NJSZK Tigers CAP'N dokumentáció

Konvertáláshoz használt tool

A konvertáláshoz a hivatalos oldalról letöltött Windowsos csomag tartalma elég, amiből nekünk konkrétan csak a capnp.exe szükséges. Ezen felül egy séma fájl kell, ami alapján szövegesből bináris, vagy binárisból szöveges adatot tudunk csinálni.

A konvertálás parancssoros művelet; a karakterkódolásos baszakodás miatt sajnos nem tudtam máshogy megoldani, csak az alábbi parancssoros hívás segítségével:

capnp.exe <encode/decode> <SchemaFile.capnp> <SchemaName> <InputFile >OutputFile

Jól látszik, hogy miért játszik itt szerepet a parancssoros megoldás: a stdin és stdout átirányítása miatt. A capnp.exe erről olvassa be és ide írja ki a konvertálás eredményét.

A sémafájlokkal nem kell kvázi foglalkoznunk, mivel azokat mindig készen kapjuk. A weben elérhető a sémafájlok felépítése, de elég hosszú, szerintem fölösleges azzal baszakodni.

A szöveges bemeneti adatok az alábbi (közelítőleg, a megfigyelésem alapján):

- A szöveges értékadás így néz ki: valtozo = "ertek"
- A számok értékadása: valtozo = ertek
- Az összetett adatstruktúrák, classok így néznek ki: (valt1 = "a", valt2 = 23)
- Ez a fenti zárójeles szarakodás bármilyen mélységig mehet
- Az union típusú struktúrák esetén az union egyik tagja lehet csak az adatszerkezeten belül (lsd doksi)
- És még amit vagy nem írtam vagy én sem tudok róla

Ez alapján képesek vagyunk kapott bináris adatokból szöveges adatot készíteni, amiből nekem már csak Javában egy parser osztályt kell csinálnom, ami globálisan képes kezelni az ilyen adatszerkezeteket, és szövegesen lehet belőle lekérni bármilyen mélységig adatot. Én ezt így képzelem el:

```
CAPNPStruct s = new CAPNPStruct("(t1 = 1, t2 = "234", t3 = ( t4 = 34, t5 = "012" )) ");
string s1 = s.get("t2");
int i1 = s.get("t1");
```

A többszintű változók esetén vagy így:

```
CAPNPStruct s2 = s.get("t3");
int i2 = s2.get("t4");
```

vagy csak így:

```
int i3 = s.get("t3.t4");
```

Amennyiben a Java megengedi, dinamikus tulajdonságneveket is fogok lehetővé tenni, de szerintem ez csak kavarcot okozna.

Érdekesség: a szöveges ábrázolásnál a nem string literalban található szóközök száma mindegy.