

Увод у Програмски језик C

Александар Пајкановић

nanoluka.org
YouTube.com/nanoluka
instagram.com/nanolukaorg
twitter.com/nanolukaorg
github.com/nanoluka
nanolukaorg@gmail.com

Октобар 2020



Кориштење овог документа

- Лиценца:

- ▶ Документ објављујем под лиценцом *Attribution-ShareAlike — CC BY-SA*. Ова лиценца дозвољава ремикс и прераду, као и комерцијално кориштење дјела, ако/док се **правилно назначавате име аутора** и ако се **прерада лиценцира под истим условима**. Ова лиценца се често упоређује са „*copyleft*” лиценцирањем слободног софтвера или софтвера отвореног кода. Сва дјела настала на основу дјела лиценцираног овом лиценцом, требало би да буду лиценцирана истом лиценцом, која, поред осталог, дозвољава комерцијално кориштење.

- Навођење (цитирање):

- ▶ Ако сте користили овај документ као извор у сопственим материјалима, молим да цитирате на сљедећи начин: А. Пајкановић, „Увод у програмски језик С кроз практичне примјере”, доступно на github.com/nanoluka/jezik-c.git

Програмирање и програмски језици

- Рачунар у општем смислу је машина способна да изврши било који алгоритам пратећи дефинисан скуп правила - тзв. Тјурингова машина (теоријска замисао).
- Алгоритам је недвосмислен метод рјешавања конкретног проблема
- Рачунар се већ читав вијек састоји од процесора и меморије - тзв. Фон Нојманова архитектура (реализација)
- Програмирање, у практичном смислу, можемо рећи да представља састављање текстуалног (понегдје и графичког) описа жељеног понашања рачунара у смислу обраде корисничких података и представљања резултата тог процеса кориснику
- Програмски језик је скуп инструкција (кључних ријечи) које, ако су посложене у складу са унапријед дефинисаним правилима, рачунар може да прихвати - при чему ово „прихвати” значи да је у стању да, узев дате податке корисника на улазу, кориснику прикаже очекивани излаз

Програмски језик C

- Постоји много подјела програмских језика, најважнија је на:
 - ▶ језике вишег нивоа, и
 - ▶ језике нижег нивоа
- C спада у језик вишег нивоа. Генерално, сви програмски језици за које сте чули су, у ствари, у истој групи. Ова друга група своди се на машински, тј. асемблер - о том потом, неки други пут.
- Друга важна подјела је према парадигми којој програмски језик припада, тренутно су актуелне четири парадигме (које се, често, међусобно пресјецају):
 - ▶ императивно програмирање,
 - ▶ функционално,
 - ▶ логичко, и
 - ▶ објектно оријентисано.
- Наш данашњи језик спада у прву групу, а његов млађи брат `C++` у посљедњу.

Програмски језик C

- Иначе, C је насљедник програмског језика B (гле чуда), настао је почетком 70их у Бел Лабораторијама, човјек се звао Денис Ричи.
- Најважније карактеристике су веома ефикасан превод у асемблерске инструкције, писан је за извршавање на различитим платформама, брзина извршавања, велика слобода, па је зато веома распрострањен - данас највише у домену програмирања уграђених система.
- C друге стране, за разлику од пајтона, на примјер, који је можда најраспрострањенији на планети, крива учења је ипак нешто стрмија.
- Укратко, поступак од идеје до реализације састоји се од пет корака.

Програмски језик C

- Укратко, поступак од идеје до реализације састоји се од пет корака:
 - ❶ алгоритам,
 - ❷ псеудо кôд,
 - ❸ изворни кôд - овдје се пише синтакса (.c),
 - ❹ компајлирањем добијамо објектни кôд (.obj), и
 - ❺ линкер нам даје реализацију, тзв. извршни кôд, односно извршну датотеку (.exe)
- На том путу, сусрећемо се са разним невољама, а подијелићемо их у грешксе:
 - ▶ синтаксичке, и
 - ▶ семантичке.
- Ова презентација не представља потпуне информације о рачунарима, програмирању и програмском језику C, него служи као увод или преглед, како бисмо се упознали са основама и знали гдје и шта да тражимо како бисмо стварно научили.

Потребни алати

- Рачунар - бар током овог предавања, али може и телефон или таблет - само је унос проблематичан
- *Windows* оперативни систем - може, наравно, и неки други али данас рад демонстрирамо овако
- *Dev C++* - интегрисано развојно окружење (енгл. *IDE*), потпун алат и лак за инсталацију и кориштење, бесплатно доступан на <https://sourceforge.net/projects/orwelldvcpp/>
- *Arduino* - интегрисано развојно окружење за рад са Ардуино платформом, пакет са свим потенцијалним додацима доступан на: <https://www.arduino.cc/>
- Познавање енглеског језика
- Добра воља, чврста одлука да се не одустаје и упоран рад

Примјер 1 - Испис текста

HelloICBL

```
#include <stdio.h>
int main() {

    // printf() displays the string inside quotation
    printf("Hello , ICBL!");
    return 0;
}
```


Типови података

- Основна подјела сигнала:
 - ▶ аналогни - звук, бежични пренос података, зрачење, слика
 - ▶ дигитални - запис аналогних, али користећи само ограничен број нивоа - ако су само два нивоа, онда су то бинарни сигнали
- Данашњи комерцијално доступни рачунари су дигитални и разумију искључиво бинарне податке
- Математички запис је дат прије два вијека, данас познат као бинарна или Булова алгебра
- Једна бинарна цифра назива се бит (енгл. *binary digit* → *bit*)
- Осам бита је бајт (енгл. *byte*), kilo, mega, итд.
- Подаци су осмобитни, 32-битни, итд.
- Корисни записи су још и октални и хексадецимални
- Децимални број 6, осмобитно се представља као 00000110, број 126 пишемо 01111110, а -126 је 10000010
- Означенио и неозначени

Примјер 2 - Типови података

DataTypes

```
#include <stdio.h>
int main() {
    int a;
    long b;
    char c;
    float d;

    printf("size of int = %d bytes\n", sizeof(a));
    printf("size of short = %d bytes\n", sizeof(b));
    printf("size of char = %d bytes\n", sizeof(c));
    printf("size of float = %d bytes\n", sizeof(d));
    return 0;
}
```

Примјер 3 - Аритметичке и логичке операције

ALU

```
int a = 2;
int b = 4;
char c = 'a';
float d = 6.2;

printf("zbir: a+b = %d\n", a+b);
printf("razlika: a-b = %d\n", a-b);
printf("proizvod: a*c = %d\n", a*c);
printf("kolicnik - cjelobrojno: a/b= %f\n", a/b);
printf("kolicnik - decimalno: a/d= %f\n", a/b);
printf("jednakost: a == b = %d\n", a==b);
printf("binarno l: a & b = %d\n", a & b);
printf("logicko l: a && b = %d\n", a && b);
```

Контрола тѳка

- Рачунари не одлучују. Бар не још увијек
- Рачунари поступају по инструкцијама
- Инструкција може да се изврши или не изврши у зависности од услова, односно може да се изврши једна или друга, опет, наравно, зависно од стања нечег другог:
 - ▶ ако је температура већа од 22° C, искључи гријање
 - ▶ ако је температура мања од 18° C, укључи гријање
 - ▶ не извршавај ништа, док се не притисне овај тастер
 - ▶ изврши сабирање свих бројева у овом низу, тј. 10 000 сабирања
- Графички приказ тѳка извршавања назива се дијаграм тѳка
- Тѳк контролишемо користећи се гранањем (if-else, switch) и петљама (for, while, do-while)

Примјер 4 - Унос података

DataInput

```
#include <stdio.h>
int main()
{
    int a;
    printf("Unesite broj: ");
    scanf("%d", &a);

    printf("Broj je = %d", a);
    return 0;
}
```

Примјер 5 - Гранање

Branch

```
int a;  
printf("Unesite broj: ");  
scanf("%d", &a);  
  
if (a > 9) {  
    printf("Broj %d je dvocifren", a);}  
else if (a > 100) {  
    printf("Broj %d je trocifren", a);}  
else {  
    printf("Broj %d je jednocifren", a);}
```

Примјер 6 - Петља

Loop

```
int a;  
int f = 1;  
printf("Unesite broj: ");  
scanf("%d", &a);  
  
int i;  
for(i = 0; i < a; i++){  
    printf("Ispisujem %d. put\n", i);  
    f = f * (i+1);  
}  
printf("Faktorijel broja %d je: %d\n", a, f);
```

Енкапсулација и апстракција

- Тако далеко сам видио, зато што сам стојао сам на плећима дивова
- Све што постоји, а да је створено људском руком, настало је комбинацијом већ постојећег
- Зашто да измишљамо топлу воду?
- Дајте ми полугу, помјерићу свијет
- Право питање је: шта је проблем?
- Видјели смо који су типови података, видјели смо које су доступне аритметичке операције.
- Како ћемо израчунати коријен? Експонент? Синус? Интеграл?

Примјер 7 - Кориштење библиотека

Library / Lib

```
#include <math.h>
#include <stdio.h>

int main() {
    float a = 3, e = 4, r = 0;

    // calculates the power
    r = pow(a, e);

    printf("%.1lf^%.1lf = %.2lf", a, e, r);
    return 0;
}
```

Примјер 8 - Писање функција

Function

```
#include <stdio.h>
float stepen(float x, float y) {
    int i;
    float r = 1;
    for(i = 1; i <= y; i++) {
        r = r * x; }
    return r; }
int main() {
    float a = 3, e = 4, r = 0;
    r = stepen(a, e);
    printf("%.1lf^%.1lf = %.2lf", a, e, r);
    return 0; }
```

Уграђени (енгл. *embedded*) системи

- Уграђени системи су рачунарски системи намијењени за извршавање специфичне функције у реалном времену, и могу да буду саставни дио обимнијих електромеханичких (над)система.
- Рачунари су, на примјер, системи опште намјене, дакле нису „уграђени” у овом смислу.
- Махом су дигитални, а централни дио је увијек микроконтролер.
- Микроконтролер је процесор са периферијама и (малом) меморијом.
- Internet of Things - интернет свега, паметни системи
- Аутомобилска индустрија (енгл. *automotive*)
- Интересантан свијет, на граници између хардвера и софтвера, рачунарска електроника, рачунарски инжењеринг, мехатроника, роботика.
- Ардуино је сјајан први корак на том путу.

Примјер 9 - Трепћући Ардуино

Blink

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);    //izlazni  
}  
void loop() /*PETLJA*/ {  
    digitalWrite(LED_BUILTIN, HIGH); //VISOK nivo  
    delay(1000);                      //sacekaj  
    digitalWrite(LED_BUILTIN, LOW);  //NIZAK nivo  
    delay(1000);                      //sackej  
}
```

Примјер 10 - Зглоб једног робота

Motor

```
digitalWrite(smjer, HIGH); // Izbor smjera rotacije
for(int x = 0; x < 200; x++) {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500); }
delay(1000); // Sacekaj jednu sekundu
digitalWrite(smjer, LOW); // Promjena smjera rotacije
for(int x = 0; x < 400; x++) {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500); }
```

Референце

- Google!!!
- Увод у програмирање кроз *C*, Школа кода
- Programiz
- Coursera и edX
- Утичнионица- основно о Ардуину, са Електротехничког факултета у Бањој Луци
- Упутство за рад са *CNC* наставком за Ардуино, видео и текст
- 1000 за будућност- увод у рад са:
 - ▶ Пи рачунар
 - ▶ програмски језик *Python*
- Званична презентација *Raspberry Pi*
- Увод у електронику - серија Три минута електронике
- Књига о *LTspice*, софтверском симулатору електричних кола