

Serijalizacija objekata pomoću introspekcije

Dušan Jovanović, dusan@djovanovic.me
Serbian C++ User Group Meetup
20.07.2022. u Beogradu

Pregled

- ❖ Teorijske osnove
- ❖ Implementacija
- ❖ Praktična primena

- Šta je serijalizacija?
- Šta je introspekcija?
- Zašto idu zajedno?
- Kako da dodam u svoj projekat?

Serijalizacija

Serialization

*the process of **translating** a data structure or object state into a format that can be **stored** or **transmitted** and **reconstructed** later.*

- Wikipedia, 10 July 2022

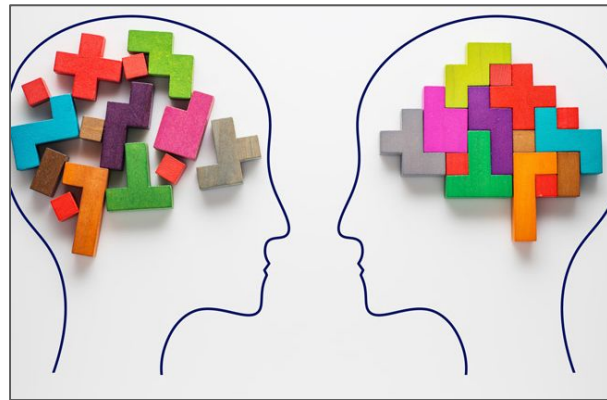
Marshalling

*the process of **transforming** the memory representation of an object into a data format suitable for **storage** or **transmission**.*

- Wikipedia, 10 July 2022

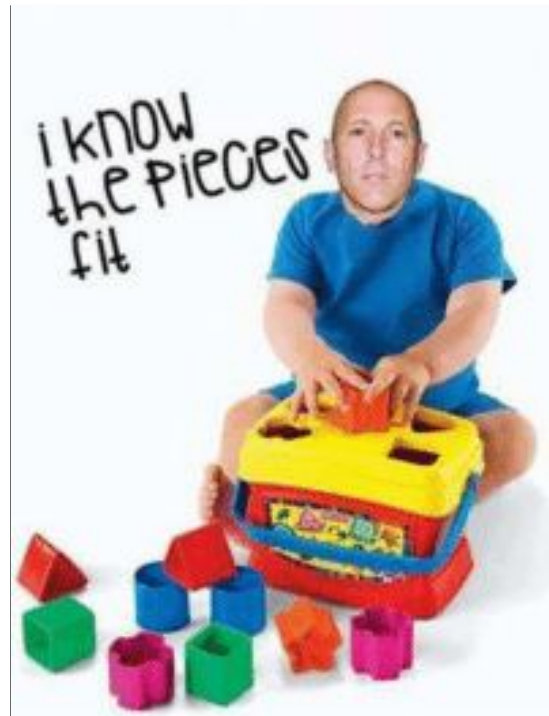
Zašto serijalizacija? Komunikacija!

- ❖ Između programa i korisnika
 - Korisnički interfejs
 - Ulazni fajlovi
- ❖ Između dva programa
 - Preko mreže
 - Preko memorije
 - Preko fajla
- ❖ Između dva pokretanja istog programa
 - Čuvanje stanja



Ali zašto serijalizacija?

- ❖ Zašto samo ne snimimo memoriju?
- ❖ Reprezentacija objekata u memoriji nije stabilna
 - Arhitektura
 - Pokazivači / reference
 - C++ - compiler, ABI, alignment, padding
 - Programski jezik



Serijalizacija u jeziku

Binarne

- Java Object Serialization
- PHP serialization format
- Pickle (Python)

Tekstualne

- JSON (JavaScript)
- EDN (Clojure)

Definicija podataka van jezika

- ❖ Šema podataka - definisanje tipova
- ❖ Formati za definisanje šeme
 - XSD (XML)
 - JSON Schema (JSON)
 - Custom za većinu binarnih formata

Formati serijalizacije

❖ Binarni

- Protocol Buffers (Google)
- Thrift (Facebook)
- Ion (Amazon)
- FlatBuffers (Google)

Formati serijalizacije

❖ Binarni

- Simple Binary Encoding
- CBOR
- BSON
- MessagePack
- Cap'n Proto

Formati serijalizacije

❖ Binarni

- ZIP
- ID3 (MP3 metadata)
- Matroska Multimedia Container (mkv)
- TIFF

Formati serijalizacije

❖ Tekstualni

- CSV
- XML
- JSON
- YAML
- TOML
- Property List (plist)
- EDN
- ...

Formati serijalizacije

❖ Tekstualni

- CSV
- XML
- JSON
- YAML
- TOML
- Property List (plist)
- EDN
- ...

```
uid,username,name,email  
12359,duxi90,Dusan Jovanovic,dusan@djovanovic.me
```

Formati serijalizacije

❖ Tekstualni

- CSV
- XML
- JSON
- YAML
- TOML
- Property List (plist)
- EDN
- ...

```
<user>  
  <uid>12359</uid>  
  <username>duxi90</username>  
  <name>Dusan Jovanovic</firstName>  
  <email>dusan@djovanovic</email>  
</user>
```

Formati serijalizacije

❖ Tekstualni

- CSV
- XML
- **JSON**
- YAML
- TOML
- Property List (plist)
- EDN
- ...

```
{  
  "user": {  
    "uid": 12359,  
    "username": "duxi90",  
    "name": "Dusan Jovanovic",  
    "email": "dusan@djovanovic.me"  
  }  
}
```

Formati serijalizacije

❖ Tekstualni

- CSV
- XML
- JSON
- **YAML**
- TOML
- Property List (plist)
- EDN
- ...

```
user:  
  uid: 12359  
  username: duxi90  
  name: Dusan Jovanovic  
  email: dusan@djovanovic.me
```


Formati serijalizacije

❖ Tekstualni

- CSV
- XML
- JSON
- YAML
- **TOML**
- Property List (plist)
- EDN
- ...

```
[user]
uid = 12359
username = "duxi90"
name = "Dusan Jovanovic"
email = "dusan@djovanovic.me"
```

Formati serijalizacije

❖ Tekstualni

- S-expressions
- SQL Query
- HTTP
 - URI path
 - Query parameters
 - Header fields
- Command Line Interface

C++ demo

❖ Početak:

- [Sandbox](#)

❖ Rezultati:

- [Deserijalizacija objekta](#)
- [Deserijalizacija niza objekata](#)
- [Deserijalizacija kroz biblioteku](#)
- [Deserijalizacija celog dokumenta](#)

Introspekcija

Introspection

the examination of one's own conscious thoughts and feelings.

- Wikipedia, 10 July 2022



Type Introspection

*the ability of a program to **examine** the **type** or **properties** of an object at runtime*

- Wikipedia, 10 July 2022

Reflection

*the ability of a process to **examine**, **introspect**, and **modify** its own structure and behavior.*

- Wikipedia, 10 July 2022

Type Introspection

*the ability of a program to
examine the **type** or
properties of an object at
runtime*

- Wikipedia, 10 July 2022

~~Reflection~~

~~*the ability of a process to
examine, **introspect**, and
modify its own structure and
behavior.*~~

~~— Wikipedia, 10 July 2022~~

Type Introspection

*the ability of a program to
examine the **type** or
properties of an object at
~~runtime~~ **compiletime***

- Wikipedia, 10 July 2022

~~Reflection~~

~~*the ability of a process to
examine, **introspect**, and
modify its own structure and
behavior.*~~

~~— Wikipedia, 10 July 2022~~

Python - Runtime introspekcija

```
class Greeter:
    def __init__(self, name):
        self.name = name

    def greet(self):
        print("Hello ", self.name, "!")

print(dir(Greeter("Foo")))

print([
    method for method in dir(Greeter("Foo"))
    if not method.startswith('_')
])
```

```
[
    '__class__', '__delattr__', '__dict__', '__dir__',
    '__doc__', '__eq__',
    '__format__', '__ge__', '__getattr__',
    '__gt__', '__hash__',
    '__init__', '__init_subclass__', '__le__',
    '__lt__', '__module__',
    '__ne__', '__new__', '__reduce__', '__reduce_ex__',
    '__repr__', '__setattr__', '__sizeof__', '__str__',
    '__subclasshook__', '__weakref__', 'greet', 'name'
]

['greet', 'name']
```

C++ - Compiletime introspekcija

```
#include <string>
#include <fmt/ostream.h>

struct Greeter
{
    std::string name;

    void greet() { fmt::print("Hello {}!", name); }
};

static_assert(requires(Greeter g) { g.greet(); });
static_assert(requires(Greeter g) { g.name; });
```

Introspekcija tipova

❖ Otkrivanje tipa

- Jednostavno u compiletime-u

❖ Otkrivanje osobina tipa

- Takođe jednostavno
- Problem je iskazati ih

C++ i introspekcija

❖ Jezik

- [rtti - typeid, std::type_info](#)
- [Type traits](#)
- [Concepts i constraints](#)
- [P1240 Scalable Reflection in C++](#)

❖ Biblioteke

- [refl-cpp](#)
- [Magic Get \(boost.PFR\)](#)
- [nameof](#)
- [ctti](#)

Serijalizacija i introspekcija

Kako?

❖ Otkrivanje forme

- Generički prolaz kroz strukturu podataka

❖ Transformacija

- Informacija o tipu

Zašto?

❖ Generički kod

- Smanjenje koda - inače postoji kombinatorna eksplozija
- Uklanjanje priliku da se javi greška

❖ Eksplicitna definicija osobina

- Varijabilna ili fiksna velicina
- Inline polje
- Formatiranje pri serijalizaciji (camelCase, snake_case, ...)

❖ Odvaja definiciju od implementacije

- Objekat ne zna u šta može da se serijalizuje
- Serijalizacija ne zna za sve objekte

Postojeće C++ implementacije

- ❖ ORM biblioteke
 - [ODB](#)
- ❖ Game Engines & GUI Frameworks
 - [Unreal Engine](#)
 - [Entt](#)
 - [Qt Meta Objects](#)
- ❖ Data serialization biblioteke
 - [Thrift](#)
 - [nlohmann::json](#)

C++ demo

❖ Sandbox

➤ [Serijalizacija bez introspekcije](#)

❖ Results:

➤ [Introspekcija i serijalizacija](#)

C++ source #1

Save/Load Add new... Vim CppInsights Quick-bench C++

```
1 #include <stdint>
2 #include <string>
3 #include <tuple>
4
5 #include <fmt/ostream.h>
6
7 #include <nlohmann/json.hpp>
8
9 struct User
10 {
11     std::uint64_t uid;
12     std::string username;
13     std::string name;
14     std::string email;
15
16     auto getFields() {
17         return std::tuple{
18             std::pair{"uid", std::ref(uid)},
19             std::pair{"username", std::ref(username)},
20             std::pair{"name", std::ref(name)},
21             std::pair{"email", std::ref(email)},
22         };
23     }
24 };
25
26 template<typename T>
27 void from_json(const ::nlohmann::json& j, T& u) {
28     std::apply([&j](auto &&...fields) {
29         (j.at(fields.first).get_to(fields.second), ...);
30     }, u.getFields());
31 }
32
33 int main ()
34 {
35     using namespace std::literals;
36
37     auto input = R"(
38     {
39         "users": [
```

x86-64 clang 14.0.0 (C++, Editor #1, Compiler #1) Output of x86-64 clang 14.0.0 (Compiler #1)

A Wrap lines Select all

<source>:30:10: error: no member named 'getFields' in 'std::vector<User>'
 }, u.getFields());
 ^
/opt/compiler-explorer/libs/nlohmann_json/v3.6.0/single_include/nlohmann/json.hpp:1629:16: note: in instantiation of function template specialization 'std::vector<User>::get'
 return from_json(j, val);
 ^
/opt/compiler-explorer/libs/nlohmann_json/v3.6.0/single_include/nlohmann/json.hpp:2186:9: note: in instantiation of function template specialization 'nlohmann::from_json(std::forward<BasicJsonType>(j), val);'
 ::nlohmann::from_json(std::forward<BasicJsonType>(j), val);
 ^
/opt/compiler-explorer/libs/nlohmann_json/v3.6.0/single_include/nlohmann/json.hpp:15360:36: note: in instantiation of function template specialization 'nlohmann::json::JSONSerializer<ValueType>::from_json(*this, ret);'
 JSONSerializer<ValueType>::from_json(*this, ret);
 ^
<source>:60:34: note: in instantiation of function template specialization 'nlohmann::basic_json<>::get<std::vector<User>>'
 auto users = parsed["users"].get<std::vector<User>>();
 ^
1 error generated.
ASM generation compiler returned: 1
<source>:30:10: error: no member named 'getFields' in 'std::vector<User>'
 }, u.getFields());
 ^
Executer x86-64 clang 14.0.0 (C++, Editor #1)
A Wrap lines Libraries (2) Compilation Arguments Stdin Compiler output
x86-64 clang 14.0.0 -std=c++2b -O2
Could not execute the program
Compiler returned: 1
Compiler stderr
<source>:30:10: error: no member named 'getFields' in 'std::vector<User>'
 }, u.getFields());
 ^
/opt/compiler-explorer/libs/nlohmann_json/v3.6.0/single_include/nlohmann/json.hpp:1629:16: note: in instantiation of function template specialization 'std::vector<User>::get'
 return from_json(j, val);
 ^
/opt/compiler-explorer/libs/nlohmann_json/v3.6.0/single_include/nlohmann/json.hpp:2186:9: note: in instantiation of function template specialization 'nlohmann::from_json(std::forward<BasicJsonType>(j), val);'
 ::nlohmann::from_json(std::forward<BasicJsonType>(j), val);
 ^
/opt/compiler-explorer/libs/nlohmann_json/v3.6.0/single_include/nlohmann/json.hpp:15360:36: note: in instantiation of function template specialization 'nlohmann::json::JSONSerializer<ValueType>::from_json(*this, ret);'
 JSONSerializer<ValueType>::from_json(*this, ret);
 ^
<source>:60:34: note: in instantiation of function template specialization 'nlohmann::basic_json<>::get<std::vector<User>>'
 auto users = parsed["users"].get<std::vector<User>>();
 ^
x86-64 clang 14.0.0 - 3285ms

Read the new cookie policy Compiler Explorer uses cookies and other related techs to serve you

Don't consent Consent

C++ demo

❖ Sandbox

- Serijalizacija bez introspekcije

❖ Results:

- Introspekcija i serijalizacija - loše
- Introspekcija i serijalizacija - ne radi
- Introspekcija i serijalizacija
- Introspekcija i serijalizacija - bolje sa makroima

Kuda dalje?

Generalizacija

- ❖ Iteracija
 - [forEachField](#)
- ❖ Podrška za nove tipove serijalizacije
 - Rad sa generisanim tipovima (ProtoBuff, SBE, ...)
- ❖ Atributi
 - Inline
 - Required
 - Description
 - Classified / Confidential / Sensitive
- ❖ Generisanje koda

Reference

- ❖ Petar Trifunović (petar.trifunovic@inceptive.io)
- ❖ Andrew Sutton ([Reflection: Compile-Time Introspection of C++](#))
- ❖ Bjarne Stroustrup ([The Beauty and Power of “Primitive” C++](#))
- ❖ Martin Thompson ([Interaction Protocols: It's All About Good Manners](#))
- ❖ Mike Acton ([Data-Oriented Design and C++](#))

Demo Repository

