Speak Python with Devices

Petertc Chu @ EuroPython 2020

Why this session?

We will see:

- How Python can be applied in IoT/infrastructure automation tasks

You might:

- wish to know how Python can be used beyond data analysis and web dev
- a Pythonista who interested in craft some touchable things 🚂🚂🚂
- want to acquire something new into your Python skillset 💪💪💪

Outline

- Devices in Linux/UNIX-like system
- How to manipulate a device
- Manipulate a device in Python, a mini example
- A more attractive example

User space

Your **Python** interpreter, packages and code here

device files in the /dev/ directory

```
$ ls /dev/
autofs          log               mhvtl34           nst1a  nst7    sch0  sg6
block           loop0             mhvtl90           nst1l  nst7a   sch1  sg7
bsg             loop1             mhvtl91           nst1m  nst7l   sch2  sg8
btrfs-control   loop2             mhvtl92           nst2   nst7m   sda   sg9
char            loop3             mhvtl93           nst2a  nst8    sda1  shm
console         loop4             mhvtl94           nst2l  nst8a   sdb   snaps
core            loop5             mqueue            nst2m  nst8l   sdb1  snd
cpu_dma_latency loop6             net               nst3   nst8m   sdc   st0
cuse            loop7             network_latency   nst3a  nst9    sdc1  st0a
disk            loop-control      network_throughput nst3l  nst9a   sg0   st0l
ecryptfs        mapper            nst0              nst3m  nst9l   sg1   st0m
fb0             mcelog            nst0a             nst4   nst9m   sg10  st1
fd              mem               nst0l             nst4a  null    sg11  st10
```

Kernel space — **Driver** of LED panel, camera, sensors and other devices...

Hardware — LED panel, camera and sensors on bluetooth/USB/serial/parallel ports...

Computer oganization, briefly

Everything is a file, so is a device

Manipulate a device

with common file operations:

- open()
- write()
- read()
- close()

Example: blink an LED on Raspberry Pi

```
18  def led_on():
19      with open('/sys/class/leds/led0/brightness', 'w') as brightness:
20          brightness.write('255')
21      print('on')
22
23  def led_off():
24      with open('/sys/class/leds/led0/brightness', 'w') as brightness:
25          brightness.write('0')
26      print('off')
27
28  import time
29
30  def blink(sec: int):
31      disable_trigger()
32      for i in range(sec):
33          led_on()
34          time.sleep(1)
35          led_off()
36          time.sleep(1)
```

and a more interesting one...

IOCTL()

IOCTL - What? Why?

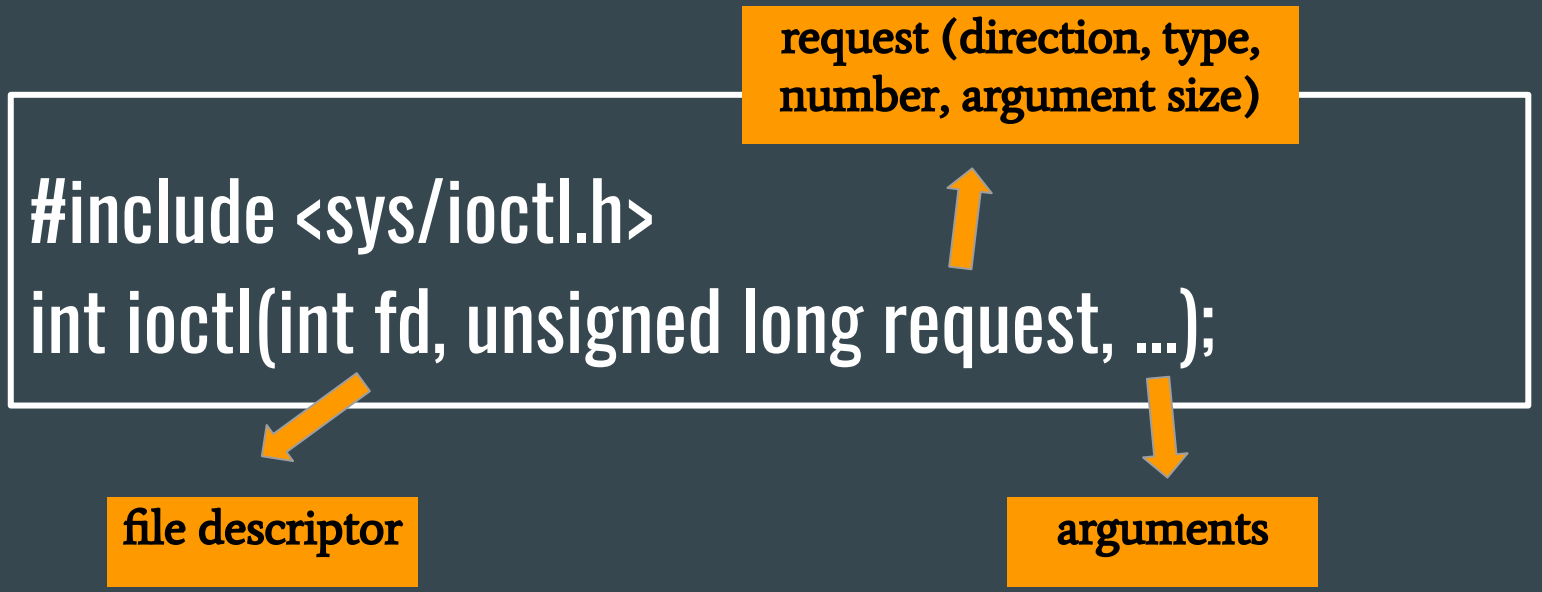
Input Output ConTroL

Read/write is not enough for a device which is more complex than an LED

Example: a modem

- READ - reveive data
- WRITE - send data
- IOCTL - talk to the modem itself, e.g., set bitrate, get config

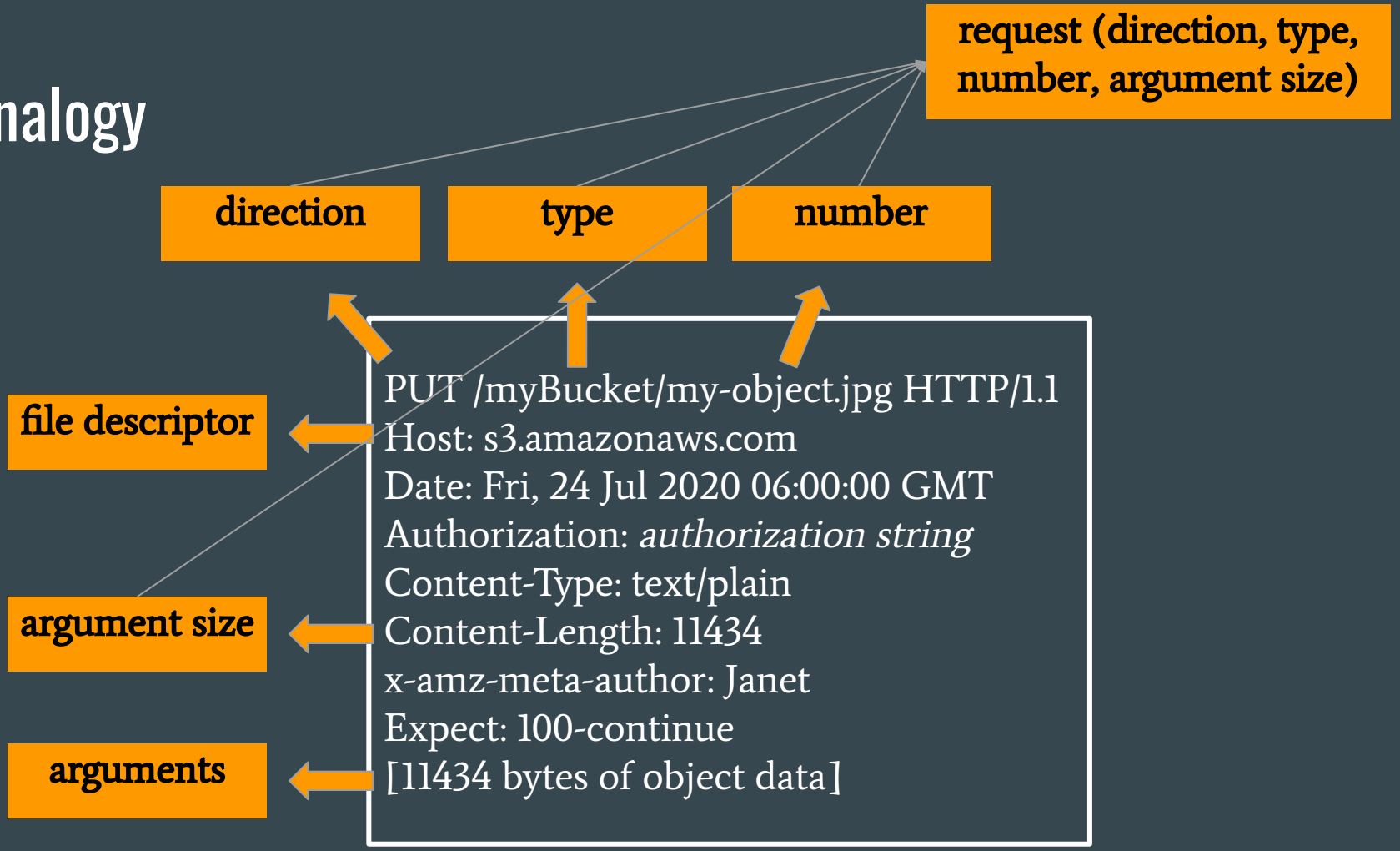
IOCTL - Decoration



IOCTL - Parameters

- file descriptor
- request
  - a.k.a. (device-dependent) request code or command
  - composed of:
    - type (8 bits, a~z)
    - number (8 bits, 1~255)
    - argument size (14 bits, max 16KB)
    - direction (2 bits, R/W/RW/NONE)
- argument (string, a C struct or anything)

An analogy


request (direction, type, number, argument size)
direction
type
number
file descriptor
argument size
arguments
PUT /myBucket/my-object.jpg HTTP/1.1
Host: s3.amazonaws.com
Date: Fri, 24 Jul 2020 06:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 11434
x-amz-meta-author: Janet
Expect: 100-continue
[11434 bytes of object data]

“DON’T PANIC!”

just like RESTful APIs we use every day!

IOCTL 💛 Python

Let’s start from a mini example:

Get the name of input devices

```
...: if __name__ == '__main__':
...:     for i in range(7):
...:         with open('/dev/input/event' + str(i)) as fd:
...:             print(get_device_name(fd))
...:
Power Button
Sleep Button
AT Translated Set 2 keyboard
Video Bus
ImExPS/2 Generic Explorer Mouse
VirtualBox USB Tablet
VirtualBox mouse integration
```

Do IOCTL() from Python

Things to do:

- Create an IOCTL request (header)
- C<->Py type convertion (body)
- Do IOCTL system call

(At least) two approaches:

- C extension module
- Pure Python solution

Approach 1: C extension module

Step 1: IOCTL call

```c
char* getDeviceName(char* device, char* name, size_t len)
{
    int fd = -1;
    int result = 0;

    fd = open(device, O_RDONLY);

    // EVIOCGNAME(len) _IOC(_IOC_READ, 'E', 0x06, len) (linux/input.h)
    int request = EVIOCGNAME(len);

    result = ioctl(fd, request, name);

    close(fd);
}
```

Create IOCTL request (header) by macros

Approach 1: C extension module

Step 2: C<->Py type convertion (req/resp body)

```
25  static PyObject * get_device_name(PyObject *self, PyObject *args) {
26      // parse input
27      const char *device;
28      if (!PyArg_ParseTuple(args, "s", &device))
29          return NULL;
30
31      char name[256] = "Unknown";
32      getDeviceName(device, name, 256);
33
34      // return name
35      return Py_BuildValue("s", name);
36  }
```

Approach 1: C extension module

Step 3: packaging

```
38  static PyMethodDef EVIOCG_Methods[] = {
39      {"get_device_name",  get_device_name, METH_VARARGS,
40       "Get device name."},
41      {NULL, NULL, 0, NULL}        /* Sentinel */
42  };
43
44  static struct PyModuleDef EVIOCG_MODULE = {
45      PyModuleDef_HEAD_INIT,
46      "eviocg",   /* name of module */
47      NULL, /* module documentation, may be NULL */
48      -1,       /* size of per-interpreter state of the module,
49                   or -1 if the module keeps state in global variables. */
50      EVIOCG_Methods
51  };
52
53  PyMODINIT_FUNC PyInit_eviocg(void) {
54      PyObject* m = PyModule_Create(&EVIOCG_MODULE);
55      if (m == NULL) {
56          return NULL;
57      }
58      return m;
```

Approach 1: C extension module

Install and use it as usual

```
/pyeviocg# pip list|grep eviocg
eviocg            0.0.0



IPython 7.9.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import eviocg

In [2]: eviocg.get_device_name('/dev/input/event3')
Out[2]: 'Video Bus'
```

Approach 2: Pure Python solution

Step 1: Create an IOCTL request (header)

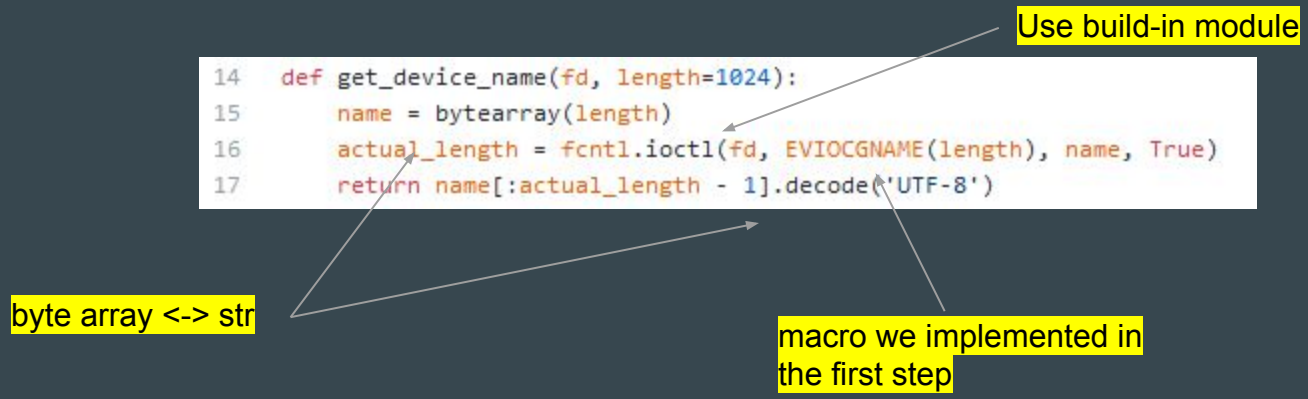
- ~~porting IOC* macros from asm-generic/ioctl.h~~ => Someone has already done it!
  - olavmrk/python-ioctl
  - vpelletier/python-ioctl-opt
- porting driver specific macros

```
6  from ioctl_opt import IOC, IOC_READ
7
8  # porting from linux/input.h
9  EVIOCGNAME = lambda length: IOC(IOC_READ, ord('E'), 0x06, length)
```

Courtesy of
vpelletier/python-ioctl-opt/blob/master/README.rst

Approach 2: Pure Python solution

Step 2: ioctl call and C<-> data type convertion



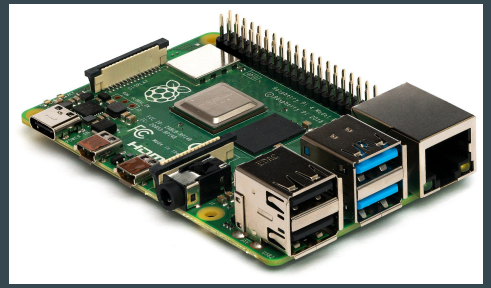
Approach 2: Pure Python solution

Same result we saw before

```
...: if __name__ == '__main__':
...:     for i in range(7):
...:         with open('/dev/input/event' + str(i)) as fd:
...:             print(get_device_name(fd))
...:
Power Button
Sleep Button
AT Translated Set 2 keyboard
Video Bus
ImExPS/2 Generic Explorer Mouse
VirtualBox USB Tablet
VirtualBox mouse integration
```

OK now I know how these things work but...

Question: any use case? 🤔



https://zh.wikipedia.org/wiki/%E6%A0%91%E8%8E%93%E6%B4%BE#/media/File:Raspberry_Pi_4_Model_B_-_Side.jpg

+

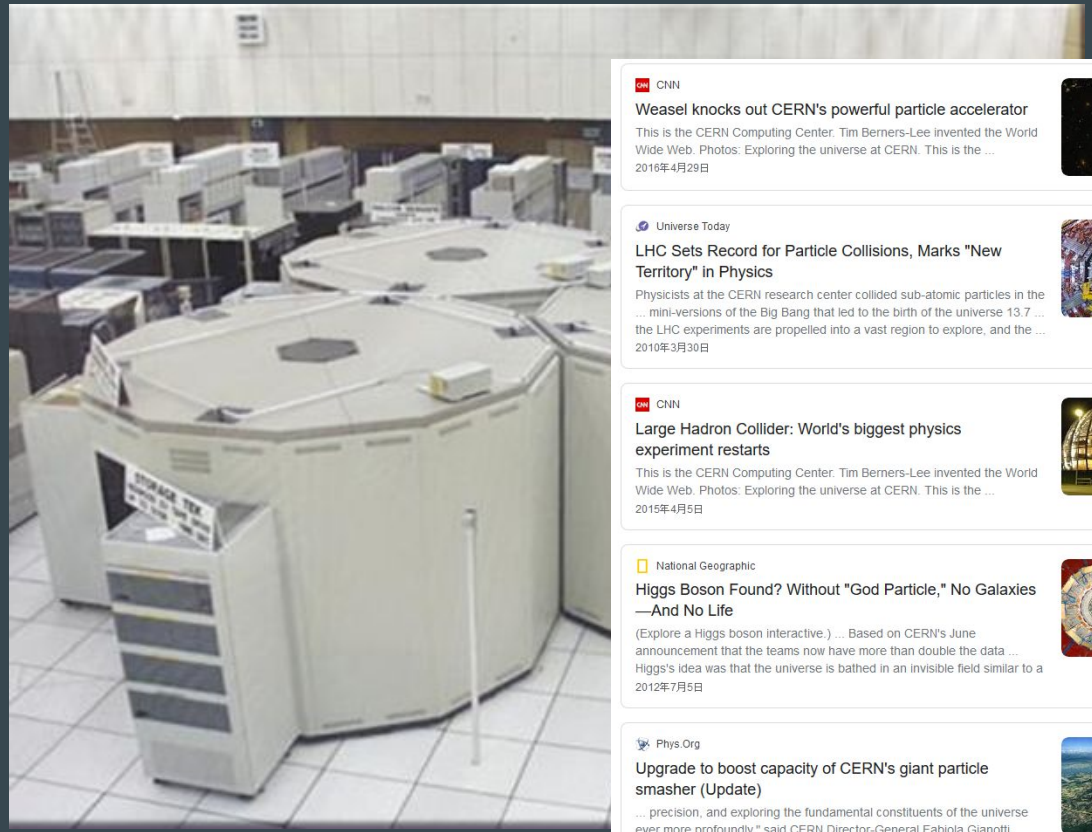
https://twitter.com/MISSINGEGIRL/status/1123647491025428480?s=20

=

https://youtu.be/KQKCf5u9axk

!

a cat food feeder?



CNN

Weasel knocks out CERN's powerful particle accelerator

This is the CERN Computing Center. Tim Berners-Lee invented the World Wide Web. Photos: Exploring the universe at CERN. This is the ...

2016年4月29日

Universe Today

LHC Sets Record for Particle Collisions, Marks "New Territory" in Physics

Physicists at the CERN research center collided sub-atomic particles in the ... mini-versions of the Big Bang that led to the birth of the universe 13.7 ... the LHC experiments are propelled into a vast region to explore, and the ...

2010年3月30日

CNN

Large Hadron Collider: World's biggest physics experiment restarts

This is the CERN Computing Center. Tim Berners-Lee invented the World Wide Web. Photos: Exploring the universe at CERN. This is the ...

2015年4月5日

National Geographic

Higgs Boson Found? Without "God Particle," No Galaxies—And No Life

(Explore a Higgs boson interactive.) ... Based on CERN's June announcement that the teams now have more than double the data ... Higgs's idea was that the universe is bathed in an invisible field similar to a

2012年7月5日

Phys.Org

Upgrade to boost capacity of CERN's giant particle smasher (Update)

... precision, and exploring the fundamental constituents of the universe ever more profoundly," said CERN Director-General Fabiola Gianotti.

http://castor.web.cern.ch/castor/

What kind of data ?

Digitized tracks of particles in detectors

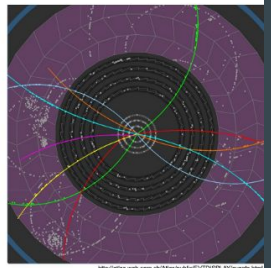
Data must be collected as it is generated (~18 months uninterrupted)

One event similar to the others

Volume: 15-20 PB/year

Transfer rates: ~0.5 – 1.5 GB/s

Keep for > 10 years (forever)



http://storageconference.us/2010/Presentations/MSST/15.Bahyl.pdf

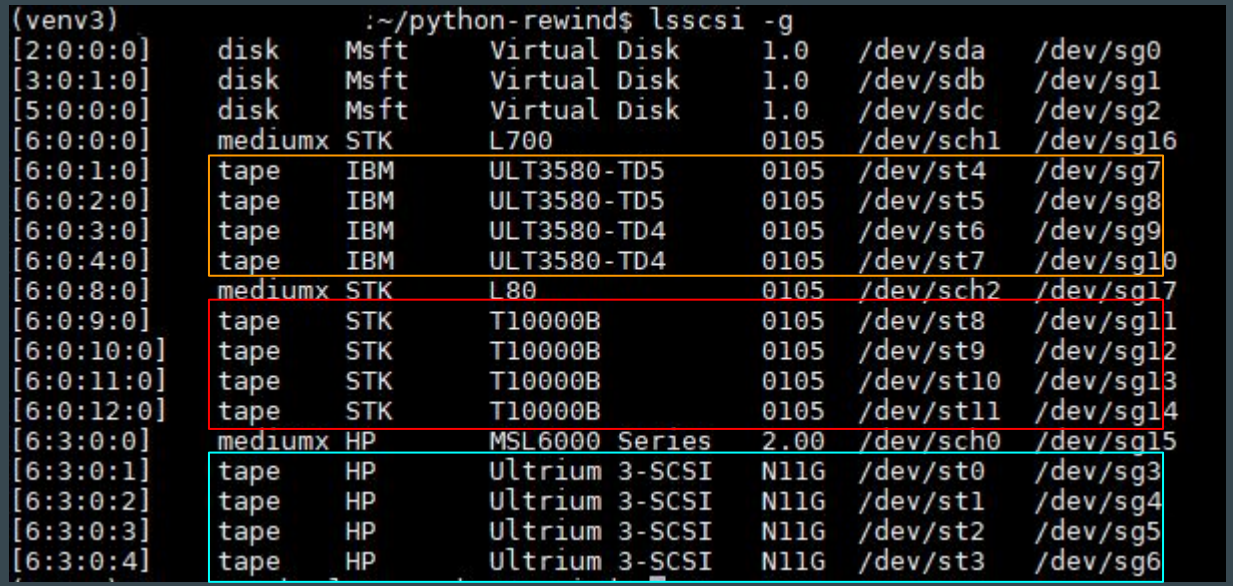
https://youtu.be/IDgXa0ioVTs

The CERN Advanced STORage system (CASTOR)

Explore the universe 🚀
with what we learn today!

Quick start

- Device: mhVTL simulated
- Driver: Linux SCSI tape (st) driver



Quick start

Typical tape write procedure:

1. Find the cartridge by barcode scanner
2. Load the cartridge by a robot arm
3. **Check the cartridge status is ready**
4. **Rewind the cartridge by a tape drive**
5. Write data on the cartridge
6. Unload the cartridge

👉 👉 👉 What we're gonna do today

Snippet 1: Get tape status by C extension

```
  // open device file
  int fd;
  if ((fd = open(device, O_RDONLY)) < 0) {
      PyErr_SetFromErrno(PyExc_OSError);
      return NULL;
  }
  // execute ioctl command
  struct mtget status;
  if (ioctl(fd, MTIOCGET, (char *)&status) < 0) {
      PyErr_SetFromErrno(PyExc_OSError);
      return NULL;
  }
  if (status.mt_type != MT_ISSCSI2) {
      PyErr_SetString(PyExc_NotImplementedError, "Unsupported tape
type");
      return NULL;
  }
  close(fd);

  // return status info in dict
  return Py_BuildValue("{s:i,s:i,s:i}",
          "file number", status.mt_fileno,
          "block number", status.mt_blkno,
          "partition", (status.mt_resid & 0xff)
  );
}
```

Snippet 2: use struct

Convert function arguments back and forth. struct.pack() and struct.unpack() are your friends here.

```
def rewind(device):
    MTREW = 6
    mt_com = struct.pack('hi', MTREW, 1)
    MTIOCTOP = IOW(ord('m'), 1, len(mt_com))

    with open(device, 'r') as fd:
        fcntl.ioctl(fd, MTIOCTOP, mt_com)

def status(device):
    long_size = 8
    int_size = 4
    status = bytearray(long_size * 5 + int_size * 2)
    MTIOCGET = IOR(ord('m'), 2, len(status))

    with open(device, 'r') as fd:
        fcntl.ioctl(fd, MTIOCGET, status)
        status = struct.unpack('llllli', status)
        return {
            "file number": status[-2],
            "block number": status[-1],
            "partition": status[1] & 0xff
        }
```

Bonus: rewind cartridge by ctypes

Define input/output/buffer data structure by extending ctypes.Structure

```
class mtop(ctypes.Structure):
  _fields_ = [
    ("mt_op", ctypes.c_short),
    ("mt_count", ctypes.c_int)
  ]


def rewind(device):
  MTIOCTOP = ioctl.linux.IOW('m', 1, ctypes.sizeof(mtop))
  MTREW = 6
  mt_com = mtop(MTREW, 1)
  with open(device, 'r') as fd:
    ioctl.ioctl(fd.fileno(), MTIOCTOP,
ctypes.byref(mt_com))
```

PoC

```
/venv/playioctl/bin/python /playioctl/demo.py
Demo tape rewind/status operation powered by <module 'mt' from '/venv/playioctl/lib/python3.5/site-packages/mt-0.0.0-py3.5-linux-x86_64.egg/mt.cpython-35m-x86_64-linux-gnu.so'>
AS-IS:
{'file number': 880, 'block number': -1, 'partition': 0}
TO-BE:
{'file number': 0, 'block number': 0, 'partition': 0}
Demo tape rewind/status operation powered by <module 'by_fcntl.mt' from '/playioctl/by_fcntl/mt.py'>
AS-IS:
{'file number': 880, 'block number': -1, 'partition': 0}
TO-BE:
{'file number': 0, 'block number': 0, 'partition': 0}
Demo tape rewind/status operation powered by <module 'by_ctypes.mt' from '/playioctl/by_ctypes/mt.py'>
AS-IS:
{'file number': 880, 'block number': -1, 'partition': 0}
TO-BE:
{'file number': 0, 'block number': 0, 'partition': 0}
```

Takeaway

- You can manipulate a device like a file
- IOCTL is just like RESTful APIs we use every day
- Yes, we can speak Python while working on IoT and infra automation tasks

Example code is available on https://github.com/hrchu/playioctl

Thank you! 🙏🙏🙏

 @petertc_chu