

# Hat man wirklich alles beisammen? Schneller Dateivergleich mit APL2

---

If you have to make a change,  
have you lost anything important?

## Fast data comparing with APL2

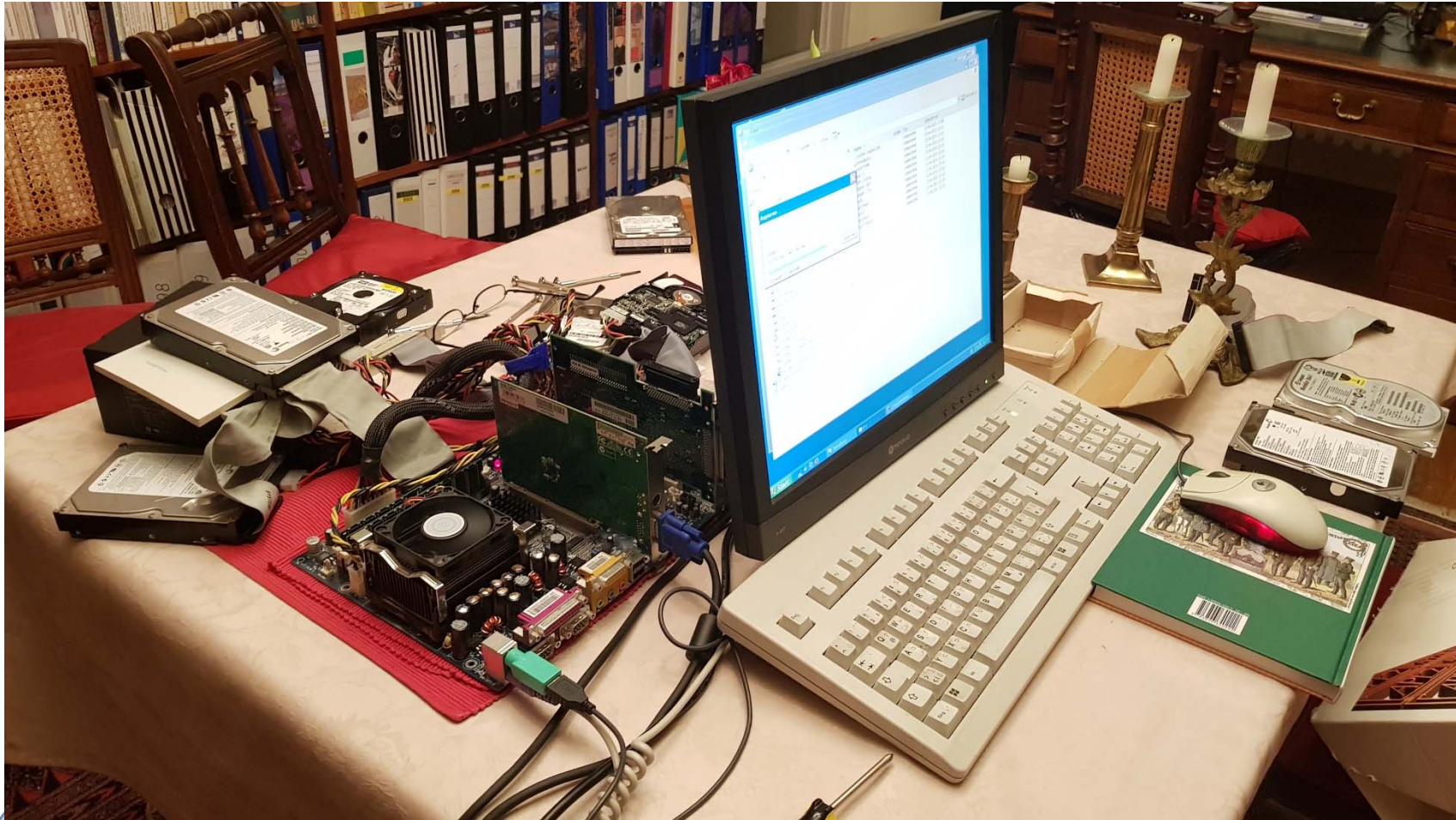


# Outline

- A way between redundancy and loss
- Redundancy of hardware
- Data or files are static or dynamic
- Most of files representing data are static but increasing fast (photos, sounds, videos)
- Principles of order: you have ordered everything by name, type, size and date
- Advantages and disadvantages of folders



# First you need hardware





# Redundancy\* of HDD



\* Latin.:having abundance



# Two methods of research

- After scanning through all folders how is the degree of redundancy (comparing internally)
- If you had a change, which data were lost? (comparing external)



# Which data are important

- Images, photos: \*.jpg, \*.tif, \*.bmp
- Videos, films: \*.mp4, \*.mov
- Programs: \*.apl, \*.atf, \*.dws
- Sounds: \*.mp3, \*.wav, \*.wma
- Documents: \*.txt, \*.doc, \*.docx, \*.pdf

Almost each data is static



# indices ← matrix1 RF matrix2

- This external function returns the index of the first row in matrix1 of each row of matrix2. The function is equivalent to the expression:
- $\text{indices} \leftarrow (\text{c}[1+\square|\text{O}]\text{matrix1})\iota\text{c}[1+\square|\text{O}]\text{matrix2}$
- The arguments must be rank 2 character, integer or nested arrays that have the same number of columns. Both arguments must be the same type. If the arguments are nested, their depth must be 2 or less, and all of their subitems must be character or integer arrays.
- This function may provide a **performance improvement** over the APL2 expression for some types of data.



# First, looking for image data (internally)

```
3 11 □NA 'RF' a activate external function RF
ph←>('DIR /T /S ', 'D:\bilder\*.jpg ')PIPE ',
31223 84          a 31223 files at drive D
/T: date and time of creating the file
/S: scan also all subdirectories
pi1←h RF h a dyadic, very fast
□wa is 8720416
31223
pi1←(h1) RF(h1←h, [1]h)          a dyadic
62446          a with double data
```





# Remove empty lines, it looks better

10↑i1

1 2 3 4 3 6 7 8 3 10     a empty lines at 3

7 50↑h

Datenträger in Laufwerk D: ist Daten

Volumeseriennummer: F843-04BC

Verzeichnis von d:\bilder

05.03.2018 11:57                   392.747 martin\_67.jpg

15.08.2019 09:50                   507.988 martin\_fu.jpg

          ph←(i1≠3)/[1]h           a without empty lines

30976 84

6 50↑h

Datenträger in Laufwerk D: ist Daten

Volumeseriennummer: F843-04BC

Verzeichnis von d:\bilder

05.03.2018 11:57                   392.747 martin\_67.jpg

15.08.2019 09:50                   507.988 martin\_fu.jpg

          2 Datei(en),             900.735 Bytes



Martin Barghoorn   APL-Online

Mai 2021

# Find and show redundancy image files

```
predun←((i2∖i2)≠∖pi2)/i2      A replicates  
816
```

```
7↑redun
```

```
9842 9843 9845 9847 9849 9850 9852
```

```
h[9842 14624;∖50]
```

```
10.02.2018 14:56 1.071.616 P1080746.JPG
```

```
10.02.2018 14:55 1.028.608 P1080744.JPG
```

The degree of redundancy:  $100 \times (\text{predun}) \div \text{pi}2$   
2.6 % not much



# Find and show redundancy **atf** files

```
h[10;55]          a atf-files since 1977
Datenträger in Laufwerk D: ist Daten
Volumeseriennummer: F84304BC
Verzeichnis von d:\DATABASE\APL_GER\APLJournal\2004
16.11.2004  15:35          4.428 MIDIPLAY.ATF
                1 Datei(en),          4.428 Bytes
    Verzeichnis von d:\DATABASE\APL_GER\APLJournal\2
06.03.2007  19:35          476.338 CALL225.atf
13.02.2007  16:19          249.690 OPY51.atf
                2 Datei(en),          726.028 Bytes
Verzeichnis von d:\DATABASE\APL_GER\Exif
degree of redundancy:  100×((pi2)-puni)÷pi2
17.2 %                to much?
```



# Remember the old-time APL-files (mainframe)

Verzeichnis von d:\martin\maintran\argus

19.04.1996	11:52	409.360	berl8485.atf
19.11.1996	18:19	27.040	berlin92.atf
18.04.1996	12:26	371.214	bh83.atf
19.04.1996	11:51	328.160	bh84.atf
19.04.1996	14:08	323.920	bh85.atf
19.04.1996	11:51	20.080	bh89.atf
19.04.1996	11:52	13.600	bh90.atf
19.04.1996	11:52	43.120	bh91.atf
19.04.1996	11:52	29.040	bh92.atf
19.04.1996	11:51	36.000	dass92.atf
19.04.1996	11:51	31.760	dass93.atf



# The other method: which data was lost?

```
ρh1←⇒('DIR /T /S ', 'D:\bilder\*.jpg ')PIPE ''
31223 84
ρh2←⇒('DIR /T /S ', 'K:\bilder\*.jpg ')PIPE ''
30417 80          A extern drive K
      ρi1←h1 RF h1          A dyadic RF
31223
      ρi2←h2 RF h2          A dyadic RF
30417
      ρ¨h1 h2
30976 84 30180 80
ρuni←UNIQUE i  ♦  MAX uni
29149          30977
```





# Show lost files, find index

```
10↑i      A  where are lost files
30977 30977 30977 4 5 6 30977 9 10 11
plost←(i=30977)/ι30180
216
10↑lost   A  index lost files
1 2 3 7 119 120
jpeg_lost←h2[lost;]
one_image←'06.12.2020 10:39 2.212.899
20201206_093941.jpg'
```



# Show record of lost files

06.06.2020	06:29	151.378	buda_pionie r72.jpg
06.06.2020	06:29	284.874	freies_euro pa.jpg
06.06.2020	06:29	200.079	istanbul72 .jpg
06.06.2020	06:29	89.477	kaefer_1972 .jpg
06.06.2020	06:29	81.459	martin_ness ebar.jpg
06.06.2020	06:29	127.230	moskwitsch_ 72.jpg
06.06.2020	06:29	175.044	niesky_boot .jpg
06.06.2020	06:29	130.712	saloniki_da vid.jpg



# Show old but good drive



# Summary

APL2 has had for many years  
phantastic, powerful, fast  
and ready to use external  
functions on board,  
one of them is the function

**RF**

Ich bedanke mich herzlich für  
Ihre Aufmerksamkeit!

