

Gems of stdlib

- struct
- ctypes

- CSV
- sqlite

struct

- Binární data do python proměnných a naopak
- Použití:
 - Čtení binárních formátů
 - Implementace binárních síťových protokolů
- Takové přesné printf / scanf

struct – BYTE[] → Python

```
# data = socket.recv(...)
BASE = 20+BOARD_TYPE_LEN
(board, sn, o_id, status, op_count) = struct.unpack(
    "<50sqiii", data[0:BASE])
board = c_string(board)
i = 0
ops = []
for i in range(op_count):
    (step, op, res) = struct.unpack(
        "<ii20s",
        data[BASE + i*28:BASE + i*28 + 28],
    )
    res = c_string(res)
    ops.append(Operation(step, op, res))
```

struct – Python → BYTE[]

```
data = struct.pack(
    "<iii50sqiii20s%ds" % len(info),
    user_id, workplace, order_id,
    str(board), idx,
    step0, step1, op,
    res, info,
)
# socket.sendall(data)
```

ctypes

Volání funkcí z dynamických knihoven

```
iCubeLib = ctypes.windll.LoadLibrary("ICubeSDK.dll")  
cam_count = iCubeLib.ICubeSDK_Init()
```

```
GetSerialNum = iCubeLib.ICubeSDK_GetSerialNum  
GetSerialNum.restype = ctypes.c_int  
GetSerialNum.argtypes = [ctypes.c_int, ctypes.c_char_p]  
for i in range(cam_count):  
    iCubeLib.ICubeSDK_Open(i)  
    sn = (ctypes.c_char*50)()  
    result = GetSerialNum(i, sn)  
    CAMERA_MAP[sn.value.decode("ascii", "replace")] = i  
    iCubeLib.ICubeSDK_Close(i)
```

ctypes

Dokonce i callbacky

```
def frame_cbx(buf, buffer_size, context):  
    np_a = np.ctypeslib.as_array(buf, shape=(W, H, 3))  
    last_frame = np_a.copy()  
    return 0  
  
CALLBACK_TYPE = ctypes.WINFUNCTYPE(  
    ctypes.c_long, ctypes.POINTER(ctypes.c_ubyte),  
    ctypes.c_long, ctypes.c_long)  
  
cb_f = CALLBACK_TYPE(frame_cbx)  
iCubeLib.ICubeSDK_SetCallback(cam_idx, 1, cb_f, cam_idx)  
iCubeLib.ICubeSDK_Start(cam_idx, None, False, True)
```

CSV

Čtení / Zápis csv souborů

- Detekce typu csv:

```
f = open('/etc/passwd', 'rb')
sniff_data = csvfile.read(8192)
f.seek(0)
```

```
sniffer = csv.Sniffer()
has_header = sniffer.has_header(sniff_data)
dialect = sniffer.sniff(sniff_data)
```

```
assert dialect.delimiter == ':'
assert not has_header
```

CSV

Čtení / Zápis csv souborů

- Pokračujeme čtením:

```
reader = csv.reader(f, dialect)

for (idx, row) in enumerate(reader):
    if not idx and has_header:
        # this is a header row
        print('Header:', row)
    else:
        print(row)
```

CSV

Použijeme pro zápis, protože:

- uvozovkuje
- escapuje

csv → sqlite

Čtení / Zápis csv souborů

- Malá úprava:

```
reader = csv.reader(f, dialect)
table = None
columns = None
for (idx, row) in enumerate(reader):
    if not table:
        table, columns = create_table(
            row, has_header)
    if has_header and not idx:
        continue
    insert_row(table, columns, row)
```

sqlite

Levná databáze se standardním rozhraním DB-API 2.0

```
import sqlite3

db_con = sqlite3.connect(":memory:")

def create_table(columns, is_header):
    if not is_header:
        columns = ["c%d" % (i+1) for i, _ in
                   enumerate(columns)]
    cur = db_con.cursor()
    cur.execute("""create table t1(%s)""" %
               ", ".join(column_names))
    cur.close()
    return ("t1", columns)
```

sqlite

Levná databáze se standardním rozhraním DB-API 2.0

```
def insert_row(table, columns, data):
    cur = db_con.cursor()
    if len(data) == len(columns):
        cur.execute(
            """insert into %s (%s) values (%s)""" % (
                table,
                ", ".join(columns),
                ", ".join( [ "?" ] * len(columns)),
            ),
            [ i.strip() for i in data ])
    else:
        raise Exception("Wrong number of columns in the row.")
    cur.close()
```

csv → sqlite

Ukázka:

<https://gist.github.com/petrblahos/9088495>

<https://www.blahos.com/download/>