S NAŠIH FAKULTETA

Prikaz cijelih brojeva u računalu – predavanje s prve godine studija

Goranka Nogo[1], *Saša Singer*[2]

Kolegij Programiranje 1 obvezni je kolegij za studente prve godine preddiplomskog sveučilišnog studija Matematika te izborni kolegij za studente preddiplomskog sveučilišnog studija Matematika; smjer: nastavnički. Nakon položenog kolegija studenti, između ostalog, mogu objasniti prikaz cijelih i realnih brojeva u računalu te su svjesni problema koji nastaju kao posljedica činjenice da su prikazivi skupovi brojeva konačni. U ovom članku bit će prikazano jedno predavanje na kolegiju Programiranje 1.

Na početku, prisjetimo se osnovnog problema: standardni skupovi brojeva $\mathbb{N}$, $\mathbb{Z}$ i $\mathbb{R}$ su beskonačni te ih stoga ne možemo prikazati u računalu. Umjesto toga, u računalu možemo prikazati samo neke konačne podskupove (modele) odgovarajućeg matematičkog skupa. Numeričke tipove podataka možemo podijeliti u tri grupe, prema beskonačnom skupu kojeg modeliramo:

- "cijeli" brojevi bez predznaka – model za $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$
- "cijeli" brojevi s predznakom – model za $\mathbb{Z}$
- "realni" brojevi – model za $\mathbb{R}$.

Navodnici naglašavaju da su prikazivi skupovi brojeva konačni. Svaka grupa može imati nekoliko podgrupa ovisno o broju bitova predviđenom za prikaz.

Pretpostavimo da imamo n bitova za prikaz broja. Tipične vrijednosti za n su 8, 16, 32 i 64. U n bitova možemo prikazati točno 2^n različitih podataka. Dakle, skup prikazivih brojeva ima najviše 2^n elemenata.

Prikaz cijelih brojeva bez predznaka

Cijeli brojevi bez predznaka modeliraju skup $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Dogovor je da se u računalu prikazuje najveći početni komad tog skupa. Dakle, skup svih prikazivih cijelih brojeva bez predznaka jednak je $\mathbb{Z}_{2^n} = \{0, 1, 2, \ldots, 2^n - 2, 2^n - 1\}$. Najveći prikazivi cijeli broj bez predznaka jednak je $2^n - 1$. Veće brojeve ne možemo prikazati koristeći n bitova. Tipične vrijednosti za najveći prikazivi broj bez predznaka su

n	$2^n - 1$
8	255
16	65 535
32	4 294 967 295
64	18 446 744 073 709 551 615

Kako stvarno izgleda prikaz brojeva u tih n bitova? Prikaz (prikazivog) broja je doslovna kopija prikaza tog broja u pozicionom zapisu u bazi 2, uz dopunu nulama "sprijeda" do n bitova. Pogledajmo sljedeći primjer.

[1] Autorica je docent na MO PMF-a Sveučilišta u Zagrebu, e-pošta: nogo@math.hr

[2] Autor je izvanredni profesor na MO PMF-a Sveučilišta u Zagrebu, e-pošta: singer@math.hr

Primjer 1. Jednostavnosti radi, pretpostavimo da je $n = 8$ i pogledajmo zapis broja 59. Primijetimo da je broj 59 prikaziv jer je $59 < 255 = 2^8 - 1$.

$$59 = 32 + 16 + 8 + 2 + 1 = 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0.$$

Nakon dopune nulama dobivamo:

$$59 = 0 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0.$$

Dakle, broj 59, kao cijeli broj bez predznaka, ima prikaz

7	6	5	4	3	2	1	0
0	0	1	1	1	0	1	1

ili, $59 = [0\,0\,1\,1\,1\,0\,1\,1]$.

Aritmetika cijelih brojeva bez predznaka s n bitova za prikaz brojeva je tzv. modularna aritmetika. To znači da aritmetičke operacije $+$, $-$ i $\cdot$, na skupu $\mathbb{Z}_{2^n}$ cijelih brojeva bez predznaka, daju rezultat koji je jednak ostatku rezultata pripadne cjelobrojne operacije (u skupu $\mathbb{Z}$) pri dijeljenju s 2^n. Zašto se aritmetika cijelih brojeva realizira na ovaj način, tj. kao modularna aritmetika? Tehnički gledano, ova realizacija je najlakša i brza. U pozadini ove realizacije je klasična algebarska struktura prstena ostataka modulo 2^n (o tome ćete učiti kasnije).

Važna napomena. Ako je "pravi" cjelobrojni rezultat, npr. zbroj dva broja, prevelik (neprikaziv), računalo ne javlja nikakvu grešku.

Primijetite da dijeljenje do sada nismo spomenuli. Rezultat dijeljenja dva cijela broja ne mora biti cijeli broj. Zamjena za to "obično" dijeljenje u cijelim brojevima je tzv. dijeljenje s ostatkom. Uočite da cjelobrojno dijeljenje s ostatkom daje dva rezultata: cjelobrojni kvocijent i ostatak. Za cjelobrojni ostatak često se koristi oznaka *div* (u programskom jeziku C to je operator `/`), a za cjelobrojni koeficijent oznaka *mod* (operator `%` u C-u). Dakle, operacije cjelobrojnog dijeljenja s ostatkom *div* i *mod* daju iste rezultate kao da dijelimo u $\mathbb{N}_0$.

U programskom jeziku C cijelim brojevima bez predznaka odgovara tip koji se zove `unsigned int`, ili, skraćeno, `unsigned`. Ovaj tip, osim u standardnoj veličini, postoji i u veličinama `short`, `long` i `long long`, ovisno o broju bitova predviđenih za prikaz.

Primjer 2.

```
#include <stdio.h>
int main(void) {
   unsigned short i = 65535; /* n=16 */
   printf("%d\n", i / 10); /* 6553 */
   i = i + 3;
   printf("%d\n", i); /* 2: 65536+2 */
   return 0;
}
```

Primjer 3.

```
#include <stdio.h>
int main(void) {
   unsigned short i = 2, j = 4;
   i = i - j;
   printf("%d\n", i); /* 65534: 2+65536-4 */
return 0;
}
```

Kod pridruživanja vrijednosti i kod čitanja također vrijedi modularna aritmetika.

Zadaća. Koju vrijednost će poprimiti varijabla `i` nakon izvršavanja sljedećih naredbi:

a)
```
unsigned short i = 65546;
```

b)
```
unsigned short i = 1024*1024;
```

c)
```
unsigned short i;
scanf("%d",&i); /* upišemo 65546 */
```

Prikaz cijelih brojeva s predznakom

Cijeli brojevi s predznakom modeliraju skup $\mathbb{Z}$. Među prikazivim brojevima moraju biti i neki negativni brojevi. Dogovor je da se u računalu prikazuje najveći mogući podskup uzastopnih brojeva iz $\mathbb{Z}$ koji je "skoro" simetričan oko 0. Dakle, prikazivih negativnih brojeva ima podjednako mnogo kao i nenegativnih. Prisjetimo se: ako imamo n bitova na raspolaganju za prikaz, onda skup svih prikazivih brojeva ima najviše 2^n elemenata. Ako želimo da točno polovina tih brojeva bude negativna, onda njih mora biti $2^n/2 = 2^{n-1}$. Dakle, skup svih prikazivih brojeva cijelih brojeva s predznakom jednak je $\mathbb{Z}_{2^n}^- = \{-2^{n-1}, -2^{n-1}+1, \ldots, -2, -1, 0, 1, 2, \ldots, 2^{n-1}-2, 2^{n-1}-1\}$. Brojeve izvan tog skupa ne možemo prikazati (kao cijele brojeve s predznakom) koristeći samo n bitova. Najmanji i najveći prikazivi cijeli broj s predznakom su, redom: -2^{n-1}, $2^{n-1}-1$. Tipične vrijednosti za te brojeve su

n	-2^{n-1}	$2^{n-1}-1$
8	-128	127
16	$-32\ 768$	32 767
32	$-2\ 147\ 483\ 648$	2 147 483 647
64	$-9\ 223\ 372\ 036\ 854\ 775\ 806$	9 223 372 036 854 775 805

U programskom jeziku C cijelim brojevima s predznakom odgovara tip koji se zove `int`. Ovaj tip, osim u standardnoj veličini, postoji i u veličinama `short`, `long` i `long long`, ovisno o broju bitova predviđenih za prikaz. U zaglavnoj datoteci `limits.h` definirane su cjelobrojne konstante za osnovne cjelobrojne tipove. Npr. `SHRT_MIN` je najmanji prikazivi cijeli broj za tip `short int`, a `INT_MAX` najveći prikazivi cijeli broj za tip `int`.

Kako stvarno izgleda prikaz brojeva u tih n bitova? Nenegativni brojevi imaju isti prikaz kao i cijeli brojevi bez predznaka. Stoga aritmetika za cijele brojeve s predznakom mora i dalje biti ista, tj. modularna aritmetika modulo 2^n. No, što je s prikazom negativnih brojeva? Pravilo za prikaz je vrlo jednostavno: iste prikaze imaju oni brojevi koji imaju isti "pravi" ostatak modulo 2^n. Drugim riječima, za $B = 1, 2, \ldots, 2^{n-1}$

- prikaz negativnog broja $-B$ (s predznakom)
- jednak je prikazu broja $2^n - B$ (bez predznaka).

Primjer 4. Neka je $n = 3$, tj. imamo samo 3 bita za prikaz. Prikazivih brojeva ima $2^3 = 8$. Prikazi cijelih brojeva bez predznaka iz $\mathbb{Z}_8$ i cijelih brojeva s predznakom iz $\mathbb{Z}_8^-$ su:

$B \in \mathbb{Z}_8$	$B \in \mathbb{Z}_8^-$	prikaz
0	0	000
1	1	001
2	2	010
3	3	011
4	-4	100
5	-3	101
6	-2	110
7	-1	111

Primjer 5.

- $2^{n-1} - 1 \longleftrightarrow [0\,1\,1\,1\,1\,1\,1\,1]$
- $-1 \longleftrightarrow 2^n - 1 \longleftrightarrow [1\,1\,1\,1\,1\,1\,1\,1]$
- $-2^{n-1} \longleftrightarrow 2^n - 2^{n-1} = 2^{n-1} \longleftrightarrow [1\,0\,0\,0\,0\,0\,0\,0]$.

Primjer 6.

```
#include <stdio.h>
int main(void) {
   short int i = 32766; /* n = 16 */
   i += 1;
   printf("%d\n", i); /* 32767 */
   i += 1;
   printf("%d\n", i); /* -32768, a ne 32768 */
return 0;
}
```

Sljedeći puta ćemo reći nešto o prikazu suprotnog broja preko komplementa te o dijeljenju cijelih brojeva s predznakom. Za kraj, pogledajmo jedan primjer klasične pogreške.

Primjer 7. Računanje $n!$ u cjelobrojnoj aritmetici.

Za prirodni broj n funkciju faktorijela definiramo na sljedeći način:

$$1! = 1,\ n! = n \cdot (n-1)! \quad \text{za} \quad n \geq 2.$$

Napišimo sada program koji u cjelobrojnoj aritmetici računa $n!$.

```
#include <stdio.h>
int main(void) {
   int i, f50 = 1; /* n = 32 */
   for (i = 2; i <= 50; ++i)
      f50 *= i;
   printf("f50 = %d\n", f50); /* f50 = 0 */
return 0;
}
```

Zašto je izlazna vrijednost `f50=0`? Točna vrijednost je $50! = 30414093201713378043\,61260\,81660\,64768\,84437\,76415\,68960\,51200\,00000\,00000$.

$50! = 2^{47} \cdot 3^{22} \cdot 5^{12} \cdot 7^8 \cdot 11^4 \cdot 13^3 \cdot 17^2 \cdot 19^2 \cdot 23^2 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47$, odnosno $50! = 0 \mod 2^{32}$.