

Programiranje (C)

1. predavanje

Saša Singer

`singer@math.hr`

`web.math.hr/~singer`

`www.math.hr/~singer`

PMF – Matematički odjel, Zagreb

Na samom početku

- Moja malenkost (u punom “sjaju”):
doc. dr. sc. Saša Singer
- Službeni osobni podaci:
 - ured (soba, kabinet): 227, drugi kat,
 - e-mail: singer@math.hr
(Molim plain text poruke.)
 - web stranica: <http://web.math.hr/~singer>
odn. <http://www.math.hr/~singer>
- Konzultacije (zasad): petak, 12–14 sati.

Sadržaj predavanja

- Cilj kolegija Programiranje (C).
- Kratki pregled sadržaja kolegija.
- Pravila igre ili kako položiti C.
- Literatura i korisni linkovi.

Nakon toga ide “pravo” uvodno predavanje.

- Kratki uvod u programski jezik C:
 - Pisanje i prevodenje programa,
 - Nekoliko primjera programa.

Cilj kolegija Programiranje (C)

Iz naslova je **očito** da kolegij ima **2 osnovna cilja**:

- **osnovne tehnike programiranja**, tj. realizacija **osnovnih algoritama** (nastavak na **UuR**),
- **konkretni programski jezik — C**, koji je sredstvo za **realizaciju** tih algoritama,

ili, **drugim riječima**,

- **pisanje** **osnovnih algoritama** u **programskom jeziku C**.

To je ono što **Vi** morate **naučiti**, **napraviti** i/ili **savladati**, da biste položili **C**.

Pregled sadržaja kolegija (1)

Što sve moramo napraviti? Za početak:

- Uvod u programske jezike i, posebno, C.
- Osnove programiranja u Unix okruženju (praktikumi).
- Isto to, ali u Windows okruženju (praktikumi).
- Kratki uvod u programiranje u C-u, primjeri jednostavnih programa (koristimo UuR).

To ide relativno brzo, a treba za početak vježbi.

Ostatak sadržaja ([na predavanjima](#)), uglavnom, redom obrađuje

- razne elemente programskog jezika C.

Pregled sadržaja kolegija (2)

Teme — elementi programskog jezika C:

- Osnovni tipovi podataka. Konstante, variable, enumeracije.
- Operatori, izrazi, prioriteti operatora, naredba dodjeljivanja (pridruživanja) vrijednosti.
- Ulaz/izlaz podataka, funkcije `scanf`, `printf`, `gets`, `puts`, `getchar` i `putchar`. Funkcije za rad sa znakovima.
- Kontrola toka programa (složene naredbe):
 - petlje `while`, `for` i `do -- while`,
 - naredbe `if`, `switch`, `break`, `continue` i `goto`.

Pregled sadržaja kolegija (3)

- Funkcije, rekurzivne funkcije, funkcije s varijabilnim brojem argumenata.
- Pre(d)procesorske naredbe, uvjetno uključivanje, parametrizirane makro naredbe, `assert` makro.
- Struktura programa, memorijske klase, doseg variable. Povezivanje programa smještenog u više datoteka.
- Polja, polja znakova, višedimenzionalna polja, funkcije za rad sa stringovima, `sscanf`, `sprintf`.
- Pokazivači (pointeri), aritmetika pokazivača, veza pokazivača i polja, pokazivači i funkcije, dinamička alokacija (rezervacija) memorije.

Pregled sadržaja kolegija (4)

- Argumenti komandne linije, složene deklaracije.
- Strukture, strukture i pokazivači, samoreferentne strukture (vezane liste i složenije strukture), apstraktni tipovi podataka, `typedef`, unije.
- Datoteke, vrste datoteka, otvaranje i zatvaranje datoteka, standardne funkcije za čitanje i pisanje, binarno čitanje i pisanje.
- Operacije nad bitovima, polja bitova.
- Pregled sadržaja standardne C biblioteke.

Dakle, sasvim lijepa (stvarno — **ogromna**) količina posla.
I to nije sve!

Pregled sadržaja kolegija (5)

Osnovne tehnike programiranja (na predavanjima) radimo

- na primjerima upotrebe tih elemenata jezika.

I, naravno, na **vježbama!**

Trebat će vam znanje osnovnih algoritama iz UuR-a.

- Ponovite to na vrijeme!

Kako položiti C?

Ispit (ocjena) se sastoji iz **3** dijela (slično kao na UuR):

- **10%** — domaće zadaće. To su programi koje predajete (i “branite”) za računalom!
- **40%** — 1. kolokvij,
- **50%** — 2. kolokvij.

Svaki od ta **3** dijela **morate zadovoljiti** (tj. položiti ga).

Usmenog **NEMA**, osim po želji, u okviru završnog ispita.
Usput, to se radi na računalu!

Više detalja o “**praktičnom**” dijelu nastave — na vježbama.
Iskoristite praktikume!

Kako položiti C — Upozorenje

“Nema šale” — programiranje se uči prvenstveno

- samostalnim pisanjem programa **na računalu**.

Nema zamjene za to iskustvo! Ne može ga netko steći za vas, umjesto vas.

Upozorenje: C **nije jednostavan** jezik i **nije** izmišljen za učenje programiranja.

Napomene:

- otvoriti **račun** za računala na odjelu,
- koristiti DevC++ (ili **cc**, **gcc**), nabaviti CD za C,
- probati programe s predavanja i vježbi (bit će dostupni na CD-u, webu).

Literatura za C (1)

Osnovna literatura su, naravno,

- predavanja i vježbe,

s popratnim materijalima (na pr. programi na webu).

Dobrom voljom prof. Juraka, na (mom) webu dostupna je njegova skripta:

- Mladen Jurak, Programski jezik C, ak. god. 2003/04 (verzija 1), PMF–MO, 2004.

Iskoristite to!

Prof. Jurak i ja znamo da toj skripti fale

- ozbiljniji primjeri i potpuni programi.

Literatura za C (2)

Dakle, skripta **ne pokriva** jedan dio **tehnika programiranja**.
Dorada skripte u tom smjeru ovisi o raspoloživom vremenu.

Zato iskoristite dostupne **programe s vježbi**. Ne samo za
čitanje!

- Probajte ih prevesti i **izvršiti**,
- **testirati** za razne ulazne podatke,
- **mijenjati** i gledati što se događa.

I, **najvažnije**,

- sami napišite, utipkajte, prevedite i izvršite ponešto programa.

To je jedino “pravo” iskustvo.

Dodatna literatura za C (1)

Razno–raznih knjiga o C-u i programiranju ima zaista mnogo.

Svakako **najpoznatija** je knjiga **autora** jezika C:

- Brian W. Kernighan i Dennis M. Ritchie,
The C Programming Language (second edition),
Prentice Hall, Upper Saddle River, New Jersey, 1988.

Drugo izdanje pokriva tzv. **ANSI C** standard iz 1990. godine.

Knjiga je malo “**sažeta**” na nekim mjestima, pa se katkad **teže** čita, ali

- sadrži niz **izrazito korisnih primjera** (dijelova programa i cijelih programa) za **osnovne tehnike programiranja**.

Dodatna literatura za C (2)

Uz KR2 knjigu, mogu vam dobro poslužiti i dodatne lekcije u HTML formatu na web stranici:

<http://www.eskimo.com/~scs/cclass>

Možete ih dobiti od mene (ZIP), ali ih **ne smijem** staviti na web. (“Redistribute” smije, “republish” ne.)

Spomenut ću još dvije knjige iz literature u skripti:

- B. S. Gottfried, Theory and Problems of Programming with C (second edition), Schaum's outline series, McGraw-Hill, New York, 1996.
(Uputa: tražite najnovije izdanje.)
- S. Prata, C Primer Plus (4th edition), SAMS, 2002.

Dodatna literatura za C (3)

Nadalje, svaka relativno nova knjiga iz C-a može sasvim dobro poslužiti za učenje.

- Ima ih i na hrvatskom, pa birajte.

Savjet: otvoriti knjigu i početi čitati na nekom mjestu. Ako razumijete, probajte na još par mjesta. Ako i dalje nema problema, možete razmisliti o kupnji.

Usput, svaka iole “pristojna” knjiga iz C-a mora pokriti i osnovne tehnike programiranja. Inače je stvarno smeće.

I, na kraju, ne zaboravite da na webu “ima svega”, pa tako i gomila kurseva iz programiranja u C-u.

Korisni linkovi (1)

Službena web stranica kolegija je:

<http://web.math.hr/nastava/c>

Ako ne radi (tj. javlja čudnu poruku o grešci, tipa “... SQL error ...”), odmah javite meni ili asistentima.

Svratite katkad na stranice elektronskog matematičkog časopisa **math.e**

<http://web.math.hr/math.e>

gdje ima vrlo zgodnih članaka i za buduće računarce.

Korisni linkovi (2)

Također, pogledajte studentske članke na studentskom webu

<http://student.math.hr/>

Tamo ima nekoliko zanimljivih članaka iz računarstva.

Na “vrhu” je članak:

- [Anamari Nakić, Najjače računalo na svijetu.](#)

Na dnu tog članka je link na članak:

- [Jakiša Tomić, Kako odabratи pravi C/C++ kompjajler?](#)

To, očito, ima veze s kolegijem i možda vam može pomoći!

Korisni linkovi (3)

Dodatak, bez veze sa C-om, ali vrlo koristan, posebno za studente 1. i 2. godine. Na webu postoji i baza definicija matematičkih pojmove (na hrvatskom i engleskom):

<http://www.baza.iugrina.com>

Autori su naši studenti:

- Anamari Nakić (sad već dipl. inž.) i
- Ivo Ugrina (4. godina).

Ako ste zainteresirani da im pomognete u “punjenju” i oblikovanju baze, javite mi se.

Uostalom, pogledajte math.e, broj 6.

Uvod u programske jezike

Gruba podjela programskih jezika:

- Strojni jezik — izvršni program, instrukcije u binarnom kôdu.
- Asembler — izvorni program (tekst kojeg treba prevesti), jezik prilagođen arhitekturi računala (tekst je direktna zamjena za binarne instrukcije i lako se prevodi u njih).
- Viši programski jezici — izvorni program, program je tekst sastavljen od naredbi, jezik je prilagođen posebnim zadaćama za koje je jezik namijenjen.
Najpoznatiji jezici:
 - C, FORTRAN, Pascal, C++, Java, Python, ...

Primjer strojnih instrukcija i Assemblera (8086)

Primjer s prastarog Intelovog procesora **8086** za IBM XT.
(To je 16 bitni procesor, instrukcije se pišu byte po byte).

Strojni jezik Ekvivalentan asemblerski kôd

00011110	PUSH	DS
00101011	SUB	AX, AX
11000000		
10111000	PUSH	AX
10111000	MOV	AX, MYDATA
00000001		
10001110		
11011000	MOV	DS,AX

Mane strojnog jezika i asemblera

Osnovne mane takvih programa:

- Instrukcije su strogo određene arhitekturom računala.
Posljedica: programi se ne mogu prenijeti na računalo drugačijeg tipa.
- Pisanje programa je izrazito neefikasno.
 - Instrukcije se vrlo elementarne, prilagođene stroju, a ne poslu kojeg treba napraviti.
 - Rezultat su dugački, nepregledni programi.
 - Podložno greškama, koje se teško otkrivaju.
- Asembler je samo malo humaniji od strojnog jezika (zamjena binarnog kôda tzv. mnemoničkim kôdom — tekstualnim imenima instrukcija).

Svrha programskih jezika

Stvarna svrha programiranja u asembleru je samo za:

- specifične zadaće vezane uz upravljanje hardwareom.

Programer ima punu kontrolu nad svim komponentama računala.

Za sve ostale, ljudima “bliže” poslove koriste se tzv. viši programski jezici.

- “Viši” = bliži čovjeku, a ne stroju.

Viši programski jezici

Postoji ih gomila. Najpoznatiji jezici:

- C, FORTRAN, Pascal, C++, Java, Perl, Python, ...

Osnovne prednosti nad asemblerom:

- Neovisnost o arhitekturi računala (prenosivost).
- Prilagođenost pojedinim specifičnim zadaćama (naredbe prilagođene tipovima podataka i operacijama nad njima — u odgovarajućoj primjeni).
- Složenije naredbe, bliže ljudskom načinu mišljenja.

Rezultat: Veća efikasnost programiranja (brže i jednostavnije obavljanje posla).

Viši programski jezici (nastavak)

Program u takvom jeziku prvo treba **prevesti**

- iz **izvornog** kôda (engl. **source code**)
- u **izvršni** kôd (engl. **executable code**).

Ovaj postupak, obično, ide u više koraka (v. malo kasnije).

- Posao prevodenja radi program — **prevoditelj** (engl. **compiler**) za dani jezik.
- **Prenosivost** — program se može (barem u principu) izvršiti na bilo kojem računalu koje ima prevoditelj za određeni jezik.
- Prevodenje je bitno **složenije** nego kod asemblera.
- Zato jezici imaju svoja **stroga pravila** (gramatike).

Programski jezik C

C je viši programski jezik opće namjene.

- Autor: Dennis Ritchie (Bell Telephone Laboratories).
- Razvijen sedamdestih godina prošlog stoljeća.
- Osnovna svrha:
 - Pisanje jezgre operacijskog sustava Unix.
- Ideja:
 - maksimalna portabilnost Unixa na razne vrste računala.
- Zato je jezgra napisana u posebno smišljenom višem jeziku, a ne u strojnom, za neko posebno računalo.

Programski jezik C — osnovna svojstva (1)

Početna svrha — razvoj sistemskih programa, uvelike određuje “izgled” jezika.

- C je jezik relativno “niskog nivoa”, ne jako daleko od arhitekture računala (“high level Assembler”).
- To znači da C operira s istim objektima kao i većina računala:
 - znakovima, brojevima, adresama (pointerima ili pokazivačima).
- C podržava sve operacije na tim podacima koje su podržane arhitekturom računala:
 - aritmetičke, logičke, relacijske (usporedbe), ali i
 - posebne operacije na bitovima, poput pomaka (“shift”).

Programski jezik C — osnovna svojstva (2)

S druge strane, kao **viši jezik**, C ima

- grupiranje naredbi, složene naredbe za kontrolu toka (petlje, uvjetne naredbe),
- izvedene ili složene tipove podataka, poput polja, struktura, datoteka,
- mogućnost razbijanja programa u manje cjeline koje mogu biti smještene u raznim datotekama,
- manje programske cjeline (potprograme) — u C-u su to funkcije,
 - koje mogu vratiti vrijednosti svih osnovnih tipova i nekih složenih tipova (strukture),
 - i mogu se rekursivno pozivati.

Programski jezik C — osnovna svojstva (3)

Na kraju, C ima **standardiziranu** programsku biblioteku koja sadrži **sve strojno ovisne** elemente jezika. Sastoji se iz **dva** bitna dijela.

- Niz funkcija za interakciju s okolinom (operacijskim sustavom), poput
 - čitanja i pisanja datoteka,
 - formatirani ulaz i izlaz,
 - alokaciju memorije,
 - operacije sa znakovima i stringovima, itd.
- Skup standardnih zaglavljia (tzv. “headers”) koji uniformiraju pristup
 - deklaraciji funkcija i tipova podataka.

Programski jezik C — opis i standardi (1)

Opis jezika dan je u knjizi (KR)

- Brian W. Kernighan i Dennis M. Ritchie,
The C Programming Language,
Prentice Hall, New Jersey, 1978.

I jezik C i operacijski sustav Unix brzo se šire sedamdesetih i osamdesetih godina prošlog stoljeća.

- American National Standard Institute (ANSI) pristupa standardizaciji C-a, koja je dovršena 1989. godine.

Novi standard uveo je značajne izmjene u jeziku.

- Osnovna pravila jezika (gramatika) su znatno stroža.

Posljedica: Lakše prevođenje i otkrivanje grešaka.

Programski jezik C — opis i standardi (2)

Za razliku od **prvotne** verzije, **novi** standard često se naziva **ANSI-C**. Opisan je u knjizi (**KR2**)

- Brian W. Kernighan i Dennis M. Ritchie,
The C Programming Language (second edition),
Prentice Hall, Upper Saddle River, New Jersey, 1988.

Implementiran je u svim modernim C–prevodiocima.

- ANSI standard usvojila je i **Međunarodna organizacija za standarde** (ISO) 1990. godine (tzv. **ISO C**).

Ovaj **ANSI/ISO** standard skraćeno zovemo **C90** standard.

- Godine 1999. ISO je prihvatio **novi C standard**, koji uvodi **manje dopune** u C90 standard.

Skraćeni naziv: **C99** standard.

Opći postupak pisanja programa

Postupak “programiranja” u programskom jeziku C pod raznim operacijskim sustavima vrlo je sličan.

Bitni koraci:

- Prvo se **tekst programa**, tj. **izvorni kôd**, upiše u neku tekstualnu datoteku. Standardne ekstenzije su **.c** ili **.h**, za zaglavlja. Na primjer, u datoteku **prvi.c**.
- Zatim se poziva program prevodtelj (C compiler) koji transformira (prevodi) napisani program u **izvršni kôd**. Kao rezultat dobiva se **izvršna datoteka** (standardne ekstenzije su **.out** na Unixu, ili **.exe** na Windowsima).
- Pozivom te datoteke **izvršava** se program.

Opći postupak pisanja programa (nastavak)

Napomena: U pisanju programa redovito se javljaju greške, pa treba dodati:

- Nalaženje i ispravljanje grešaka, što, obično, rezultira ponavljanjem prethodnih koraka, sve dok program ne “proradi”!

Ovo iznad (više–manje) vrijedi

- i za Unix, i za Linux, i za MS Windows.

Samo se imena programa koje pozivamo (koristimo) u pojedinim fazama razlikuju.

U nastavku, malo više o tome kako se to radi na Unixu. Kako to ide na Windowsima — više o tome na vježbama.

Postupak pisanja programa u Unix okruženju

- Napisati izvorni kôd. On se sastoji od jedne ili više
 - programskih datoteka (**.c**),
 - datoteka zaglavlja (**.h**).
- Pozvati C prevoditelj (compiler) koji prevodi svaku **.c** datoteku u tzv. objektni kôd kojeg spremi u pripadnu **.o** datoteku.
- Linker povezuje sve potrebne **.o** datoteke i programske biblioteke u izvršni kôd.
- Ako mu se **ne naredi** suprotno,
 - prevoditelj **sam** poziva linker,
 - linker generira izvršnu datoteku **a.out**

Primjeri rada u Unix okruženju

Poziv prevoditelja i zadavanje imena pojedinih datoteka.

- Tekst programa je u `prvi.c`:

```
cc prvi.c
```

```
./a.out
```

- Zadavanje imena izvršne datoteke.

```
cc prvi.c -o prvi
```

```
prvi.out
```

- Istovremeno prevođenje više datoteka.

```
cc prvi.c drugi.c treci.c -o svi
```

```
./svi.out
```

Alati za programiranje u Unix okruženju (1)

- Editor teksta

- standardni `vi`,
 - ili neki drugi, na pr. `pico`.

Svrha: kreiranje tekst datoteka, poput `ime.c`, `ime.h`.

- Prevoditelj (compiler)

- `cc`, `gcc`.

Napomena: prevoditelj ima brojne mogućnosti koje zadajemo tzv. **opcijama**.

- Povezivač ili punjač (engl. `linker` ili `loader`)

- `ld`.

Povezivanje objektnih datoteka s programskim bibliotekama u izvršni kôd.

Alati za programiranje u Unix okruženju (2)

- Priručnik (engl. manual) za razne stvari (i opcije)

- `man`.

Primjeri:

- `man cc` za opcije prevoditelja,
 - `man vi` za standardni editor,
 - `man scanf` za standardne funkcije iz C biblioteke.

- Debugger — alat za traženje grešaka u programu (kad se program “**slomi**” pri izvršavanju)

- `dbx, gdb`.

- Alat za verifikaciju programa i provjeru sintakse

- `lint`.

Alati za programiranje u Unix okruženju (3)

- Make — alat razvoj većih programa “razasutih” u puno datoteka
 - `make`, `gmake`.
Automatizira prevodenje i povezivanje nakon promjena pojedinih datoteka.
- Programske biblioteke
 - uključivanje se vrši `-l` opcijom.