



PROGRAMIRANJE

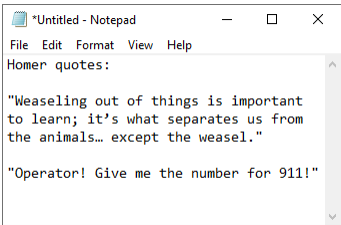
*Predavanje 15 – Binarne datoteke

Ishod učenja 6

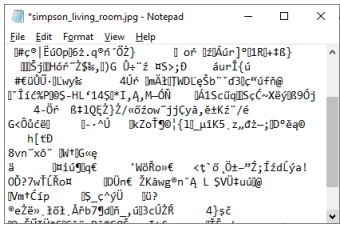
1

Uvod

- Tekstualne datoteke koriste linije za čuvanje slova, znamenke i druge znakove koji su čitljivi čovjeku
- Binarne datoteke sadrže bajtove
 - Nema linija
 - Nije čitljivo čovjeku




Homer.txt



simpson_living_room.jpg

- Je li Word dokument tekstualna ili binarna datoteka?

Strana • 2



2

Otvaranje binarne datoteke

- Moramo poslati dodatni parametar u konstruktore za `ifstream` and `ofstream`
- Taj dodatni parametar mora biti `ios_base::binary`
 - `::` je operator dosega – kaže da se desna strana nalazi unutar lijeve strane
 - U stvari je cjelobrojna konstanta vrijednosti 32

```
ifstream dat1("file1.bin", ios_base::binary);
ofstream dat2("file2.bin", ios_base::binary);
```
- Također moramo provjeriti uspjeh i zatvoriti kad smo gotovi
 - Jednako kao i kod tekstualnih datoteka

Strana • 3



3

Pretvaranje tipova podataka

- Pretvaranje tipova podataka (engl. *cast*) je proces pretvaranja podataka iz jednog tipa u drugi
- Postoje dva tipa konverzije:
 - Implicitna konverzija
 - Događa se automatski od strane kompajlera, bez intervencije programera

```
int a = 10;
cout << a + 5.5 << endl;
```
 - Eksplicitna konverzija
 - Događa se kad to programer zatraži

```
int a = 10;
cout << (double)a + 5.5 << endl;
```

Strana • 4



4

PISANJE U BINARNU DATOTEKU

Strana • 5



5

Pisanje u binarnu datoteku

- Pisanje u binarnu datoteku podrazumijeva kopiranje dijela radne memorije u datoteku na disku
- Za pisanje nam treba pokazivač na **prvi bajt** te **ukupan broj bajtova** koje želimo zapisati
 - Problem: ne postoji tip podataka byte
 - Rješenje: pošto je tip podataka char uvijek velik točno 1 bajt, koristit ćemo njega
 - Za naše potrebe ćemo smatrati char = byte
 - Time vrijedi i ovo: char* = byte* (pokazivač na bajt)

Strana • 6



6

Pisanje u binarnu datoteku

- Za pisanje koristimo metodu `write()` na objektu tipa `ofstream`

```
dat.write(pokazivač_na_početak, veličina);
```

- Parametri:

- *pokazivač_na_početak* je tipa `char*`

- Predstavlja adresu komada memorije iz kojeg želimo zapisati podatke u datoteku

- *veličina* je tipa `int`

- Predstavlja broj bajtova (tj. veličinu komada memorije) koji želimo prepisati u datoteku

Strana • 7



7

Pisanje u binarnu datoteku

- Ovako pišemo u binarnu datoteku:

```
int n1 = 85;
dat1.write((char*)&n1, sizeof(n1));
```

```
double n2 = 15.5;
dat1.write((char*)&n2, sizeof(n2));
```

- Ovaj pristup se primjenjuje na sve ugrađene tipove podataka

Strana • 8



8

ČITANJE IZ BINARNE DATOTEKE

Strana • 9



9

Čitanje iz binarne datoteke

- Čitanje iz binarnih datoteka je isti postupak, samo u suprotnom smjeru
 - Pripremimo prazni blok u memoriji
 - Prenešemo podatke iz datoteke u taj blok
- Koristimo metodu `read()` na objektu tipa `ifstream`
`dat.read(pokazivač_na_početak_bloka, veličina);`

Strana • 10



10

Čitanje iz binarne datoteke

- Ovako čitamo iz binarne datoteke:

```
int n1;  
dat.read((char*)&n1, sizeof(n1));
```

```
double n2;  
dat.read((char*)&n2, sizeof(n2));
```

- Ovaj pristup se primjenjuje na sve ugrađene tipove podataka

Strana • 11



11

ČITANJE I PISANJE STRINGOVA IZ/U BINARNU DATOTEKU

Strana • 12



12

Uvod

- Za primitivne tipove točno znamo koliko bajtova zauzimaju
- Stringovi su objekti:
 - Svaki string ima proizvoljni broj znakova (varijabilna duljina)
 - Dio stringa se nalazi na stogu, dio na hrpi
- Rješenje:
 - Kod zapisivanja u datoteku ćemo uvijek prvo zapisati broj znakova u stringu, a tek onda sâm string
 - Kod čitanja iz datoteke ćemo prvo pročitati broj znakova u stringu, a tek onda sâm string:
 - Rezerviramo dovoljno memorije na hrpi i u nju učitamo znakove
 - Od tih znakova konstruiramo novi string

Strana • 13



13

Pisanje stringa u binarnu datoteku

- String ima metodu `c_str()` koja vraća `char*`:

```
// String to write
string s = "Miro";

// Number of characters (bytes)
int n = s.length();

// First write number of characters
dat1.write((char*)&n, sizeof(n));

// Then write characters
dat1.write(s.c_str(), n);
```

Strana • 14



14

Čitanje stringa iz binarne datoteke

▪ Postupak:

- Prvo pročitamo duljinu stringa
- Alociramo dovoljno memorije na hrpi
- Učitamo iz datoteke znakove na hrpu – to još nije string!
- Iskoristimo konstruktor klase string da bismo konstruirali novi string:


```
string s = string(pchar, n);
```
- Otpustimo memoriju s hrpe

Strana • 15



15

Čitanje stringa iz binarne datoteke

```
// Read string length
int n;
dat.read((char*)&n, sizeof(n));

// Allocate storage for characters
char* pchar = new char[n];

// Read characters
dat.read(pchar, n);

// Construct a new string
string s = string(pchar, n);

// Release memory
delete[] pchar;
```

Strana • 16



16

Primjeri

1. Spremite neki cijeli broj u binarnu datoteku.
2. Učitajte od korisnika 5 cijelih brojeva u polje pa ih zatim spremite u binarnu datoteku.
3. Učitajte od korisnika 5 decimalnih brojeva u polje pa ih zatim spremite u binarnu datoteku.
4. Učitajte od korisnika 5 imena pa ih onda spremite u binarnu datoteku.
5. Zadana je struktura Film (naziv, prosječna ocjena).
Napravite polje od 3 filma s nekim vrijednostima.
Spremite sadržaj polja u binarnu datoteku.

Strana • 17



17

Primjeri

U prilogu je datoteka "filmovi.bin" koja sadrže podatke o filmovima sa strukturom:

Duljina u bajtovima	Tip podataka	Opis
4	int	n = duljina naziva filma
n	string	naziv filma
4	int	m = duljina popisa glumaca
m	string	popis glumaca
4	int	trajanje filma
4	unsigned int	žanr (oznaka prema tablici žanrovi)
8	double	zarada filma

6. Prikažite sve filmove.

Strana • 18



18



19