

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 **-p** позволяют работать в нескольких MIDAS одновременно.

Пример :

inmidas 00 или **inmidas 10**

inmidas xx (вы предполагаете работу без графики)

-p/-P/-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 **-p** или **-P**, 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 -p — 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 предпочитаю 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 -p — 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 "y": 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 $\tilde{\text{midwork}}$. 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 комментарий
```

- 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 PLOT=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 перевызвать, 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 перевызвать 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 заголовки 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 исполнить 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 послед-
них двухнедельных расчетов, быстро убеждается в том, что последст-
вия действительно 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 Вашими 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/GRAPHICS`, `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 droups	→	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 scanima.

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 TEX/L^ATEX, 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 САО".

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/GRAPH postscript`. В текущей директории появится файл с именем `postscript.ps`. Переименуйте его, чтобы его имя было Вам понятно. Это и есть искомый PostScript с Вашим рисунком.

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

```
ASSIGN/GRAPH postscript
PLOT/TABLE ...
ASSIGN/GRAPH 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}
```

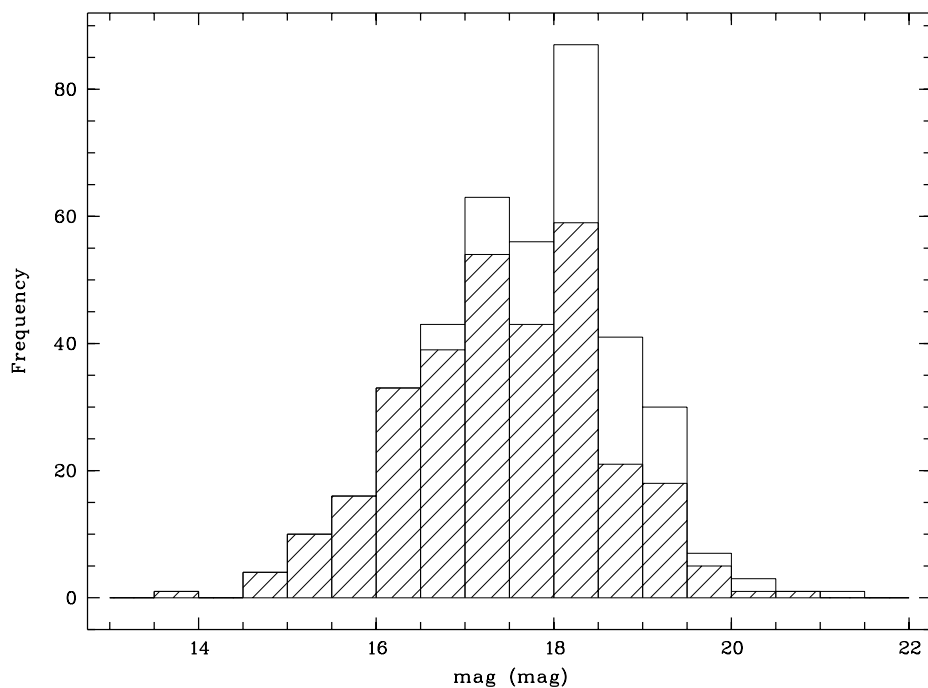


Рис. 1. Эффективность обзора Гамбург/CAO по поиску эмиссионных галактик

```

\begin{figure}[hbt]
  \centering{
    \hspace*{-1.cm}
    \vbox{\psfig{figure=HSS.ps,width=14cm,angle=-90}}\par
  }
  \caption{Эффективность обзора Гамбург/CAO 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 картинку в один рисунок. Результат виден на Рисунке 2.

```

\documentclass{article}
\usepackage{psfig}

\begin{document}

\begin{figure}[hbt]
\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}

```

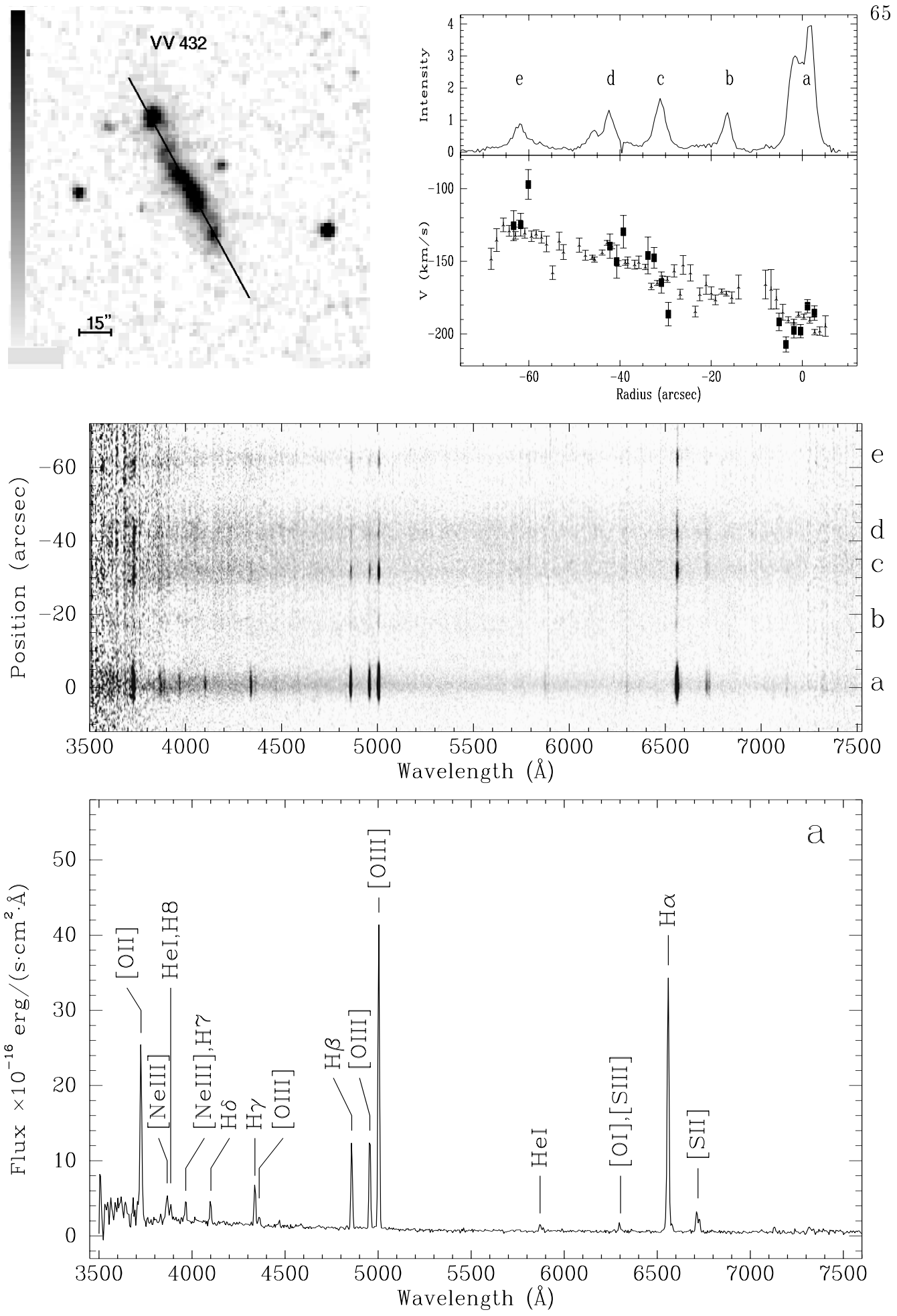


Рис. 2. Сводная информация о галактике VV432

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 означает, 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, переживавшей 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 процесс-сына, имея MIDAS-монитор как процесс-отца. Сын благополучно выполняет команду перехода, о чем и сообщает отцу. И завершается. А отец за сына не в ответе — он остался в старом состоянии. Так вот, чтобы встряхнуть 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:

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

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".