

Kako radi Zip? Kompresija na delu!

Marko Petrović

Vrlo korisni alati koje ste verovatno do sada koristili mnogo puta su alati za komprimovanje i arhiviranje podataka. Najpopularniji od njih je verovatno *WinZip* čije DOS verzije se sigurno sećaju "stariji korisnici". Osim njega, u upotrebi su i RAR, ARJ... Svi oni omogućuju da se smanji prostor koji neki podaci zauzimaju na disku. Ranije je njihova upotreba bila još izraženija jer su diskovi bili daleko manjeg kapaciteta, dok se veličina prosečnih dokumenata nije drastično menjala. Bez obzira na činjenicu drastičnog smanjenja cena CD pisaača i medija, povećanja kapaciteta diskova, prodora DVD tehnologije, upotreba alata za komprimovanje podataka ne gubi na značaju. I dalje je npr. veoma bitno da se što više smanji veličina podataka koji se prenose, recimo preko Interneta. Osim toga, komprimovanje podataka je veoma rasprostranjeno ne samo u računarskoj tehnici, već i u telekomunikacijama (mobilna telefonija, prenos slike, zvuka...) i drugim sferama.

Kako funkcioniše kompresija?

Svaki postupak komprimovanja podataka se oslanja na matematiku. Ako ste tokom školovanja izbegavali ovaj predmet, slobodno nastavite sa čitanjem jer ćemo sve objasniti pomoću adekvatnih i "pitkih" primera.

Bez obzira na sadržaj nekog dokumenta (tekst, slika, muzika, film, itd.), u računaru je zapisan u binarnom obliku (nule i jedinice). Sama kompresija jednostavno rečeno, smanjuje veličinu dokumenta time što smanjuje potreban broj nula i jedinica za zapis dokumenta. Npr. ako je na početku bilo potrebno 300 nula i jedinica, nakon primene postupka za komprimovanje biće ih potrebno nekoliko puta manje.

Stepen kompresije

Stepen kompresije je odnos veličina originalnog (nekomprimovanog) dokumenta i njegove komprimovane verzije. Taj stepen nije neka fiksna veličina, već zavisi od više parametara. Vrsta dokumenta je recimo jedan od njih. Verovatno ste primetili da se npr. Word-ovi dokumenti drastično smanje nakon "zipovanja", dok slike uglavnom ostaju iste veličine.

Sadržaj dokumenta takođe utiče na stepen kompresije. U sledećem primeru smo kreirali tri tekstualna dokumenta iste veličine (istog broja slova) u *Windows*-ovom programu *Notepad*, a zatim ih komprimovali već pomenutim *WinZip*-om.

Dokument	Veličina (bajt)	Stepen kompresije
Originalni dokument (nekomprimovan)	982	1
Komprimovan, sastavljen od svih istih slova	133	7.38
Komprimovan, sastavljen od jedne reči koja je ponovljena više puta	138	7.12
Komprimovan, sastavljen od različitih slova	608	1.62

Sadržaj nekomprimovanog dokumenta nije bitan, jer svako slovo zauzima istu veličinu (1 bajt).

Kao što se vidi iz tabele, najveći stepen kompresije ima dokument koji u sebi ima sva ista slova. Iznad njega je dokument koji ima jednu reč ponovljenu više puta. Ovde je stepen kompresije nešto niži, dok je najmanji za dokument gde nije zastupljeno neko ponavljanje, već se dokument sastoji od slučajno dobijenih slova.

Najbolju statistiku ima prvi, a najlošiju poslednji dokument. Za kucani tekst iz realnog života, gde se mogu pojaviti iste reči, očekivani stepen kompresije je negde između stepena kompresije drugog i trećeg dokumenta.

Zašto je to tako?

Da bismo odgovorili na ovo pitanje moramo objasniti način na koji *WinZip* funkcioniše. Kao što je već rečeno, matematika je u pozadini ovakvih postupaka. Procedura, odnosno algoritam koji se u tom slučaju primenjuje je Hafmanov (Huffman) algoritam.

Hafmanov algoritam

Postoje mnoge verzije ovog algoritma, kako jednostavne, tako i kompleksnije. On pripada grupi algoritama sa promenljivom dužinom kodne reči. To znači da se pojedini simboli (u našem slučaju slova) predstavljaju nizovima bitova (kodnim rečima), koji su različite dužine. Dakle, kodne reči su nizovi nula i jedinica različite dužine.

Karakteristika ovog algoritma je da smanjuje nepotrebno ponavljanje simbola i na taj način omogućuje kompresiju podataka. To smanjenje se zasniva na različitim verovatnoćama različitih simbola. Simboli koji imaju veću verovatnoću pojavljivanja se koduju kraćim kodnim rečima, a simboli koji imaju manju verovatnoću pojavljivanja se koduju dužim kodnim rečima.

Kao primer prikazaćemo kompresiju (kodiranje) reči "OMEGAMAGAZIN". Za skup znakova (slova) koristićemo samo slova koja se koriste u našem primeru. Jedno slovo zauzima jedan bajt, a jedan bajt sadrži 8 bitova. Naša reč ima 12 bajtova, tj. 96 bitova. Pošto naša reč ima 12 slova, lako se izračunava da je verovatnoća pojavljivanja jednog slova $1/12$ ili 0.083. Npr. novčić ima dve različite strane (pismo i glava) i verovatnoća da će se prilikom bacanja dobiti pismo ili glava je $1/2$.

Krenimo na posao!

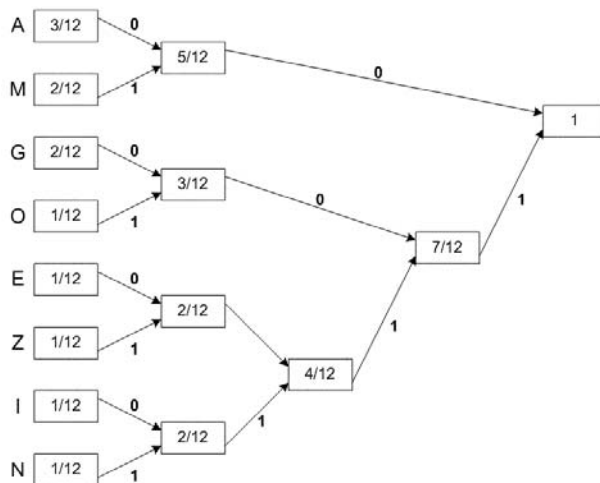
Prvo se prebroji koliko puta se pojavljuje određeno slovo i verovatnoća pojavljivanja se pomnoži sa tim brojem. Zatim se napravi tabela:

Slovo	Broj ponavljanja	Verovatnoća
O	1	0.083
M	2	0.166
E	1	0.083
G	2	0.166
A	3	0.249
Z	1	0.083
I	1	0.083
N	1	0.083

Postupak (algoritam) kodiranja je sledeći:

1. Poređaju se simboli po opadajućim verovatnoćama
2. Povežu se simboli sa najmanjim verovatnoćama u novi simbol, čija je verovatnoća pojavljivanja zbir verovatnoća ta dva simbola
3. Korak 2. se ponavlja sve dok se ne generiše simbol čija je verovatnoća 1
4. Pravi se kodno drvo (ili tabela), i u njemu se donjim granama dodeli vrednost "1", a gornjim "0" (binarni brojevi)

Sada pravimo stablo:



Sa drveta se dobijaju kodovi za pojedina slova na sledeći način: krenemo iz korena drveta (mesto u kojem je verovatnoća 1) prema željenom simbolu i jednostavno pročitamo brojeve na granama drveta preko kojih se krećemo. Tim postupkom dobijamo sledeću tabelu:

Slovo	Verovatnoća	Kod	broj bitova
A	0.249	00	2
M	0.166	01	2
G	0.166	100	3
O	0.083	101	3
E	0.083	1100	4
Z	0.083	1101	4
I	0.083	1110	4
N	0.083	1111	4

Potvrđena je činjenica da se slova sa najvećom verovatnoćom pojavljivanja koduju sa najmanjom dužinom kodne reči (u našem slučaju slovo "A" - 2 bita), a slova sa najmanjom verovatnoćom pojavljivanja se koduju sa 4 bita. Ako se naš primer predstavi u kodovanom obliku (OMEGAMAGAZIN), videćemo da je on veličine 35 bitova. Ovaj broj se dobije ako saberemo sve nule i jedinice potrebne da se ispišu sva slova našeg primera korišćenjem Hafmanovog koda. Ako sada podelimo broj bitova sa početka priče (96) koji je bio neophodan za prikaz nekodirane reči, dobija se da je stepen kompresije 2.74.

Hafmanov algoritam se još primenjuje i u kompresiji slika (npr. JPEG). Sigurno ste se nekad zapitali kako se neki dokumenti mogu veoma dobro komprimovati, dok drugi ne. Odgovor je vrlo jednostavan: zato što su ti dokumenti već komprimovani na neki način. To se naročito vidi, ako pokušate da komprimujete JPEG slike, MPEG video *clip*-ove, MP3 pesme, i sl. uz pomoć *WinZip*-a. Nemoguće je komprimovati komprimovane dokumente jer je njihova statistika već "istrošena".