GDL (IDL) für Einsteiger

Die Programmiersprache GDL - eine freie Variante von IDL - verfügt über Bibliotheksfunktionen, die speziell für die Datenanalyse von astrophysikalischen Daten ausgelegt sind und auch Routinen für das Lesen und Schreiben von FITS-Dateien bereithält.

Der Aufruf des Interpreters erfolgt aus der Konsole mittels:

```
gdl
```

Man kann nun GDL-Programmcode zeilenweise eingeben, der vom Interpreter sofort ausgeführt wird.

Eleganter ist das Schreiben eines Programms für IDL in Dateiform. Dazu öffnet man - außerhalb von GDL - eine Datei (Endung sollte `.pro` heißen) mit einem geeigneten Texteditor (*kate*, *emacs*, *nedit*, *vi* ...). Hat man das Programm (z.B. `meinprog.pro`) abgespeichert, wird es in GDL mit dem Kommando

```
.compile meinprog.pro
```

kompiliert, d.h. übersetzt. In der ersten Zeile der Datei steht nach dem Schlüsselwort *pro* der (GDL-interne) Name des geschriebenen Programms, z.B.

```
pro prog1
```

Nach dem Kompilieren des Programms wird es durch Aufruf des Programmnamens

```
prog1
```

ausgeführt.

Syntax

Achtung, anders als in der normalen Linuxkonsole, werden innerhalb von GDL Groß- und Kleinbuchstaben bei Kommandos nicht unterschieden. Das bedeutet die Zeilen

```
print, "Hallo Welt!"
PRINT, "Hallo Welt!"
```

und sogar

```
PrInT, "Hallo Welt!"
```

liefern alle dasselbe Resultat. Innerhalb von Strings, wie hier “Hallo Welt!”, und Dateinamen wird natürlich zwischen Groß- und Kleinbuchstaben differenziert.

Grundlagen

	Kommando
Aufruf des Interpreters:	`gdl`
Aufruf der GDL-Hilfe:	`?`
Anzeigen aller Variablen:	`help`
Kompilieren eines eigenen Scripts:	`.compile meinprogramm.pro`
Ausführen des Scripts in gdl:	`meinprogramm`
Beenden von GDL:	`exit`
Erste Zeile:	`pro meinprogramm`
Letzte Zeile (Ende des Programms):	`end`

FITS-Dateien

Einlesen eines FITS-Files in die Variable *bild1*:

```
bild1 = readfits('filename.fits')
```

Liegt das Bild nicht im aktuellen Verzeichnis muss man hier den ganzen (relativen) Pfad vor dem Dateinamen angeben (z.B. '../daten/filename.fits').

Je nach interner Rechnerarchitektur werden Daten, die mehr als ein Byte umfassen entweder in High- oder Lowbyteordnung abgespeichert. In seltenen Fällen kann es passieren, dass die Standardordnung zwischen Quellsystem und Auswertesystem nicht identisch ist, was sich in (scheinbar) unsinnigen Werten ausdrückt und sich leicht mit einem Blick auf das Bild der FITS-Datei ermitteln lässt. In solchen Fällen muss die Byteordnung konvertiert werden mittels:

```
byteorder, bild1
```

Abspeichern eines Arrays (hier *bild2*) als FITS-Datei:

```
writefits, 'neuerfilename.fits', bild2
```

Beachte: Alle FITS-Befehle benötigen die (frei verfügbare) Astrolib für GDL/IDL (siehe hier für eine Installationsanleitung]]).

Ein- und Ausgabe

Schreiben auf die Konsole

```
print, "Hallo ", namevar
```

Mittels Komma getrennt können beliebig viele weitere Werte oder Variablen angehängt werden. Es erfolgt am Ende ein Zeilenumbruch.

Schreiben in eine Datei:

```
openw, handle, 'dateiname.dat'
```

```
printf,handle,i,variable1,"Beispieltext",variable2
close, handle
```

handle muss dabei ein beliebiger, aber eindeutiger Integerwert sein. Alternativ bietet sich folgende Syntax an:

```
openw,out,'filename.dat',/get_lun
printf,out,i,variable1,"some text",variable2
close,out
free_lun,out
```

GDL wird in diesem Fall automatischen einen *handle* zuweisen (mittels */get_lun*). Das Kommando *free_lun* ist nötig um den *handle* anschließend wieder freizugeben.

Variablen und Felder

Indizierte Variablen beginnen mit dem Index 0. Ein Array mit den Indizes 0,1,…,MMAX hat somit MMAX+1 Einträge.

Dimensionierung eines zweidimensionalen Arrays für Integer-Zahlen durch:

```
variablenname = intarr(MMAX+1,NMAX+1)
```

In machen Fällen werden größere Zahlenwerte benötigt, als der Datentyp Integer hergibt ($i>2^{15}-1=32767$), in diesem Fall können Long- oder sogar Long64-Arrays (32 bit or 64 bit) verwendet werden:

```
variablenname = lonarr(MMAX+1,NMAX+1)
variablenname = lon64arr(MMAX+1,NMAX+1)
```

Für Fließkommazahlen (Float) lautet der Befehl;

```
variablenname = fltarr(MMAX+1,NMAX+1)
```

Für Fließkommazahlen mit doppelter Genauigkeit (Double) lautet der Befehl:

```
variablenname = dblarr(MMAX+1,NMAX+1)
```

Für Strings lautet der Befehl:

```
variablenname = strarr(MMAX+1,NMAX+1)
```

Die Anzahl der Einträge eines Arrays (hier *arrayvar*) kann über den Befehl

```
size(arrayvar,/dimensions)
```

ermittelt werden. Allerdings liefert `size` wiederum ein Array mit Informationen, von denen nur ein Wert die Anzahl der Einträge ist. Um diesen direkt zu ermitteln, sollte der Befehl ergänzt werden zu

```
anz = (size(arrayvar,/dimensions))(0)
```

Kontrollstrukturen

Schleifen-Syntax:

```
for j=n1,n2 do begin
  ...
endfor
```

Syntax für Bedingungen:

```
if a gt b then begin
  ...
endif
```

Zusätzliche Informationen bietet die GDL-Hilfe (Befehl: ?) sowie die GDL-Dokumentation.