

# ATK tähtitieteessä

Osa 4 - IDL input/output

19. syyskuuta 2014

# IDL - INPUT/OUTPUT-rutiinit

- ▶ IDL pystyy lukemaan ja kirjoittamaan monentyyppisiä tiedostoja, esim. FORTRAN ja C-kielten ohjelmien tulostusta.
- ▶ Käytössä on myös monipuoliset tulostuksen formatointikomennot.
- ▶ Pääperiaate:
  - ▶ Ennen lukemista tiedosto on avattava vapaana olevaan laitenumeroon. Komento on erilainen riippuen halutaanko tiedostoon kirjoittaa, vai vain lukea se.
  - ▶ Tiedosto pitää myös sulkea kun sen käsittely lopetetaan.

# IDL - Tiedoston avaaminen ja sulkeminen

- ▶ Tiedoston avaaminen vain lukemista varten laitenumeralle 1:

```
openr,1,'tiedostonimi'
```

- ▶ Tiedoston avaaminen kirjoittamista varten laitenumeralle 2, mahdollinen aiempi sisältö tuhoutuu:

```
openw,2,'tiedostonimi'
```

- ▶ Tiedoston avaaminen kirjoittamista ja lukemista varten:

```
openu,3,'tiedostonimi'
```

- ▶ HUOM! Samalla laitenumeralta voi olla vain yksi tiedosto avoinna kerralla. Tämän vuoksi on tärkeää sulkea tiedostot niiden käsittelyn jälkeen.

- ▶ Tiedoston sulkeminen laitenumeroista 1:

```
close,1
```

# IDL - Lukeminen ja kirjoittaminen (formatoitu)

- ▶ Normaalin formatoidun tekstitiedoston käsittelyssä käytetään komentoja, jotka vastaavat IDL-teksti-ikkunasta lukua ja kirjoitusta (read, print):
  - ▶ `readf,1,muuttuja1,muuttuja2,...` – lukee laitenumeralta 1.
  - ▶ `printf,1,muuttuja1,muuttuja2,...` – kirjoittaa laitenumeralle 1.
- ▶ Huomaa siis, että laitenumero joudutaan aina antamaan tiedostoja käsiteltäessä.

# IDL - Lukeminen ja kirjoittaminen (formatoitu)

- ▶ Esimerkiksi jos meillä on yksinkertainen kaksirivinen tiedosto nimeltä *testi*:

```
10  20  30
40  50  60
```

- ▶ se voidaan lukea seuraavasti:

```
IDL> openr,1,'testi'
IDL> readf,1,a,b,c
IDL> readf,1,d,e,f
IDL> close,1
```

- ▶ Tässä esimerkissä tiedoston jokainen alkio siis tallennetaan eri muuttujaan (a=10, b=20, c=30, d=40, e=50, f=60).

- ▶ Vastaavasti tiedoston kirjoittaminen:

```
IDL> openw,1,'testi'
IDL> printf,1,10,20,30
IDL> printf,1,40,50,60
IDL> close,1
```

# IDL - Lukeminen (formatoitu)

- ▶ Yleensä luettavat tiedostot eivät kuitenkaan ole lyhyitä. Tällöin voidaan ensin tarkistaa tiedoston rivien lukumäärä, ja käyttää FOR- tai WHILE-rakennetta, tai suoraan taulukkoon lukua tiedoston käsittelyssä.
- ▶ Oletetaan jälleen tiedosto nimeltä *testi*, joka sisältää x,y,z-arvoja muodossa:

x1	y1	z1
x2	y2	z2
...		
xn	yn	zn

- ▶ Luetaan tämä tiedosto nyt kaikilla tavoilla.

# IDL - Lukeminen (formatoitu)

► FOR-rakenteella:

```
;rivien lukumäärä muuttujaan n proseduurin file_lines avulla
n=file_lines('testi')
;luodaan taulukot muuttujille
x=fltarr(n) & y=x & z=x
;avataan tiedosto lukua varten
openr,1,'testi'
for i=0,n-1 do begin
;rivin luku temp-muuttujiin
  readf,1,temp1,temp2,temp3
;ja tallennetaan arvot taulukoihin
  x(i)=temp1
  y(i)=temp2
  z(i)=temp3
endfor
;suljetaan tiedosto
close,1
```

# IDL - Lukeminen (formatoitu)

► WHILE-rakenteella:

```
;rivien lukumäärä muuttuun n proseduurin file_lines avulla
n=file_lines('testi')
x=fltarr(n) & y=x & z=x
;avataan tiedosto lukua varten
openr,1,'testi'
i=0
;not eof(laitenumero) kertoo että jatketaan tiedoston loppuun
while not eof(1) do begin
;rivin luku temp-muuttujiin
    readf,1,temp1,temp2,temp3
;ja tallennetaan arvot taulukoihin
    x(i)=temp1
    y(i)=temp2
    z(i)=temp3
    i=i+1
endwhile
;suljetaan tiedosto
close,1
```



# IDL - Lukeminen (formatoitu)

► Suoraan taulukkoon:

```
;rivien lukumäärä muuttujaan n proseduurin file_lines avulla
n=file_lines('testi')
x=fltarr(n) & y=x & z=x
;avataan tiedosto lukua varten
openr,1,'testi'
;luodaan 3*n taulukko
temp=fltarr(3,n)
;rivin luku temp-tilukseen
readf,1,temp
;ja tallennetaan arvot taulukoihin
x=temp(0,*)
y=temp(1,*)
z=temp(2,*)
;suljetaan tiedosto
close,1
```

# IDL - Lukeminen (formatoitu)

- ▶ On myös tapauksia joissa tiedosto sisältää sekaisin lukuarvoja ja tekstiä. Esimerkiksi:

Tämän tiedoston ensimmäinen rivi on selitystä sisällöstä.

x1	y1	z1	huono
x2	y2	z2	hyvä
...			
xn	yn	zn	huono

- ▶ Tällöin tiedoston avaamisen jälkeen selitysrivi alusta voidaan lukea `readf`-komennolla, ennen kuin loppu tiedostosta luetaan.

# IDL - Lukeminen (formatoitu)

- ▶ Esimerkiksi edellisen esimerkkitiedoston voisi lukea seuraavalla koodinpätkällä:

```
;rivien lukumäärä ilman alun selitysriviä
n=file_lines('testi') - 1.
x=fltarr(n) & y=x & z=x & laatu=strarr(n)
openr,1,'testi'
roskaa='' ;tyhjä merkkimuuttuja
readf,1,roskaa ;luetaan selitysrivi
for i=0,n-1 do begin
    readf,1,temp1,temp2,temp3,temp4
    x(i)=temp1
    y(i)=temp2
    z(i)=temp3
    laatu(i)=temp4
endfor
close,1
```

# IDL - Kirjoittaminen (formatoitu)

- ▶ Huomaa, että ylläolevia esimerkkejä voi käyttää soveltaen myös tiedostojen kirjoittamiseen.
- ▶ Esimerkiksi:

```
x=findgen(10) & y=x/2. & z=x*2.  
n=n_elements(x)  
;avataan tiedosto kirjoitusta varten  
openw,1,'testi'  
for i=0,n-1 do begin  
;rivin kirjoitus tiedostoon  
  printf,1,x(i),y(i),z(i)  
endfor  
;suljetaan tiedosto  
close,1
```

# IDL - Lukeminen ja kirjoittaminen (formatoimaton)

- ▶ Formatoimaton tiedosto on koneen sisäisellä esitystavalla tallennettu tiedosto, jota ei pysty normaaleilla tavoilla käsittelemään.
- ▶ Formatoimattomia tiedostoja käytetään esimerkiksi kuvien tallentamiseen niiden kompaktiuden vuoksi. Niitä voidaan luoda myös esimerkiksi FORTRAN-kielellä.
- ▶ Tiedoston avaaminen ja sulkeminen tapahtuu samoilla komennoilla kuten aiemmin.
- ▶ Tiedostosta lukeminen ja siihen kirjoittaminen tapahtuvat komennoilla:
  - ▶ `readu,1,muuttuja1,muuttuja2,...` – laitenumeroista 1 lukeminen.
  - ▶ `writeu,1,muuttuja1,muuttuja2,...` – laitenumeroon 1 kirjoittaminen.

# IDL - Muotoilulauseet

- ▶ Käytettäessä aiempia print- tai printf-komentoja tulostukseen ilman muotoilulauseita, IDL tulostaa luvut koko tarkkuudellaan mikä on usein tarpeetonta.
- ▶ Tulostuksen muokkaamiseen voidaan käyttää `format='muotoilulause'` avainsanaa. Esimerkiksi:  

```
x=2.  
print,x ;tulostuu muodossa 2.00000  
print,x,format='(F6.2)' ;tulostuu muodossa 2.00
```
- ▶ Erilaisia format-määrittelyjä:
  - ▶ `Fn.k` – liukuluvuille, `n`=luvulle varattava tila, `k`=desimaalien määrä.
  - ▶ `En.k` – Eksponenttiesitys.
  - ▶ `In` – kokonaisluvuille, `n`=luvulle varattava tila.
  - ▶ `An` – merkkitieto, `n`=tulostettavien merkkien määrä.
  - ▶ `nX` – tyhjää tilaa, `n`=tyhjien merkkien määrä.

# IDL - SAVE ja RESTORE-komennot

- ▶ IDL:ssä erittäin käyttökelpoinen muuttujien tallennustapa on sen sisäisellä formaatilla tallennetut save-tiedostot. Niihin voi tallentaa mielivaltaisessa järjestyksessä kaikkia IDL:n data-muotoja (lukuja, tekstiä, ...).
- ▶ Tiedostoa palautettaessa ei tarvitse myöskään tietää sen sisältöä tai muuttujien kokoa, vaan ne kaikki palautetaan alkuperäisessä muodossa ja nimettyinä.
- ▶ Tallennus:  
`save,var1,var2,...,filename='tiedostonimi.save'`
- ▶ Antamalla avainsanan `/variables` tallentuu kaikki työtilassa sillä hetkellä olevat muuttujat. Eli:  
`save,filename='tiedostonimi.save',/variables`

# IDL - SAVE ja RESTORE-komennot

- ▶ Vastaavasti palautus:  
`restore, 'tiedostonimi.save'`
- ▶ Antamalla avainsanan `/verbose` näet komentoriviltä palautettavien muuttujien nimet. Eli:  
`restore, 'tiedostonimi.save', /verbose`
- ▶ HUOM! Muuttujat palautetaan aktiivisena olevaan työtilaan. Tällöin jo olemassa olevia muuttujia voidaan korvata tiedostossa olevilla, jos muuttujat ovat saman nimisiä.
- ▶ Myös aliohjelmat ja funktiot voivat käyttää `restore` ja `save` lauseita.