисунке 1.

LOAD/IMA h038Rbffd 0 4 1002,1060 cuts=11300,11650 — визуализуем изображение в черно-белом цвете;

LOAD/ITT neg — устанавливаем ITT-таблицу, которая выбирает светлый цвет (белый) для фона и темный (черный) для галактики;

@a vertlut ? black 10 f5.0 — рисуем вертикальный бар с распределением цвета в зависимости от интенсивности (при этом стандартная команда выводит на дисплей (в оверлейный канал) значения уровней);

CLEAR/CHANNEL OVER — очищаем оверлейный канал, чтобы этих надписей не было;

ASS/GRA d,0 — переназначаем вывод графики на image-дисплей;

SET/GRAPH tsize=0 ssize=0 — чтобы следующая команда не рисовала ничего лишнего, устанавливаем нулевые размеры для линий и букв;

PLOT/AXES [<,<:>,>] — совмещаем визуализированное изображение с оверлейной памятью;

ВНИМАНИЕ : Эта команда есть тайное знание, которым я делаюсь со всеми! Помните о ней! И всегда используйте, если Вы хотите, чтобы нарисованная контурная карта лежала поверх визуализированного изображения, а не стояла задумчиво в стороне.

SET/GRA tsize=1 ssize=1 — возвращаем исходные числа;

DRAW/LINE 100,50,200,50 S ? black — рисуем горизонтальный бар. Чтобы он выглядел достаточно "жирным" (мы ведь потом будем рисунок сжимать), проводим три линии с легким сдвигом одна относительно другой. Рисуем командами для работы с оверлеем (а не с графикой), поскольку эти команды работают в единицах экрана, а не изображения (которое я загрузил с увеличением 4), значит сдвиг на один в координатах изображения — это сдвиг на 4 в координатах экрана, и мой трюк с тремя линиями не сработает (можете попробовать);

DRAW/LINE 100,51,200,51 S ? black

DRAW/LINE 100,52,200,52 S ? black

DRAW/LINE 99,46,99,56 S ? black — рисуем вертикальную черту на левом конце бара.

Используем тот же самый прием;

DRAW/LINE 100,46,100,56 S ? black

DRAW/LINE 101,46,101,56 S ? black

DRAW/LINE 199,46,199,56 S ? black — рисуем вертикальную черту на правом конце бара;

DRAW/LINE 200,46,200,56 S ? black

DRAW/LINE 201,46,201,56 S ? black

LABEL/DIS "5" 55,145 ? black 2 — рисуем надпись рядом с баром, показывающим масштаб. Работаем в координатах экрана (для однообразия);

LABEL/DIS "" 57,157 ? black 2

LABEL/DIS "" 57,162 ? black 2

LABEL/DIS "303 pc" 25,112 ? black 2

SET/GRA lwidth=3 color=8 — устанавливаем цвет и ширину линий, которыми будем рисовать контуры;

OVERPLOT/CONT h038Rbfff ? 11700,11800:15000:600 LTYPE — рисуем контуры только некоторой центральной (самой яркой) части;

SET/GRA lwidth=1 color=1 — возвращаем исходные числа;

`COPY/DIS p3=P p5=noprint p6=PC8NZ` — копируем получившийся результат в PostScript-файл;

`$mv screen10.ps HS0822+3542R_direct.ps` — переименовываем стандартный файл в нужное имя.

Пример :

Есть таблица **qq.tbl** со значениями измеренных скоростей для различных областей одной галактики и расстояниями для этих областей от центра галактики. Необходимо нарисовать для статьи эти точки, посчитать и наложить на рисунок получающуюся кривую скоростей (линейное приближение). Показать точки, не совпадающие с этой кривой скоростей. Я написал командный файл, который назвал **Rot.prg**, изываю его командой `@@ Rot`. Результат работы данной программы приведен на Рисунке 2.

`SET/GRA ssize=1.5 tsize=1.3 font=1 twidth=2 stype=3` — производим настройки графического пакета для рисования линий и символов;

`PLOT/AXES 0,50 950,1250 ? "\+\1 Distance (arcsec)" "\+\1 Velocity (km/s)"` — рисуем оси вместе с подписями;

`SELECT/TAB qq :POS_C.lt.0` — выделяем в таблице все точки по одну сторону от центра галактики;

`OVERPLOT/TAB qq :POS_CRA :VEL_C` — рисуем их пустыми квадратиками;

`SET/GRA stype=8` — меняем тип рисуемых символов на звездочки;

`SELECT/TAB qq :POS_C.gt.0` — выделяем в таблице все точки по другую сторону от центра галактики;

`OVERPLOT/TAB qq :POS_CRA :VEL_C` — рисуем их звездочками;

`SELECT/TAB qq all` — выделяем в таблице все точки;

`OVERPLOT/ERROR qq :POS_CRA :VEL_C :Err` — рисуем бары ошибок для всех точек;

`REGRESSION/POLYNOM qq1 :VEL_C :POS_CRA 1` — я заранее скопировал во вторую таблицу **qq1.tbl** все точки, за исключением двух, которые сильно отклоняются. Используя точки из этой второй таблицы, я приближаю их полиномом первой степени;

`SAVE/REGRES qq1 coef` — запоминаю полученные коэффициенты;

`COMP/REGRES qq1 :fit = coef` — считаю положения точек в соответствии с полученным приближением;

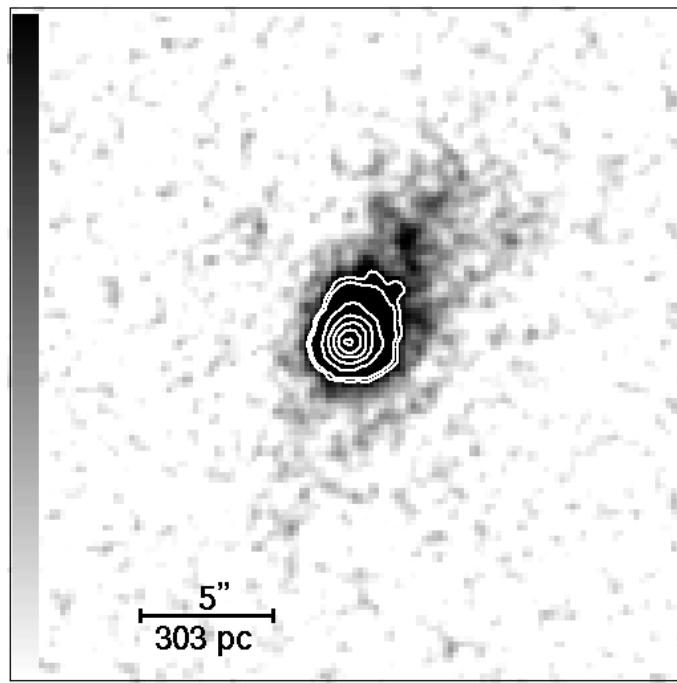


Рис. 1. Изображение галактики HS 0822+3542 с наложенными на его центральную часть контурами. Показан масштаб.

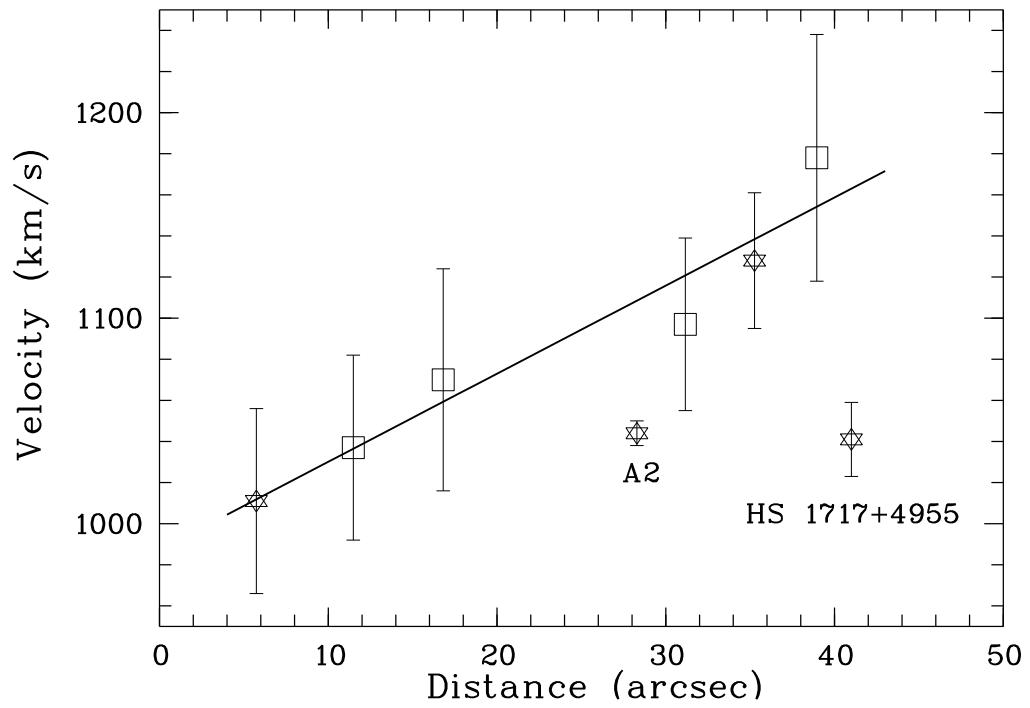


Рис. 2. Кривая вращения для галактики UGC 10806. Звездочками отмечены скорости для областей из удаляющегося, а квадратиками — из приближающегося рукава этой галактики. Вертикальными барами выделены ошибки для каждой точки. Сплошная линия показывает приближение наблюдаемых точек полиномом первой степени. Отмечены положения HS 1717+4955 и области A2, не использованные при построении кривой вращения.

`CREATE/IMA ref 1,40 4,1 nodata` — это мой "ход конем". Мне не нравится вид прямой, проведенной по шести точкам, когда я просто пытаюсь использовать `PLOT/TABLE`, установив перед этим `SET/GRAF STYPE=0 LTYPE=1`. К тому же MIDAS рисует в этом случае только до положений крайних точек. Поэтому я решил построить модель линейного приближения в виде одномерного изображения. Как первый шаг я строю так называемое reference-изображение (в нем нет данных, но есть размерность, шаг и т.д.);

`CONVERT/TABLE vel = qq1 :POS_CRA :fit ref POLY 1` — преобразую колонку таблицы в изображение `vel.bdf`, используя число точек, размерность и шаг;

`SET/GRA lwidth=3` — устанавливаю тип линии для рисования;

`OVER/ROW vel` — разрисовываю построенную модель;

`SET/GRA ssize=1 tsize=1` — возвращаю исходные установки для части параметров;

`LABEL/GRA "A2" 28.5,1025 ? 1` — делаю надписи, указав место на рисунке;

`LABEL/GRA "HS 1717+4955" 41,1005 ? 1`

`SET/GRA ssize=1 tsize=1 twidth=1 stype=5 lwidth=1` — возвращаю исходные установки для всех параметров;

`COPY/GRA postscript` — копирую полученный рисунок в PostScript-файл;

`$mv postscript.ps HS1717_rot.ps` — переименовываю стандартный файл.

И это конец третьей сказки о MIDAS.

Список литературы

MIDAS Users Guide, 1995a, Volume A

MIDAS Users Guide, 1995b, Volume B

Но послушай теперь ... мою ... премудрость, которая стбует всех твоих.

Леонид Соловьев. "Повесть о Ходже Насреддине"

Стандартная система редукции астрономических данных MIDAS

Лекция IV. Работа с различными структурами данных

А.Ю. Князев

Специальная астрофизическая обсерватория РАН, п.Нижний Архыз, 357147, Россия

Но приготовься, чтобы не поразила тебя слишком эта премудрость, ибо от нее легко потерять рассудок — настолько она поразительна, ослепительна и необъятна.

Леонид Соловьев. "Повесть о Ходже Насреддине"

Я думаю, что уже на основе разбора примеров из предыдущей лекции Вы почувствовали, что вдумчивая работа с системой редукции типа MIDAS требует от пользователя определенной логики поведения, сильно отличающейся от логики программирования, скажем, на языках типа FORTRAN или С. Я бы назвал этот стиль UNIX-подобным. Это, конечно, не чистое сравнение — трудно напрямую сравнивать UNIX, скажем, с MIDAS. Но, скажем, если перед продвинутым UNIX-пользователем встает проблема отсутствия какого-то средства для решения небольшой проблемы, то он быстренько-быстренько напишет новый командный файл, используя множество из 600 дополнительных команд этой операционной системы и НИКОГДА не будет писать новой программы на С для выполнения задуманного.

Замечание : Следствие этой аксиомы — если Вы увидели, что человек программирует на С в UNIX, то это либо "зеленый" пользователь, понятия не имеющий, что в UNIX есть команды **grep** и **find**, либо "крутой" системщик, у которого, скажем, монитор упал на бок, а он решил заставить текст на экране повернуться на 90 градусов и для этого переделывает системный драйвер...
(Конечно, это шутка...)

То же самое относится, скажем, и к MIDAS. Если пользователь тут же начинает встраивать в него свои новые С или FOTRAN программы, то он либо действительно ограничен возможностями MIDAS, либо, с большей степенью вероятности, еще просто не понимает, КАК, используя 2–20 MIDAS-команд в командном файле, совершить то, что ему необходимо. Это понимание приходит только с опытом работы.

Однако вернемся к лекциям. Следующую лекцию я хотел бы посвятить рассмотрению двух вопросов:

1. Более детальное ознакомление с различными структурами данных, поддерживаемыми MIDAS.

Замечание : При оперировании изображениями, таблицами и так далее очень важно не только представлять, какое количество разнообразных операций MIDAS позволяет производить над этими структурами (это достаточно просто узнать, посмотрев на HELP по группам команд, имеющим общий квалификатор), но и знать, как обращаться к элементам этих структур.

2. Краткому обзору команд MIDAS по группам и назначениям.

1. Более подробно о работе с различными структурами данных

Я не раз видел, как, казалось бы, умудренный опытом пользователь впадает в прострацию, пытаясь изменить значение какого-нибудь единственного элемента спектра (одномерного изображения) или понять, как при выполнении командного файла вписать одно значение в таблицу. А ведь это делается действительно элементарно — достаточно просто знать, как обращаться к элементам различных структур.

1.1. Изображения. Элементы в изображениях

Поскольку чаще всего в MIDAS астрономы-наблюдатели имеют дело с изображениями (одномерными, двумерными или трехмерными), то я начну свой краткий обзор именно с них.

Файл изображения состоит из шапки, содержащей дескрипторы и тела изображения (массив данных размерности 1-3). Как и в случае FITS-файла, дескрипторы полностью описывают данные: тип, размерность и так далее (более подробно об этом смотрите ниже). Изображение состоит из элементов, или ячеек, или пикселов. Имеется очень большой набор команд, работающих с изображениями, и подавляющее большинство этих команд имеют квалификатор .../IMAGE: READ/IMAGE, WRITE/IMAGE, COMPUTE/IMAGE, COPY/II

Кроме того, MIDAS предоставляет возможность обратиться не ко всему изображению в целом, а к некой прямоугольной его области (частный случай — один пикセル). Формат этого стандартного обращения можно записать в виде:

`frame[x1,y1:x2,y2]`, где

`x` — номер колонки,

`y` — номер строки,

`x1,y1` — координата левого нижнего угла выделяемой области,

`x2,y2` — координата правого верхнего угла выделяемой области.

В случае трехмерного массива вид обращения: `frame[x1,y1,z1:x2,y2,z2]`.

ВНИМАНИЕ : Помните о двух символах, обозначающих начало изображения "<" и конец изображения ">".

При таком стандартном обращении монитор, выполняя Вашу команду, должен из-

влечь эту область из полного изображения и подать на выполнение MIDAS-команде.

Пример :

```
COMPUTE/IMAGE qubodub = 0 — присвоение всем пикселам одномерного изображения значения 0.
COMPUTE/IMAGE qubodub[<:>] = 0 — то же самое.
COMPUTE/IMAGE qubodub[1000:>] = 0 — присвоение пикселам одномерного изображения значения 0. Номера пикселов: от 1000-го до последнего.
COMPUTE/IMAGE qubodub[1000,>:1000,>] = 0 — присвоение всей двумерной области (правый верхний угол) двумерного изображения значения 0.
PLOT/ROW Fabri[<,<,5:>,>,5] 100 — нарисовать 100-ую строку из 5-ой XY-плоскости трехмерного массива данных.
```

Полезно знать, что есть и короткая форма записи. К сожалению, она применима только для ряда частных случаев: строка в двумерном изображении или XY-плоскость трехмерного массива данных.

Пример :

```
PLOT/ROW Fabri@5 100 — нарисовать 100-ую строку из 5-ой XY-плоскости трехмерного массива данных.
```

Конечно, существуют различного рода исключения из этого правила. Это касается команд, у которых уже есть отдельный параметр для задания области работы (например, STAT/IMAGE). В этом случае все зависит от версии MIDAS, в которой Вы работаете.

Замечание : Например, у меня в версии 96NOV команда

COMP/IMA qq[1:50] = qq[51:100] не работает, а команда

COMP/IMA qq[<:50] = qq[51:100] работает, но в других версиях результат может быть иным.

Следующее, что необходимо знать каждому при работе с изображениями — это наличие двух систем координат: мировой и пиксельной — одновременно. Поскольку начало и шаг для элемента изображения (они описаны в дескрипторах изображения) могут быть любыми числами, то адресация к элементу изображения может производиться как по номеру пикселов (пиксельные координаты), так и по номерам абсолютным, вычисляемым с помощью дескрипторов START и STEP (мировые координаты). При этом задание пиксельных координат производится при помощи символа "@". Так [@30:@60] или [30:60] — это разные области одномерного изображения, хотя очень часто мировые и пиксельные координаты совпадают.

ВНИМАНИЕ : Это происходит, когда дескрипторы изображения START и STEP равны 1!

Пример :

Есть одномерный спектр **duboqub.bdf** (изображение), содержащий 1024 числа. Дескриптор START равен 3700.0 (конечно, ангстрем — спектр-то линеаризован, т.е. приведен к равномерному шагу!), а дескриптор STEP равен 3.0 (конечно, ангстрем). Тогда:

COMPUTE/IMAGE **duboqub[3700:3709] = 0** — присвоение трем первым пикселам значения 0. Задание в мировых координатах.

COMPUTE/IMAGE **duboqub[@1:@3] = 0** — то же самое. Задание в пиксельных координатах.

COMPUTE/IMAGE **duboqub[:@3] = 0** — то же самое.

ВНИМАНИЕ : Символы "<" и ">" можно использовать в обеих формах записи (начало/конец - оно и в Африке начало/конец).

Пример :

Есть двумерное изображение **duboqub.bdf** размером 512×512 чисел (пикселей). Дескриптор START равен 5.0,10.0 а дескриптор STEP равен 1.0,-1.0 — вот такой дурной случай. Тогда:

EXTRACT/IMAGE **line = cube[<, @1:>, @1]** — извлечение первой строки. Задание в пиксельных координатах.

EXTRACT/IMAGE **line = cube[<, 10:>, 10]** — извлечение первой строки. Задание в мировых координатах.

Замечание : Очевидно, что обе системы координат — линейные. Постепенно происходит переход к истинно мировой системе координат (так называемой WCS). На сегодняшний день в MIDAS уже есть команда — **GET/CURSOR**, работающая в этой системе координат. Это значит, что если Вы, например, возьмете любое изображение, полученное с оцифрованного Паломарского обзора, то эта команда будет выдавать астрометрические координаты.

ВНИМАНИЕ : Помните, что команда **COMPUTE/IMAGE** работает в мировых координатах. Если же Вам необходимо проделать арифметические операции с несколькими изображениями, забыв про мировые координаты, пользуйтесь командой **COMPUTE/PIXEL**.

Замечание : Команда **COMPUTE/IMAGE** также полезна, поскольку позволяет не только производить математические операции с изображениями, но и использовать ее как калькулятор (содержит очень большое число дополнительных функций).

1.2. Описатели (дескрипторы)

Не успели мы с Вами вспомнить о дескрипторах, как они — тут как тут. Напомню, что дескрипторы — это служебные переменные, описывающие информацию, заключенную в image-table-fit файлах MIDAS. Они содержатся в "шапках" этих структур и

полностью их описывают. Также они могут содержать некоторую служебную информацию: например, кем и когда получены эти данные, статистические характеристики, историю преобразований, которым они подверглись.

Из этого следует, что дескрипторы можно разделить на две группы — обязательные (есть всегда, ибо именно они описывают структуру изображений и таблиц) и остальные. Так, к стандартным дескрипторам для изображений (смотрите `HELP [Descr]`) относятся:

NAXIS	—	размерность
NPIX	—	число точек
START	—	начало на каждой из осей
STEP	—	шаг на каждой из осей
IDENT	—	имя
CUNIT	—	величина $[(NAXIS+1)*16]$
LHCUTS	—	min, max, vis_min, vis_max

В общем случае дескриптор — это вектор (набор) чисел или строк с максимальным количеством элементов 32767. Стандартные дескрипторы имеют длину имени, не превышающую 8 символов. Все остальные могут иметь длину имени до 72 символов. Дескрипторы бывают четырех типов: **integer(i)**, **real(r)**, **double(d)**, **character(c)**. Смешанные типы дескрипторов не поддерживаются. Общая форма обращения к дескриптору:

Имя/Тип/начало/сколько_элементовN элемент1...элементN.

Пример :

`IDENT/C/1/60` — с 1-го по 60-ый элемент character дескриптора IDENT.

`STATIST/r/1/5 5.,6.,2.4,0.,1.2` — с 1-го по 5-ый элемент real дескриптора STATIST.

`STATIST/r/5/6 5.,6.,2.4,0.,1.2` — с 5-го по 11-ый элемент real дескриптора STATIST.

Для полного определения дескриптора требуется еще имя изображения или таблицы, в котором он находится.

Пример :

`WRITE/DES testima ident/c/1/60 "Mkn 996"` — записать в дескриптор IDENT изображения **testima.bdf** имя галактики.

`WRITE/DES testima statist/r/1/5 5.,6.,2.4,0.,1.2` — записать в real дескриптор STATIST изображения **testima.bdf** ряд чисел.

ВНИМАНИЕ : При чтении и записи дескрипторов для табличных и *fit*-файлов необходимо указывать соответствующие расширения *".tbl"* и *".fit"* в имени файла. Для изображений этого не требуется, "по умолчанию" берется изображение с расширением *".bdf"*.

ВНИМАНИЕ : Самый "заумный" тип дескриптора — это массив строк. Определить его можно так:

`WRITE/DESCR testima cmore/c*5/1/10 abcde all` — определение дескриптора *cmore* как массива из 10 строк по 5 символов в каждой и присвоение

значения "abcde" им всем.

Это аналог, например, "CHARACTER CMORE(10)*5" в FORTRAN или "char cmore[10][5]" в С. Очень удобен при создании собственного пакета.

Создается дескриптор командой WRITE/DES. Если он существует, то его можно дописать (скажем, состоял из 5-ти элементов, а увеличился до 10-ти).

Все команды работы с дескрипторами имеют квалификатор ... /DESCR: READ/DESCR, SHOW/DESCR, WRITE/DESCR, DELETE/DESCR

При преобразовании MIDAS-файла данных (изображение, таблица) в FITS-формат значения дескрипторов заносятся в ASCII шапку FITS-файла.

1.3. Ключевые слова

Ключевые слова - это глобальные переменные в MIDAS. Часть из них зарезервирована для хранения системных параметров MIDAS. Остальные используются для временного хранения данных, результатов выполнения программ, передачи данных между программами.

Работа с ключевыми словами очень сильно похожа на работу с дескрипторами. В общем случае ключевое слово — это вектор (набор) чисел или строк с максимальным количеством элементов 32767. Ключевые слова могут иметь длину имени до 8 символов. Ключевые слова бывают четырех типов: **integer(i)**, **real(r)**, **double(d)**, **character(c)**. Смешанные типы ключевых слов не поддерживаются. Общая форма обращения к ключевому слову:

Имя/Тип/начало/сколько_элементовN элемент1...элементN.

Пример :

WRITE/KEY ident/c/1/60 "Mkn 996" — записать ключевое слово.
WRITE/KEY statist/r/1/5 5.,6.,2.4,0.,1.2

Замечание : "Заумный" тип ключевого слова — массив строк — также существует. Определить его можно так:

WRITE/KEY cmore/c*5/1/10 abcde all — определение ключевого слова *cmore* как массива из 10 строк по 5 символов в каждой и присвоение значения "abcde" им всем.

Никогда за все годы моей работы с MIDAS я их не использовал. И Вам не советую. Но мало ли что...

В отличие от дескрипторов, ключевые слова не добавляются. Это значит, что если Вы уже определили размер ключевого слова, то для увеличения размера надо ключевое слово удалить и создать новое, с тем же именем, но большего размера.

Все команды работы с ключевыми словами имеют квалификатор ... /KEYWORD: READ/KEY, SHOW/KEY, WRITE/KEY, DELETE/KEY, COMPUTE/KEY, COPY/KEY

Существуют стандартные ключевые слова, которые создаются, когда MIDAS стартует. Их полезно знать. Для ознакомления с ними смотрите команду HELP [Key].

Замечание : Очень полезна команда COMPUTE/KEY, которая позволяет производить математические операции с ключевыми словами, а также содержит достаточно большое число дополнительных функций.

1.4. Элементы в таблице

Наиболее часто таблицы используются при анализе полученных материалов, поскольку позволяют хранить неоднородные данные (каждая колонка может содержать свой собственный формат).

Файл таблицы состоит из шапки, содержащей дескрипторы и тела таблицы (если считать, что обыкновенная таблица — это двумерный массив данных, то MIDAS позволяет работать с трехмерными таблицами, и в этом случае каждая ячейка таблицы является, в свою очередь, одномерным массивом данных). Имеется очень большой набор команд, работающих с таблицами, и подавляющее большинство этих команд имеют квалификатор .../TABLE: READ/TAB, SHOW/TAB, WRITE/TAB, DELETE/TAB, COMPUTE/TAB, COPY/TAB Таблица состоит из элементов или ячеек. Доступ к отдельному элементу осуществляется по имени таблицы, имени (номеру) колонки и номеру строки, в которых этот элемент находится. Адресация к колонке производится по имени (:COLUMN_NAME) или номеру (#COLUMN_NUMBER). Адресация к строке производится или по номеру (@ROW_NUMBER) или с использованием референс-колонки, определяемой командой SET/REFCOLUMN.

Пример :

READ/TABLE mytab #3 @10..20 — прочитать содержимое с 10 по 20 элемент колонки номер 3 таблицы **mytab.tbl**.

READ/TABLE mytab :MAGNITUDE @10..20,100..200 — прочитать содержимое с 10 по 20 элемент, а также с 100 по 200 элемент колонки с именем MAGNITUDE таблицы **mytab.tbl**.// WRITE/TABLE mytab :MAGNITUDE @10 13.5 — записать значение.

Замечание : Внимательно изучите команду COMPUTE/TABLE. Это одна из основных команд для работы с таким типом данных. Содержит очень большое число дополнительных функций, также позволяющих преобразовывать колонки одного типа в другой.

1.5. Работа с каталогами

Каталоги используются обычно для работы с группами файлов одинаковой структуры: изображениями или таблицами и т. д. Для работы с различными группами используются различные квалифиликаторы:

.../ICAT — для работы с изображениями;

.../TCAT — для работы с таблицами;

.../FCAT — для работы с Fit-файлами;

.../ACAT — для работы с ASCII-файлами.

Для операций с каталогами существует достаточно ограниченное число команд: CREATE/xCAT, SET/xCAT, CLEAR/xCAT, READ/xCAT, ADD/xCAT, SUBTRACT/xCAT.

Однако необходимо помнить о команде EXECUTE/CATALOG, которая и позволяет, собственно, после создания каталога выполнить определенную операцию для всех файлов из каталога (пример работы с этой командой смотрите в Лекции 2).

1.6. Еще быстрее...

При использовании прямого обращения к элементам изображений, таблиц и т.д. в командных файлах помните, что при выполнении команд типа **READ/...** **WRITE/...** монитор вызывает командный файл, который, в свою очередь, вызывает исполняемую программу. Это замедляет работу, особенно если Вы исполняете подобные команды часто. Для ускорения работы используйте нижеприведенный формат обращения к данным структурам. При таком обращении команда будет выполняться самим MIDAS-монитором:

frame,descr = value — запись ОДНОГО значения в дескриптор изображения или таблицы;

frame[x,y,z = value] — запись значения ОДНОГО пикселя в изображение;

table,column,row = value — запись ОДНОГО табличного элемента.

Пример :

```
WRITE/DES testima statist/r/6/1 5. или:  
testima,statist(6) = 5.  
WRITE/IMAGE testima [@1,@1:@1,@1] 1.0 или:  
testima[@1,@1] = 1.0  
WRITE/TABLE mytable :RADVEL @20 13.5 или:  
mytab,:RADVEL,20 = 13.5
```

ВНИМАНИЕ : Пробелы при написании этих команд обязательны!

2. Обзор MIDAS-команд по группам и назначениям

Весь огромный набор команд MIDAS можно примерно разделить на следующие группы (или категории):

- Primitive commands — команды нижнего уровня. Они в свою очередь разделяются на команды ядра (команды, исполняющиеся самим MIDAS-монитором). Например, **SET/MIDAS**, **DELETE/IMAGE** и базовые команды, исполняющиеся из командных файлов (например, **PLOT/TABLE**).
- Application commands — команды всех дополнительных пакетов. В свою очередь эти пакеты разделяются на три категории:
 - Applic — команды из дополнительных пакетов, расширяющие стандартный набор команд (например, **COMPUTE/FIT**);
 - Stdred — команды из пакетов, созданных для проведения первичной редукции данных, полученных на различных приборах (например, **EXTRACT/ECHELLE**);
 - Contrib — команды из пакетов, созданных для проведения дальнейшего анализа данных (например, **SEARCH/INVENTORY**).

- Procedure control commands — командный язык монитора, используемый при написании командных файлов (например, DEFINE/PARAMETER).

Замечание : Знать (хотя бы приблизительно), какие команды к какой категории относятся, чрезвычайно полезно, особенно при написании командных файлов — это может ускорить время выполнения Вашей программы в несколько раз. А найти эту информацию можно в тех трех томах MIDAS документации, о которых я уже упоминал ранее.

Теперь я попытаюсь объединить Primitive commands в группы по назначению. При этом, я совсем не буду упоминать группы команд, необходимых для ввода-вывода и работы с image и графическими дисплеями, поскольку я рассматривал их в предыдущих лекциях.

Итак, группы по назначению:

Работа с координатами изображения — набор программ для считывания координат изображения или определения центра:

```
CENTER/GAUSS, CENTER/MOMENT  
GET/CURSOR, GET/GCURSOR
```

Координатные преобразования изображений — различные программы преобразований: вращения, извлечение части и помещение новой части в изображение, переход к новому шагу и т.д.

```
ALIGN/IMAGE, EXTRACT/IMAGE, FLIP/IMAGE, TRANSPOSE/IMAGE  
GROW/IMAGE, INSERT/IMAGE,  
REBIN/II (liner, polinom, exp, ..., user define) - только 1D изображения  
REBIN/LINEAR (start, step) - и 2D тоже  
REBIN/ROTATE (CENTER/..., ALIGN/IMA)  
REBIN/SPLINE,  
REBIN/IT (liner, polinom, exp, ..., user define) - только 1D изображения  
ROTATE/CLOCK, ROTATE/COUNTER_CLOCK, ROTATE/1DIM  
XCORR/IMAGE
```

Арифметика — программы выполнения арифметических операций над изображениями. Сюда я также отношу и усреднение нескольких изображений в одно.

```
AVERAGE/AVERA all pixels in the subframe
```

```
AVER/COLUMN, AVER/ROW
```

```
AVERAGE/IMAGES (aver,min,max,med)
```

```
AVER/KAPPA, AVER/WEIGHT (COMP/WEIGHT)
```

```
COMP/ROW, COMP/COLUMN
```

```
COMP/IMAGE, COMP/PIXEL
```

Функции: SQRT(a), EXP(a), EXP10(a), LN(a), LOG10(a), SIN(a) ASIN(a), COS(a), ACOS(a), TAN(a), ATAN(a), INT(a), ABS(a), ATAN2(a1,a2), MAX(a1,a2), MIN(a1,a2) и MOD(a1,a2)

ВНИМАНИЕ : Помните про отличие COMP/IMAGE от COMP/PIXEL!

Фильтрация — программы проведения фильтрации различного рода по всему изображению или по его части.

```
CONVOLVE/IMA, DECONVOLVE/IMA (with psf)
CREATE/FILTER
FFT/IMAGE, FFT/INVERSE, FFT/POWER, FFT/FREQ
FILTER/COSMIC, FILTER/DIGITAL, FILTER/GAUSS,
FILTER/MAX, FILTER/MIN, FILTER/MEDIAN,
FILTER/SMOOTH, FILTER/ADAPTIV
```

Image creation and extraction — программы для создания новых изображений различными способами, в том числе и путем извлечения различных частей из уже существующих изображений.

```
CREATE/IMAGE, CREATE/RANDOM (gauss, exp, pol, poisson ...)
EXTRACT/IMA, EXTRA/CURSOR, EXTRACT/SLIT
EXTRACT/CTRACE, EXTRACT/RTRACE, EXTRACT/LINE (диагональ),
EXTRACT/TRACE, EXTRACT/REFERENCE_IMAGE, EXTRACT/ROTATED_IMAGE
```

Преобразование пиксельных значений — программы модификации пиксельных значений. Модификация производится либо в областях, отмеченных курсором, либо на основе выбранных критериев.

```
FIT/FLAT_SKY, ITF/IMAGE,
MODIFY/CURSOR, MODIFY/GCURSOR
MODIFY/PIXEL, MODIFY/AREA
REPLACE/IMAGE
REPLACE/POLYGON - работает только для полигонов.
```

ВНИМАНИЕ : Хотелось бы специально выделить программу REPLACE/IMAGE. Разберитесь с ее использованием: это должно облегчить Вашу жизнь при проведении обработки или анализе. Данная программа предназначена для создания изображений-масок. Имеет тысячу других приложений.

Пример :

```
REPL/IMA input out <,100=0.
REPL/IMA input out <,100=ref
REPL/IMA input out <,100=ref+2.5,d
```

Numerical values of image pixels — программы для получения различных характеристик анализируемых изображений.

FIND/MINMAX, INTEGR/APERTURE, INTEGR/LINE
 MAGNITUDE/CIRCLE, MAGNITUDE/RECTANGLE
 STATIST/IMAGE

Fitting — программы приближения имеющихся распределений различными функциями.

COMPUTE/FIT, COMPUTE/FUNCTION,
 EDIT/FIT, FIT/IMAGE, FIT/TABLE, READ/FIT, SET/FIT
 SHOW/FIT, SELECT/FIT
 REGRESSION/POLYNOMIAL, SAVE/REGRESSION

Работа с таблицами Я хотел бы еще более подробно остановиться на работе с таблицами. С моей точки зрения, для эффективной работы с ними Вам необходимо знать следующие вещи:

1. Каждая таблица содержит две дополнительных колонки — первая — это просто номер строки (называется :SEQUENCE или :SEQ), а вторая называется SELECT или SEL и равна или 0 или 1 для каждой строки, которая удовлетворяет/не удовлетворяет набору условий соответственно. Обратите внимание, что хоть формально это и колонка, но пишется без ":". Знание этого момента, а также команд SELECT/TAB и COMPUTE/TAB позволяет творить с таблицами чудеса.

ВНИМАНИЕ : Учитите, что команда COMPUTE/TABLE в конце удаляет все выделения (практически работает как команда SELECT/TAB Table All).

Пример :

SELECT/TAB CFA :MAG.lt.15.and.:VEL.gt.1000 — выделяем в таблице, содержащей каталог галактик, все объекты ярче 15^m со скоростями больше 1000 км/сек.

COMP/TAB CFA :Z = min(:MAG,:R) + (1-SEL)*:VEL — создаем новую колонку, в которую записываем для каждой строки минимальное значение из колонок :MAG и :R и добавляем к нему значение скорости для выделенных предыдущей командой строк (в записанном мной виде выделенные значения будут умножены на 1).

COMP/TAB CFA :SURF_BRIGHT = 2.5 * log10(10**((0.4*:MAG)/(:D**2))) — считаем для всех галактик среднюю поверхностную яркость.

COMP/TAB mytab :name = concat(:star,tochar(:SEQ)) — делаем колонку: имя звезды+номер строки таблицы, в которой она есть.

2. Если в поле таблицы нет значения, это называется NULL-величина. Обозначается как "*". На эту же величину будут заменяться все результаты некорректных операций (LOG от -1, например).

3. Есть набор команд для копирования колонок из таблицы в таблицу, а также преобразования структуры "таблица" в структуру "изображение" и наоборот: COPY/KT, COPY/TK, COPY/TT, COPY/TI, COPY/IT....

ВНИМАНИЕ : Разберитесь с командой SET/REFCOLUMN в этом месте. Она позволяет то, чего не позволяет делать просто знание о колонке :SEQ.

4. Таблицы в MIDAS можно интерактивно редактировать. Редактор таблиц вызывается командой EDIT/TAB. В редакторе надо знать волшебные последовательности:

<CTrl/Z> — выход в командный режим
<CTrl/Z>q<Enter> — выход без записи
<CTrl/Z>e<Enter> — выход с записью
<CTrl/Z>help<F1><F1> — выдача списка функций

После выхода в командный режим можно пользоваться набором встроенных функций. Вот их список:

```
keypad  line   screen  word    _last_column   change_format   advance
bottom_of_table backup top_of_table  create_column  delete_column  page
command section right_page   left_page      _delete_word   _undel_word
_delete_character _undel_character _next_line    row   gold
next   find     _delete_line  _undel_line   _arrow   _bad_key   tabular
help   show    sort      status   Function
```

5. И еще набор команд, существующих для таблиц:

```
COMPUTE/HISTOGRAM, COMPUTE/REGRESSION
REGRESSION/LINEAR, REGRESSION/POLY,
SAVE/REGRESSION, COMPUTE/TAB, CONVERT/TAB,
INTERPOLATE/TT, INTERPOLATE/TI, INTERPOLATE/IT,
REBIN/TT, REBIN/TI, REBIN/IT          (1D)
MERGE/TABLE, JOIN/TABLE, NAME/COLUMN,
SELECT/TAB, SORT/TABLE, STATIST/TABLE
```

2.1. Не забывайте делать HELP/APPL

Помните про то, что всегда существует некий дополнительный набор командных файлов (так называемые applications), которые написаны другими пользователями и постоянно пополняются. Эти командные файлы иногда бывают очень полезны и вызываются как **Ca Имя_программы**. Их список можно получить, используя HELP/APPL, а полный набор параметров для каждой, — используя HELP/APPL Имя_программы. Ниже я привожу список этих программ, существующий в версии 98NOV с кратким описанием того, что данная команда выполняет.

asload — назначение дисплейного окна и визуализация в нем изображения;

areatable — построение таблицы, содержащей определение областей заданного размера, равномерно покрывающих заданное изображение;

autocuts — определение "хороших" уровней для визуализации изображения;

buildcube — построение куба данных из заданного набора двумерных изображений;

catalsync — проверка набора дескрипторов всех изображений/таблиц данного каталога этих файлов;

checker — создание изображения хитрого содержания;

checkname — проверка имени MIDAS-файла на наличие некорректных символов;

convolfft — свертка или корреляция изображения с изображением с использованием быстрого преобразования Фурье;

copyfile — копирование ASCII файлов по некоторым правилам;

diffdsc — сравнение заголовков двух изображений;

diffima — сравнение пикселей в двух изображениях;

dscedit — редактирование дескриптора;

edges — нахождение краев с использованием фильтров Sobel или Prewitt;

expand — увеличение размера изображения;

fastcomp — быстрое вычисление функций ln, log10 и sqrt от изображения;

func2d — создание изображений, соответствующих любой двумерной функции;

fftfilt — фильтрация изображений в Фурье плоскости;

gammacorr — создание "хитрой" ИТТ-таблицы;

getcoord — пересчет пиксельных координат в мировые (и наоборот) для заданного изображения;

histogram — построение изображения по гистограмме от изображения;

matconst — определение часто используемых математических констант;

normalize — нормализация входного изображения (вписывание значений в диапазон [0.,1.]);

outima — преобразование MIDAS-изображений в другие форматы (требует настройки при инсталляции MIDAS);

packdsc — убиение "пустого" места из заголовка изображения;

perspec — построение и визуализация 3-х мерной перспективы от куба данных;

polyrep — отметить полигон на визуализированном изображении и заменить его на константу;

plottab — нарисовать содержание LUT или ITT-таблиц;

replace — расширенный вариант команды REPLACE/IMAGE;

resetback — восстановить соединение с фоновым MIDAS;

sanchek — проверить изображение на наличие NaN или Inf (медленная процедура);

scale — промасштабировать изображение в данной области величин;

scaler — промасштабировать изображение по алгоритму команды LOAD/IMAGE, но сделять это на диске.

scanima — сделать из 2D-изображения 1D-изображение по определенным законам (иногда очень полезна);

sharpen — выделение краев изображения с использованием специального маскирования;

showfonts — показать шрифты, которые могут быть использованы в MIDAS;

showmidas — показать текущие MIDAS-установки;

slicube — извлечь 2D плоскости из 3D куба данных с определенным шагом;

swap4 — переставить 4 квадранта изображения по диагонали;

vertlut — нарисовать LUT-таблицу вертикально и надписать уровни;

zperspec — построение и визуализация перспективной проекции двумерного изображения.

3. Работа с контекстами

Последнее, что хотелось бы описать в этом обзоре различных групп команд — работа с контекстами. MIDAS-контекст — это пакет программ, внешний по отношению к монитору. Контекст можно "загрузить" — выполнить команду SET/CONTEXT *имя_контекста*, после которой все команды этого пакета становятся доступными для исполнения (также становится доступным HELP на эти команды) и "выгрузить" — исполнить команду CLEAR/CONTEXT *имя_контекста*.

Насколько я понимаю, было две основных идеи создания контекстов:

1. Контекст — это набор программ, которые создаются любым человеком, а после этого "встраиваются" им в MIDAS.
2. Пользуясь контекстами, производится первичная редукция различного рода данных, в результате которой эти данные сводятся к основным структурам MIDAS. После чего, опять-таки пользуясь различными контекстами, можно производить анализ полученных и обработанных данных.

Ниже я очень коротко охарактеризую пакеты, доступные в настоящее время в MIDAS:

Applications — дополнительные пакеты, расширяющие стандартный набор команд;

Plot — программы, работающие с графикой, которые не относятся к базовым программам: (1) Апертурная интеграция; (2) Вычисление звездной величины; (3) Вычисление размера апертуры в координатах экрана; (4) Интерактивная идентификация линий.

Statist — статистические тесты таблиц: сравнение эмпирического распределения с теоретическим распределением, сравнение независимых выборок данных, измерение корреляции выборок и т.д.;

Table — Различные дополнительные утилиты для работы с таблицами, в частности, разбиение данных в столбцах на интервалы, усреднение колонок;

FIT — пакет для приближения распределений с использованием набора как стандартных функций, так и функций, определяемых пользователем. Приближаются как табличные данные, так и изображения, как одномерные, так и двумерные распределения.

Stdred — пакеты, созданные для проведения первичной редукции данных, полученных на различных приборах;

CCDRED — пакет для проведения первичной редукции CCD-изображений. При полной настройке работает в автоматическом режиме.

CCDTEST — предназначается для проведения регулярного тестирования CCD-приемников. Содержит следующие тесты:

- (1) тест BIAS-изображений;
- (2) вычисление горизонтальной и вертикальной эффективностей переноса;
- (3) тест DARK-изображений;
- (4) проверка эффективности затвора;
- (5) тест линейности приемников.

DO — организация данных. В частности, используется при организации базовых таблиц при проведении первичной CCD-редукции пакетом **CCDRED**;

ECHELLE — обработка эшелле-спектров. Пакет содержит ~ 40 команд.

ECHELLEC — старый вариант обработки эшелле-спектров. Пакет содержит команды обработки эшелле-спектров, полученных с двумерным счетчиком фотонов в качестве приемника. Программы были разработаны и встроены в MIDAS с участием автора этих лекций (Kniazev, Shergin & Lipovetsky, 1992).

IRAC2 — обработка данных IRAC2;

IRSPEC — пакет для обработки спектров, получаемых со спектрографом IRSPEC (IR спектрограф);

LONG — стандартная обработка спектров, полученных с модой "длинная щель". Пакет содержит ~ 20 команд. Шаги: исправление двумерного спектра за геометрические искажения, калибровка длин волн и исправление за дисторсию вдоль щели для изучения особенностей в спектрах протяженных объектов, исправление за кривую спектральной чувствительности и т.д.;

MOS — обработка данных мульти-объектного спектрографа. Пакет содержит ~15 команд для калибровки, построения дисперсионных кривых, линеаризации данных и т.д.;

OPTOPUS — обработка данных, полученных на фиберном спектрографе;

PISCO — полная обработка поляриметрических данных, полученных с поляриметром PISCO;

QC — пакет контроля качества изображений в течение наблюдений: анализируется качество по времени изображений BIAS, DARK и научных;

SPEC — набор программ, необходимых для работы с одномерными спектрами. Эти команды обычно используется в других спектроскопических пакетах, например, LONG, ECHELLE. Пакет содержит ~10 команд для построения дисперсионной кривой, исправления за кривую спектральной чувствительности, построения кривой спектральной чувствительности, коррекции за ripple-эффект и т.д.;

Contrib — пакеты, созданные для проведения дальнейшего анализа данных;

ASTROMETRY — пакет для проведения астрометрии. MIDAS версия пакета состоит из трех основных шагов: (1) чтение измерений положений стандартных звезд и вычисление коэффициентов преобразования; (2) редактирование таблицы стандартных звезд с целью удаления плохих измерений; (3) пересчет измеренных координат с использованием коэффициентов преобразования.

CLOUD — моделирование особенностей межзвездного поглощения. Модель представляется в виде полиномиального континуума, на который можно наложить эмиссионные и/или абсорбционные линии;

DAOPHOT — получение точной фотометрии и астрометрических позиций звездных объектов. Пакет содержит следующие шаги:

- (1) нахождение всех звездных объектов на изображении;
- (2) апертурная фотометрия для всех выделенных объектов;
- (3) построение точечной функции рассеяния для изображения, с использованием одной или нескольких звезд;
- (4) вычисление точных позиций и величин всех найденных звезд.

GEOTEST — программы для создания тестовых изображений разных видов;

HST — пакет для преобразования данных, полученных на космическом телескопе Хаббла во внутренний формат MIDAS.

IMRES — программы восстановления изображений (алгоритмы, применяемые для восстановления изображений в StScI);

INVENTORY — выделение объектов разного типа на изображениях и классификация этих объектов. Функционально пакет состоит из трех шагов: поиск всех объектов на изображении, анализ всех объектов на изображении и классификация всех объектов.

IUE — работа с данными, полученными со спутником IUE.

LYMAN — пакет для проведения многокомпонентного разложения межзвездных абсорбционных линий, который может использоваться для анализа спектров высокого разрешения.

MVA — статистический анализ многомерных выборок. Содержит:

- (1) Метод главных компонент;
- (2) Определение набора неперекрывающихся кластеров;
- (3) Иерархическую кластеризацию;
- (4) Линейный дискриминантный анализ;
- (5) Множественный дискриминантный анализ; (6) Дискриминантный анализ по К-ближайшим соседям.

PEPSYS — планирование фотометрических наблюдений и полная обработка фотометрических наблюдений (не только CCD-фотометрия, но также наблюдения и обработка электрофотометрических наблюдений). Учитывается информация о телескопе и инструменте, полученные величины преобразуются в стандартную систему (если это возможно), оцениваются ошибки получаемых величин;

ROMAFOT — получение звездной фотометрии для полей с большой плотностью объектов (например, фотометрия звезд шаровых скоплений). Незвездные объекты автоматически удаляются;

SURFPHOT — набор программ, полезных при проведении поверхностной фотометрии. Содержит: (1) программы проведения изофот и фона; (2) поиск позиционного угла и наклона галактики; (3) анализ азимутальных профилей; (4) коррекция изображений, искаженных дисторсией; (5) коррекция плохих пикселей; (6) нормализация изображений; (7) восстановление изображений;

TSA — пакет для анализа временных рядов. Содержит 15 команд;

WAVELET — wavelet преобразование. Данное преобразование извлекает из изображений информацию об их пространственных и частотных свойствах. Очень полезно для описаний иерархических структур. В пакет входят определенные алгоритмы (pyramidal algorithms, Mallat's algorithm with bi-orthogonal filters, algorithms using the FFT, ...), программы визуализации и две программы работы с изображениями (фильтрация и сравнение).

SAO — пакеты, созданные как в Специальной Астрофизической Обсерватории, так и в других обсерваториях, но встроенные в MIDAS в САО РАН. Не являются стандартными в MIDAS, но доступны в САО РАН.

AIP — пакет проведения адаптивной фильтрации и поверхностной фотометрии изображений. Содержит также топологическую работу с масками. Программы созданы G.Richter в Астрофизическом Институте Потсдама (Германия). Встроены в MIDAS автором (Князев 1997). Доступны только в САО РАН или по согласованию с автором программ (gmrichter@aip.de).

SCANNER — обработка данных, полученных с одномерным счетчиком фотонов (Князев 1994; Князев 1997);

NICE — наблюдения с CCD (Князев, Шергин 1995). Позволяет проводить сбор **различных** типов данных (спектральных в модах “длинная щель” и “эшелле” и фотометрических) на **различных** телескопах (БТА и Цейс-1000) и параллельную сетевую буферизацию данных. Пакет построен таким образом, что позволяет:

1. Легко менять моды наблюдений (спектральные или фотометрические) в течение наблюдательной ночи ;
2. Быть настроенным на новые типы CCD;
3. Встраивать новые системы сбора;
4. Поддерживать FLEX-формат имен выходных данных (Кононов 1995), зависящий от систем сбора;
5. Производить сетевую буферизацию получаемых данных.

OBPR — пакет для обработки низкодисперсионных спектров, полученных с CCD в качестве приемника. Разработан автором (Kniazev et. al. 1996; Kniazev et al. 1997; Князев 1997). Содержит: программу цифрового фильтра подавления низких частот (Shergin et al. 1996), пакет поиска и выделения объектов по прямым снимкам, программы привязки спектрального изображения к прямому, программы извлечения одномерных спектров из двумерных спектральных снимков, полученных с объективной призмой, собственный пакет астрометрии и собственный пакет апертурной фотометрии.

EXAS — система редукции и анализа рентгеновских данных, полученных со спутником ROSAT.

И это конец четвертой сказки о MIDAS.

Список литературы

- Князев А.Ю.: Система управления и сбора данных спектрофотометра в Несмит-1 БТА. IV. Редукция сканнерных данных в MIDAS. — *Отчет CAO*, 1994, **223**, с.1–28.
- Князев А.Ю., Шергин В.С.: CCD наблюдения в среде MIDAS. Контекст NICE. Версия 1.0. — *Отчет CAO*, 1995, **239**, с.1–19.
- Князев А.Ю., Кандидатская диссертация, 1997, с.1–201
[\(http://precise.sao.ru/Laboratory/Dis_agn/index.html\)](http://precise.sao.ru/Laboratory/Dis_agn/index.html)
- Кононов В.К.: Унифицированный формат представления наблюдательных данных (FLEX-стандарт). — *Препринт CAO*, 1995, № 111Т, с.3–21.
- Kniazev A.Y., Shergin V.S., Lipovetsky V.A. (1992) MIDAS on small computers: reduction of low order echelle spectra. 4-th ESO/ST-ECF Data Analysis Workshop: 169–176
- Kniazev A.Yu., Salzer J., Lipovetsky V.A., Boroson T., Moody J., Thuan T., Izotov Yu.I., Herrero, J., & Frattare, L.: KISS: A New Digital Survey for Emission-Line Objects. — *in Proceeding of 179 symposium of IAU*, 1996, p.500-503.
- Kniazev A.Yu., Salzer J., Lipovetsky V.A., Boroson T., Moody J., Thuan T., Izotov Yu.I., Herrero, J., & Frattare, L.: KISS: A New Digital Survey for Emission-Line Objects. — *Newsletter Working Group on Wide field imaging, IAU Comission 9*, 1997, **9**, p.5–10.
- Shergin V.S., Kniazev A.Yu., & Lipovetsky V.A.: New family of non-linear filters for background subtraction of wide-field surveys. — *Astronomische Nachrichten*, 1996, **2**, с.95-100.

MIDAS Users Guide, 1995a, **Volume A**
MIDAS Users Guide, 1995b, **Volume B**