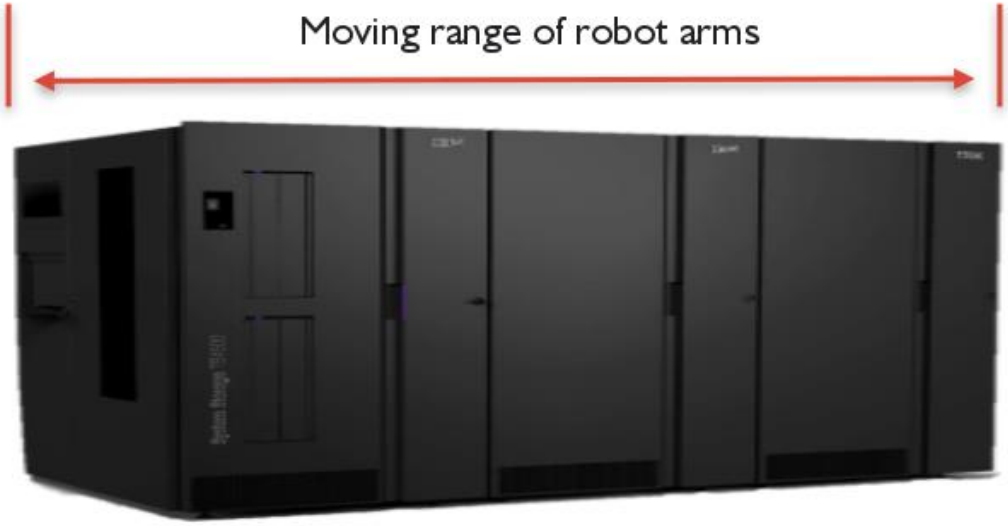
CERN Tape Archive monitoring

Filip Pajkoš, Ema Podhorná, Jozef Skokan

Content

Tape archive system at CERN

- Permanent storage for physics data
- Currently stores ~340 PB of data
- Using 4 different tape libraries
- Contain ~30 000 tapes
- Using ~100 tape drives


Moving range of robot arms
Controller
Drives
Cartridges
Drives
Cartridges
Cartridges


IBM

What is Linux?

- Modified UNIX based on Linux core

- The philosophy of UNIX:
  1. Many small programs doing one thing
  2. Output of one program -> input into another program

Linux + GNU utilities = free UNIX

All these small programs
make a distribution

Linux distributions

Ubuntu, one of the most popular desktop Linux distributions

Linux distributions

CERN uses CentOS, which is a free version of RedHat Enterprise Linux

redhat

CentOS

tpsrv616.cern.ch - PuTTY

Linux commands we used:

```
-bash-4.2$ help
```

- Network: ssh
- Shells: BASH
- Command Information: man
- Symbols: |, >, >>,
- Filters: grep, less
- Hotkeys: <ctrl><c>,
- File System: ls, cd, pwd, cat, chmod, cp
- File Editors: nano, vim

```
-bash-4.2$
```

What is Python?

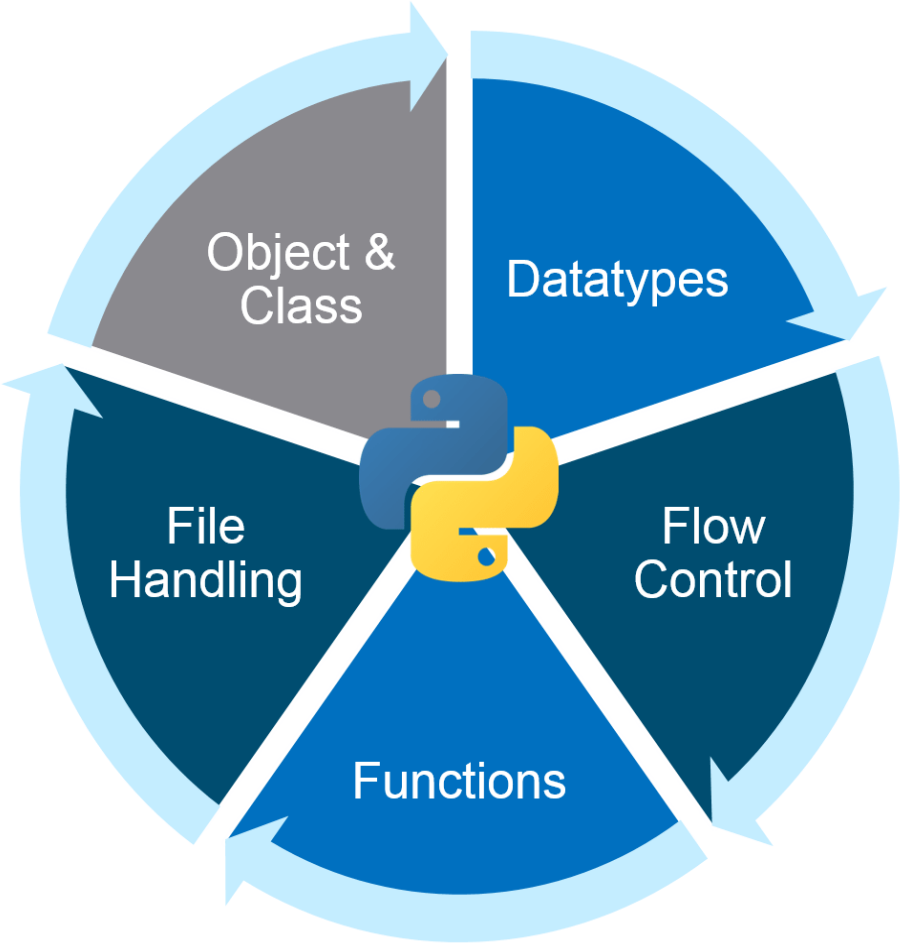
```
>>> Python_Overview
"Open-source programming language, can be used for opening apps"

>>> Python_History
["early 90s v1.0, Guido Van Rossum", "named after Monty Python"]

>>> Scope_Of_Python
{Industry, Science, Administration[Web Dev, UNIX][CGI],
AI[Machine Learning, True AI, Robots]}

>>> print (str(Python_Advantages) + ".")
Object orientation, INDENTATION, Open-source, Portable, POPULAR,
Easy to use, EASY TO LEARN, Powerful, extensive library of modules
```

Python Fundamentals



FUNDAMENTALS

Datatypes	Flow Control	Functions	File Handling	Object & Class
Numbers Strings Lists Dictionaries	If Else For While Continue	Definition Function Call Docstring Return	Reading Writing Editing	Variables Functions

Tweets by @ThePSF

Python Software Foundation @ Cern

#Cern, #DataCentrum, #idemeIT

6h

Python Software Foundation @ThePSF

Embed | View on Twitter

The PSF

The Python Software Foundation is the organization behind Python. Become a member of the PSF and help advance the software and our mission.

Python Documentation

docs.python.org/3/tutorial/inputoutput.html

stackabuse.com/read-a-file-line-by-line-in-python/

Our Code

```python
subor3 = open ("/var/hssip/pasky.json")

drls = json.loads(subor1.read())
sq = json.loads(subor2.read())
pasky = json.loads(subor3.read())


for opt, arg in options:
    if opt in ("-s", "--Status"):
        statuserror = arg
    elif opt in ("-l", "--limit"):
        #print(type(longtimeerror))
        longtimeerror = int(arg)
        #print(type(longtimeerror), longtimeerror)
    elif opt in ("-a", "--average"):
        averageError = int(arg)

print ("-------------------")
print ("|     SET UP        |")
print ("|Status:", statuserror,"|")
print ("|Limit:",longtimeerror,"    |")
print ("|Average:",averageError,"   |")
print ("-------------------")


for obj_drls in drls:
   status = obj_drls["driveStatus"]
   since = int(obj_drls["driveStatusSince"])

   if status == "TRANSFERRING" and since > longtimeerror:
      print("Prekroceny limit (Status mechaniky:",status, ", Cas chodu:",since, ")")
   else:
      print("Status mechaniky:",status, ", Cas chodu:",since)


print()

for obj_sq in sq:
   mountTypevypis = obj_sq["mountType"]
   vidvypis = obj_sq["vid"]
   tapepoolvypis = obj_sq["tapepool"]
   logicalLibraryvypis = obj_sq["logicalLibrary"]
   oldestAgevypis = int (obj_sq["oldestAge"])

   if logicalLibraryvypis == "IBM355" and oldestAgevypis > oldestAgeError:
      IBM355.append(oldestAgevypis)
   elif logicalLibraryvypis == "IBM455" and oldestAgevypis > oldestAgeError:
      IBM455.append(oldestAgevypis)

   if oldestAgevypis != (oldestAgeError):
      print("Mount type: ",mountTypevypis, ", VID: ",vidvypis, ", Tape pool: ",tapepoolvypis, ", Logical Library: ",logicalLibraryvypis, ", Oldest Age: ",oldestAgevypis)


if Average(IBM355) > float (averageError):
   print ("IBM355 Average:", round(Average(IBM355),2))
if Average(IBM455) > float (averageError):
   print ("IBM455 Average:", round(Average(IBM455),2))

exit()
```

Our Code has 3 parts

```
print("Anomalies in DRLS file:\n")

for i in DRLS:
   tapestatus = i["driveStatus"]
   tapetime = int(i["driveStatusSince"])
   if tapetime > int(longlimit):
      if tapestatus == str('TRANSFERRING'):
         print("Tapestatus is: ",tapestatus,"     Tapetime equals = ",tapetime)
   if tapetime > int(shortlimit):
      if tapestatus == str('MOUNTING'):
         print("Tapestatus is: ",tapestatus,"     Tapetime equals = ",tapetime)


#SQ file

SQsubor = open("/var/hssip/sq.json")
SQ = json.loads(SQsubor.read())

print("\n\nAnomalies in SQ file - average values:\n")

for i in SQ:
   mountType = i["mountType"]
   vid = i["vid"]
   tapepool = i["tapepool"]
   logicalLibrary = i["logicalLibrary"]
   oldestAge = int(i["oldestAge"])
   if logicalLibrary == 'IBM355':
      if oldestAge > OLDESTAGElimit:
         IBM355.append(oldestAge)
   if logicalLibrary == 'IBM455':
      if oldestAge > OLDESTAGElimit:
         IBM455.append(oldestAge)

if averagefunction(IBM355) > float(averagevalue):
   print("Average value for IBM355:", averagefunction(IBM355))
if averagefunction(IBM455) > float(averagevalue):
   print("Average value for IBM455:", averagefunction(IBM455))


#PASKY file

PASKYsubor = open("/var/hssip/pasky.json")
PASKY = json.loads(PASKYsubor.read())

print("\n\nAnomalies in PASKY file:\n")

for i in PASKY:
   tapepool = i["tapepool"]
   disabled = i["disabled"]
   full = i["full"]
   rdonly = i["rdonly"]
   vid = i["vid"]
   occupancy = int(i["occupancy"])
   lastfseq = int(i["lastFseq"])
   beginssupply = tapepool[:7]
   if (beginssupply == "supply_") and (disabled or full or rdonly or (occupancy != 0) or (lastfseq != 0)):
      print("Vid :", vid,"    Tapepool :", tapepool,"        Disabled :", disabled,"    Full :", full,"    Rdonly :", rdonly,"    Occupancy :", occupancy,"    Lastfseq :", lastfseq)
      if vidlist != []:
         vidlist.append(vid)
```

1) Checks that the tape drives are used efficiently
   - Checking the transition and running states

2) Checks the length of the tape queues

3) Checks the state of the tapes in the supply

We used the following data types

```
      else:
        tapeRUNNINGerrorstatus = tapeRUNNINGerrorstatus + 1
        RUNNINGtapes = 1

  elif tapeStatus == "TRANSFERRING":
    if tapeTime > limitTRANSFERRING:
      tapeTRANSFERRINGerrordictionary[str(tapeName)] = "Time Error: the time limit exceeded for this long-term-status tape, maintenance needed!"
      if int(len(tapeTRANSFERRINGerrordictionary.keys())) == 0:
        tapeTRANSFERRINGerrorstatus = 1
      else:
        tapeTRANSFERRINGerrorstatus = tapeTRANSFERRINGerrorstatus + 1
        TRANSFERRINGtapes = 1

tapeErrorDictionary = dict(tapeTRANSFERRINGerrordictionary, **tapeRUNNINGerrordictionary)
tapeStatusDictionary = dict(tapeTRANSFERRINGdictionary, **tapeRUNNINGdictionary)
tapeErrorStatus = tapeTRANSFERRINGerrorstatus + tapeRUNNINGerrorstatus

if tapeErrorStatus > 0:
  print("\nTime errors detected in the tapes! TimeStatus: NOT OK! " + str(tapeErrorStatus) + " errors detected.")
elif tapeErrorStatus < 0:
  print("\n\n\nWow, you broke my programme! Please tell me how you did that, VERY EXTREMELY NOT OK")
  exit()
elif tapeErrorStatus == 0:
  print("\nNo time anomalies in the long and short-term-status tapes. TimeStatus(for them): OK!")
else:
  print("\nProgram error! Please be patient")
  exit()

if TRANSFERRINGtapes == 1:
    print("\n\nAll yout TRANSFERRING errorenous tapes: \n ")
    print(tapeTRANSFERRINGdictionary)
    print(tapeTRANSFERRINGerrordictionary)
    print("\nThe number of TRANSFERRING tapes which are anomalous is " + str(tapeTRANSFERRINGerrorstatus))

if RUNNINGtapes == 1:
    print(tapeRUNNINGdictionary)
    print(tapeRUNNINGerrordictionary)
    print("\nThe number of RUNNING tapes which are anomalous is " + str(tapeRUNNINGerrorstatus))

print("\n\nEvalutation of long and short-term-status tapes done")


print("\n\n\nNow evaluating the state of your stored tapes, please wait :)")

for blocktext in tapes:
  tapePool = blocktext["tapepool"]
  tapePoolSubstr = tapePool[:7]
  disabled = blocktext["disabled"]
  vid = blocktext["vid"]
  full = blocktext["full"]
  occupancy = blocktext["occupancy"]
  readOnly = blocktext["rdonly"]
  if tapePoolSubstr == "supply_" and (disabled or full or readOnly or occupancy == 0):
      if supplyList == []:
        supplyList = [vid]
      else:
        supplyList.append(vid)
      print ("\nData of your error tapes: \nVid: " + vid + "\nDisabled: " + str(disabled) + "\nFull: " + str(full) + "\nOccupancy: " +  occupancy + "\nReadOnly: " + str(readOnly) + "\nTapePool: " + tapePool + "\n")
```

- Boolean bool(True, False)
- Number int(9) or float(42.21)
- String str("Hello!")
- List list[val1, val2, val3,...]
- Dictionary dict(key1:value1, key2:value2)

Activities GNU Emacs 25 (GUI) Sat 08:06

Trash

16 GB Volume

emacs25

File Edit Options Buffers Tools Help

What is JSON?

```
JSON Object → {                                  String Value
                "company": "mycompany",
                "companycontacts": {  ← Object Inside Object
                  "phone": "123-123-1234",
                  "email": "myemail@domain.com"
                },
                "employees": [  ← JSON Array
                  {
                    "id": 101,
                    "name": "John",
Array Inside Array →  "contacts": [
                      "email1@employee1.com",
                      "email2@employee1.com"
                    ]
                  },
                  {
                    "id": 102,  ← Number Value
                    "name": "William",
                    "contacts": null  ← Null Value
                  }
                ]
              }
```

1:%%- *GNU Emacs* All L17 (Fundamental)

About_JSON.json

What have we learned?

Tape data storage

- Principles and functions
- Libraries, drives, tapes and their statuses
- What are anomalies and how to detect them

Programming

- Commands
- Data types
- Using modules
- Handling json files

HSSIP

- CERN mission, particle physics, detectors and colliders, scientific society...

- Eat fondue
- New people
- Geneva and its attractions

Sources

https://www.python.org/doc/essays/blurb/

http://openbookproject.net/courses/intro2ict/system/os_intro.html

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/linux_anatomy.pdf

https://www.kernel.org/category/faq.html

http://timelessrepo.com/json-isnt-a-javascript-subset

Nous vous remercions pour votre attention