



Napredne tehnike programiranja

Vježba 3



Teme



- Prikaz podataka u binarnim datotekama
- Funkcije za rad sa binarnim datotekama
- Rad sa binarnim datotekama
- Organizacija podataka u datotekama

Tipovi datoteka



- Tekstualne datoteke - podaci se zapisuju formatirano kao niz ASCII znakova i čitljive su u editoru
- Binarne datoteke – podaci se zapisuju u istom obliku kako su zapisane u memoriji (neformatirano) i nisu čitljive u editoru

Tekstualne (formatirane) datoteke



- Svaki podatak se u datoteku zapisuje kao niz alfanumeričkih znakova

Podatak u memoriji	Podatak u datoteci
00000000 00000000 00000000 00000111	00110111
integer 7	ASCII kod znaka '7'

Binarne (neformatirane)datoteke



- Svaki podatak se u datoteku zapisuje kao što je zapisan u memoriji

Podatak u memoriji	Podatak u datoteci
00000000 00000000 00000000 00000111	00000000 00000000 00000000 00000111
integer 7	integer 7

Binarne datoteke



- Kreiraju se otvaraju i koriste isto kao i tekstualne datoteke
- Kod otvaranja potrebno je staviti parametar `ios::binary`
`npr: ofstream tok("datoteka.txt", ios::binary)`
- U radu s binarnim datotekama koriste se funkcije `read()` i `write()`.

Pisanje u binarnu datoteku



- Za pisanje u binarnu datoteku koristimo funkciju write:

```
write ( adresa, veličina );
```

- Prvi argument je pokazivač na char
- Drugi argument kaže koliko bajtova zauzima taj podatak

Pisanje u binarnu datoteku



- Za pisanje u binarnu datoteku koristimo funkciju write:

```
write ( adresa, veličina );
```

- Prvi argument je pokazivač na char
- Drugi argument kaže koliko bajtova zauzima taj podatak

Pisanje u binarnu datoteku



```
struct Osoba{
    char ime[50];
    int starost;
    char OIB[14];
};

int main()
{
    Osoba ja= {"Pero", 40, "123456789"};
    Osoba book[30];
    int x = 123;
    double fx = 34.54;
    ofstream outfile;
    outfile.open("datoteka.dat", ios::binary | ios::out);
    outfile.write((char *)&x, sizeof (int)); // sizeof tipa
    outfile.write((char *)&fx, sizeof(fx)); // sizeof varijable
    outfile.write((char *)&ja, sizeof(ja));
    outfile.write((char *)book, 30*sizeof(Osoba));
    outfile.close();
}
```

Čitanje iz binarne datoteke



- Za čitanje iz binarne datoteke koristimo funkciju `read`:

```
read ( adresa na koju spremamo, veličina);
```

- Prvi argument je pokazivač na `char`
- Drugi argument kaže koliko bajtova zauzima taj podatak

Čitanje iz binarne datoteke



Primjer: Čita jedan cijeli broj iz datoteke i ispisuje ga na ekran

```
int x;  
ifstream infile;  
infile.open("nesto.dat", ios::binary | ios::in)  
infile.read((char *)&x, sizeof(int)); // cita jedan int  
cout<<x<<endl;
```

Rad sa datotekama



- Tokovi interno pamte trenutnu poziciju na kojoj se nalazimo
- Ulazni tok ima poziciju za čitanje, a izlazni poziciju za pisanje (dvosmjerni ima poziciju koja je istovremeno pozicija za čitanje i pisanje)
- pozicije možemo spremiti u podatak tipa `streampos`
- Kada želimo koristiti pozicije moramo kreirati (deklarirati) objekt tog tipa:

```
streampos pozicija;
```

Rad sa datotekama



Čitanje pozicija:

`tellg()` – vraća poziciju za čitanje

`tellp()` – vraća poziciju za pisanje

Postavljanje pozicija:

`seekg(pos)` – postavlja poziciju za čitanje na pos

`seekp(pos)` – postavlja poziciju za pisanje na pos

Rad sa datotekama



`seekg(pos, offset)` – postavlja poziciju za čitanje na pos mjesta od offset

`seekp(pos, offset)` – postavlja poziciju za pisanje na pos mjesta od offset

Referentne pozicije (offset):

`ios::beg` početak toka

`ios::cur` trenutna pozicija

`ios::end` kraj toka

Rad sa datotekama



Referentna pozicija	Objašnjenje
ios::beg	Referentna pozicija je početak datoteke. Ako se ne zada drugi argument tada se podrazumijeva ios::beg
ios::cur	Referentna pozicija je trenutna pozicija (koja se dobije s tellg()funkcijom).
ios::end	Referentna pozicija je kraj datoteke. Obično se za pomak koristi negativna vrijednost. Ako pomak ima pozitivnu vrijednost pozicijski indikator se postavlja iza kraja datoteke. Čitanje s te pozicije uzrokuje dojavu greške. Pisanje iza EOF je dozvoljeno. Ono se izvršava tako da se od trenutnog kraja datoteke do zadane pozicije upis dopunjuje s nul-znakom. Postavljanje pozicijskog indikatora prije početka datoteke nije dozvoljeno.

Rad sa datotekama



```
int main() {
    int x;
    streampos pos;
    ifstream infile;
    infile.open("file.dat", ios::binary | ios::in);
    infile.seekp(243, ios::beg); // ide 243 bajta naprijed od beg
    infile.read(&x, sizeof(x));
    pos = infile.tellg();
    cout << "Pokazivac citanja je na lokaciji:" << pos << endl;
    infile.seekp(0, ios::end); // ide na kraj datoteke
    infile.seekp(-10, ios::cur); // vraća se 10 bajtova
    infile.close();
}
```

Operacije read i write mjenjanju položaj pokazivača za čitanje i pisanje !!!

Binarne datoteke



- Kako pročitati binarnu datoteku u kojoj je struktura jedan po jedan član

```
ifstream fajl("datoteka.dat", ios::binary);  
int i = 0;  
fajl.seekg(0);  
  
while (!fajl.eof())  
{  
    fajl.read(struktura, sizeof(struktura)); //čitanje  
    cout<<struktura.neki_clan;  
}
```

Organizacija podataka u datotekama



- Podatke u datotekama nastojimo organizirati tako da rad sa njima bude što jednostavniji i brži

Sekvencijalne (slijedne) datoteke

datoteka

```
939 Graševina 36.47
938 Sauvignon 32.2
936 Rizling 30
934 Pinot 26.83
928 Traminac 34.64
926 Chardonnay 37.08
910 Malvazija 51.59
908 Jezero 25.49
906 Rose 16.46
900 Dolac 43.42
897 Postup 47.57
895 Plavac 22.68
893 Merlot 22.68
891 Faust 14.87
889 Jigsaw 48.79
887 Chianti 52.45
```

- Podaci slijede jedan iza drugog
- Nižu se redoslijedom unosa
- Nema praznih mjesta
- Do zapisa se dolazi čitajući jedan po jedan zapis dok ne nađemo traženi

Direktne datoteke



datoteka

2201	Graševina	36.47
2202	Sauvignon	32.2
2203		
2204	Rizling	30
2205	Pinot	26.83
2206	Traminac	34.64
2207		
2208	Chardonnay	37.08
2209	Malvazija	51.59
2210	Jezero	25.49
2211	Rose	16.46
2212	Dolac	43.42
2213	Postup	47.57
2214	Plavac	22.68
2215	Merlot	22.68
2216	Faust	14.87
2217	Jigsaw	48.79
2218	Chianti	52.45

- Svaki zapis ima jedinstveni ključ
- Ako znamo ključ direktno možemo pročitati traženi zapis
- Svi zapisi su iste duljine
- Ako ne postoji zapis sa nekim ključem kreiramo prazni zapis sa tim ključem
- Datoteke su veće nego što je to nužno za spremanje te količine podataka

Relativne datoteke



datoteka

00	2201	Graševina	36.47
01	2202	Sauvignon	32.2
02	2203	Rizling	30
03	2205	Pinot	26.83
04	2206	Traminac	34.64
05	2208	Chardonnay	37.08
06	2209	Malvazija	51.59
07	2210	Jezero	25.49
08	2211	Rose	16.46
09	2212	Dolac	43.42
10	2213	Postup	47.57
11			
12	2214	Plavac	22.68
13	2215	Merlot	22.68
14	2216	Faust	14.87
15	2217	Jigsaw	48.79
16	2218	Chianti	52.45

- Novi ključ (pozicija) se izračunava iz originalnog ključa
- Novi ključ je kraći od originalnog da se smanji broj praznih mjesta u datoteci
- Ako je pozicija zauzeta zapis se smješta na prvu slobodnu poziciju
- Svi zapisi su iste duljine
- Datoteka je manja od direktne, ali je pretraživanje malo sporije (ako ima kolizija)

Indeksne datoteke



- Indeksne datoteke su datoteke koje koristimo da bi ubrzali rad sa podacima
- U indeksnoj datoteci čuvamo ključ i adresu zapisa u datoteci sa podacima koja odgovara tom ključu
- Za jednu datoteku sa podacima možemo kreirati više indeksnih datoteka

Indeksne datoteke



- Indeksna datoteka mora biti sortirana po ključu (da brže pronađemo podatak)
- Indeksna datoteka je puno manja od datoteke sa podacima (pa je možemo držati u memoriji)

Indeksne datoteke (primjer)



- Imamo podatke spremljene u binarnu datoteku u slijedećem formatu:

```
struct vino{  
    int sifra; // 4 bytes  
    char naziv[16]; // 16 bytes  
    float cijena; //4 bytes  
};
```


Indeksne datoteke

- Recimo da želimo ubrzati rad preko šifre (duljina sloga je 24)

adrese datoteka sa podacima

000	2201	Graševina	36.47
024	2202	Sauvignon	32.2
048	2204	Rizling	30
072	2205	Pinot	26.83
096	2206	Traminac	34.64
120	2208	Chardonnay	37.08
144	2209	Malvazija	51.59
168	2210	Jezero	25.49
192	2211	Rose	16.46
216	2212	Dolac	43.42
240	2213	Postup	47.57
264	2214	Plavac	22.68
288	2215	Merlot	22.68
312	2216	Faust	14.87
336	2217	Jigsaw	48.79
360	2218	Chianti	52.45

indeksna datoteka

2201	000
2202	024
2204	048
2205	072
2206	096
2208	120
2209	144
2210	168
2211	192
2212	216
2213	240
2214	264
2215	288
2216	312
2217	336
2218	360

Indeksne datoteke

- Traženje zapisa sa ključem 2209

adrese	datoteka sa podacima			indeksna datoteka	
000	2201	Graševina	36.47	2201	000
024	2202	Sauvignon	32.2	2202	024
048	2204	Rizling	30	2204	048
072	2205	Pinot	26.83	2205	072
096	2206	Traminac	34.64	2206	096
120	2208	Chardonnay	37.08	2208	120
144	2209	Malvazija	51.59	2209	144
168	2210	Jezero	25.49	2210	168
192	2211	Rose	16.46	2211	192
216	2212	Dolac	43.42	2212	216
240	2213	Postup	47.57	2213	240
264	2214	Plavac	22.68	2214	264
288	2215	Merlot	22.68	2215	288
312	2216	Faust	14.87	2216	312
336	2217	Jigsaw	48.79	2217	336
360	2218	Chianti	52.45	2218	360

Indeksne datoteke

- Dodavanje zapisa: '2203 Teran 35.02'

adrese datoteka sa podacima

000	2201	Graševina	36.47
024	2202	Sauvignon	32.2
048	2204	Rizling	30
072	2205	Pinot	26.83
096	2206	Traminac	34.64
120	2208	Chardonnay	37.08
144	2209	Malvazija	51.59
168	2210	Jezero	25.49
192	2211	Rose	16.46
216	2212	Dolac	43.42
240	2213	Postup	47.57
264	2214	Plavac	22.68
288	2215	Merlot	22.68
312	2216	Faust	14.87
336	2217	Jigsaw	48.79
360	2218	Chianti	52.45
384	2203	Teran	35.02

indeksna datoteka

2201	000
2202	024
2203	384
2204	048
2205	072
2206	096
2208	120
2209	144
2210	168
2211	192
2212	216
2213	240
2214	264
2215	288
2216	312
2217	336
2218	360

Indeksne datoteke

- Brisanje zapisa: '2213 Postup 47.57'

adrese datoteka sa podacima

000	2201	Graševina	36.47
024	2202	Sauvignon	32.2
048	2204	Rizling	30
072	2205	Pinot	26.83
096	2206	Traminac	34.64
120	2208	Chardonnay	37.08

I slog u podatkovnoj datoteci se može označiti kao obrisani, a periodički se takvi slogovi mogu i fizički brisati iz datoteke

240	2213	Postup	47.57
264	2214	Plavac	22.68
288	2215	Merlot	22.68
312	2216	Faust	14.87
336	2217	Jigsaw	48.79
360	2218	Chianti	52.45
384	2203	Teran	35.02

indeksna datoteka

2201	000
2202	024
2203	384
2204	048
2205	072
2206	096
2208	120
2209	144
2210	168
2211	192
2212	216
2213	240
2214	264
2215	288
2216	312
2217	336
2218	360

Indeksne datoteke

- Kreiranje indeksne datoteke za ključ naziv

adrese datoteka sa podacima

000	2201	Graševina	36.47
024	2202	Sauvignon	32.2
048	2204	Rizling	30
072	2205	Pinot	26.83
096	2206	Traminac	34.64
120	2208	Chardonnay	37.08
144	2209	Malvazija	51.59
168	2210	Jezero	25.49
192	2211	Rose	16.46
216	2212	Dolac	43.42
240	2213	Postup	47.57
264	2214	Plavac	22.68
288	2215	Merlot	22.68
312	2216	Faust	14.87
336	2217	Jigsaw	48.79
360	2218	Chianti	52.45
384	2203	Teran	35.02

indeksna datoteka

2201	000
2202	024
2203	384
2204	048
2205	072
2206	096
2208	120
2209	144
2210	168
2211	192
2212	216
2213	240
2214	264
2215	288
2216	312
2217	336
2218	360

indeksna datoteka 2

Chardonnay	120
Chianti	360
Dolac	216
Faust	312
Graševina	000
Jezero	168
Jigsaw	336
Malvazija	144
Merlot	288
Pinot	072
Plavac	264
Postup	240
Rizling	048
Rose	192
Sauvignon	024
Teran	384
Traminac	096

Višestruke indeksne datoteke



- Kad radimo sa više indeksnih datoteke možemo podacima brzo pristupati preko različitih ključeva
- Nedostatci:
 - Kod svake promjene treba ažurirati sve indeksne datoteke
 - Troši se više memorije
 - Različiti tipovi ključa različito brzo rade

Indeksne datoteke



- Zašto su bolje od relativno organiziranih datoteka:
 - datoteka sa podacima je manja
 - brži dohvat podatka (nema kolizija)
 - moguće korištenje više ključeva (više indeksnih datoteka)

Zadatak



- Napisati program koji radi sa listom vina u binarnoj datoteci listavina.dat
- Za svako vino čuvaju se slijedeći podaci:
 - šifra
 - naziv
 - cijena
- Datoteka listavina.dat i predložak zadatka se nalaze na e-učenju

Zadatak



Potrebno je dopisati slijedeće funkcionalnosti u program:

- Učitavanje sadržaja datoteke u listu i ispis ga na zaslon
- Dodavanje novog vina na listu (i u datoteku)
- Izmjena cijene vina (u listi i datoteci)
- Brisanje vina sa liste (u listi i datoteci)
- Sortiranje liste vina po sifri (u listi i datoteci)
- Zamjena redoslijede dva vina u listi (i datoteci)

Bodovanje:

- 3 funkcije (1 bod)
- 4 funkcije (2 boda)
- 5 ili više funkcija (3 boda)