

# Razlika između programskih jezika C i C++

Seminarski rad u okviru kursa  
Tehničko i naučno pisanje  
Matematički fakultet

Ime i prezime autora  
kontakt email adresa autora

24. oktobar 2017.

## Sažetak

Programski jezici C i C++ imaju veliki broj sličnosti, od zajedničke sintakse do sličnih standardnih biblioteka. Međutim, u jezik C++ su ugrađene brojne funkcionalnosti i proširenja standardne biblioteke koja značajno olakšavaju pisanje programa u ovom jeziku u odnosu na jezik C. U ovom radu navodimo neke od nedostataka jezika C koji su bili motivacija za stvaranje jezika C++, razlike između ova dva jezika, kao i gde se oni danas koriste.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>O jeziku C</b>	<b>2</b>
2.1	Nedostaci jezika C . . . . .	2
<b>3</b>	<b>Razlike između jezika C i C++</b>	<b>2</b>
<b>4</b>	<b>Gde se C danas koristi</b>	<b>3</b>
4.1	Programiranje ugrađenih sistema . . . . .	4
4.2	Sistemsko programiranje . . . . .	4
<b>5</b>	<b>Zaključak</b>	<b>4</b>
	<b>Literatura</b>	<b>4</b>
<b>A</b>	<b>Dodatak</b>	<b>4</b>

# 1 Uvod

Uvod

## 2 O jeziku C

Programski jezik C je programski jezik opšte namene koji je 1972. godine razvio Denis Riči<sup>1</sup>. C je jezik koji je bio namenjen prevashodno pisanju sistemskog softvera i to u okviru operativnog sistema Unix. C je danas prisutan na širokom spektru platformi – od mikrokontrolera do superračunara.

Jezik C spada u grupu imperativnih, proceduralnih programskih jezika. Kako je izvorno bio namenjen za sistemsko programiranje, programerima nudi prilično direktan pristup memoriji i konstrukcije jezika su tako osmišljene da se jednostavno prevodi na mašinski jezik. Jezik je kreiran u minimalističkom duhu – ima mali broj ključnih reči, a dodatna funkcionalnost programerima se nudi uglavnom kroz korišćenje bibliotečkih funkcija.

### 2.1 Nedostaci jezika C

Uprkos svojoj brzini i efikasnosti programski jezik C ima izvestan broj nedostataka, koji je možda najbolje opisan citatom Denisa Ričija, čoveka koji je napravio C: „C ima moć asemblerkih jezika i koristi se lako kao asemblerki jezici.” Asemblerki jezici su ozloglašeni po težini korišćenja i Denis Riči je ovime hteo da ukaže da je C takođe komplikovan i da ne pruža puno olakšica programerima koji ga koriste.

Neki od poznatijih zamerki na C su: način na koji se dinamički alokira memorija koristeći *malloc* i *calloc* koji može dovesti do curenja memorije ukoliko se ona ne oslobađa pravilno ili do pristupa delu memorije koji je već oslobođen što bi takođe dovelo do greske. Način na koji funkcionišu pokazivači takođe zna da bude problematičan i njihovo netačno korišćenje od strane programera može dovesti do korupcije memorije. Moguće je i da više pokazivača sadrže istu adresu i time program ne bude maksimalno optimizovan, kao što je slučaj u nekim drugim programskim jezicima. Kompajleri nisu od velike pomoći programeru i moguće je napraviti greške koje se mogu detektovati u fazi kompilacije, ali za to kompilator nije sposoban. Niske se čuvaju kao niz podataka tipa *char* sa terminirajućom nulom na kraju i njihovo korišćenje i obrada su veoma komplikovani i zahtevaju manipulaciju memorijom od strane programera. Ne postoji ništa što sprečava programera da piše neodrživ kod i istorijski se to dešavalo toliko često da postoje i takmičenja za pisanje što komplikovanijeg koda u programskom jeziku C. Objektno orijentisano programiranje takođe nije podržano u C-u kao i mnoge druge funkcionalnosti koje olakšavaju pisanje i održavanje koda (npr. introspekcija tipa).

## 3 Razlike između jezika C i C++

Najznačajniji direktni naslednik jezika C je jezik C++ koji se, u trenutku nastanka, mogao smatrati njegovim objektno-orijentisanim proširenjem.

---

<sup>1</sup>Dennis Ritchie (1941–2011), američki informatičar

Kreirao ga je Bjern Stroustrup<sup>2</sup> 1986. godine. C++ je i dalje jedan od najpopularnijih jezika i koristi se za razvoj zahtevnih aplikacija, s jedne strane zbog svojih objektno-orijentisanih svojstava, a s druge zbog bliske veze sa mašinom, u duhu jezika C.

Neka od svojstava jezika C++ koja ne postoje u jeziku C su

- mnoštvo objektnih tipova kao što su *vector*, *set*, *stack*, *queue*, implementiranih u standardnoj biblioteci
- mogućnost prenosa argumenata funkcije i po referenci
- range for petlje

Osobina	C	C++
Programska paradigma	Imperativna	Objektno-orijentisana
Osnova	Asembler	Jezik C
Petprocesorske direktive	?	?
Pristup u programiranju	S vrha - nanže	S dna - naviše
Alokacija memorije	Korišćenjem bibliotečkih funkcija	Korišćenjem ključnih reči
Prenos argumenata funkcije	Po vrednosti	Po vrednosti i po referenci
Pokazivači	Podržani	Podržani
Upravljanje izuzecima	Nije podržano	Podržano

## 4 Gde se C danas koristi

Iako je programski jezik C bio namenjen pre svega pisanju sistemskog softvera, usled njegove velike popularnosti koristio se i kao jezik opšte namene. Vremenom su mnoge uloge koje je programski jezik C imao od svog nastanka 1972. godine preuzeli drugi, moderniji programski jezici. Međutim i danas 50 godina nakon svog nastanka programski jezik C ima veliku popularnost. Neke od oblasti primena programskog jezika C danas su:

- operativni sistemi;
- programiranje ugrađenih sistema;
- izrada kompilatora i interpretatora;
- u obrazovanju.

C programski jezik imam najveću primenu u pisanju programa „bliskih hardveru“. To nije slučajnost budući da je jezik od početka tako dizajniran. Prema rečima Brajana Kernigena<sup>3</sup> i Denis Ričija „C je jezik dosta *niskog nivoa*. Ta karakterizacija nije pežorativna; to samo znači da se C nosi sa objektima na sličan način kao i većina računara“. Razlog što je C uspešan u oblastima kao što su sistemsko programiranje i programiranje ugrađenih sistema je u tome što nudi mnoge prednosti jezika višeg nivoa uz jako malo žrtvovanja u odnosu na programiranje na assembleru. Neke od prednosti u odnosu na assembler su lakše i udobnije pisanje čitkog i kompaktnog koda na jeziku koji poznatom velikom broju programera kao i visok nivo prenosivosti koda. C kompilatori su dostupni za većinu

<sup>2</sup>Bjarne Stroustrup (1950), danski informatičar

<sup>3</sup>Brian W. Kernighan (1942), kanadski informatičar

današnjih procesora, što oslobađa programera od brige o implementaciji konkretne procesorske arhitekture. Sa druge strane u odnosu na većinu viših programskih jezika, C nudi visok nivo kontrole nad hardverom i veoma visoku efikasnost u izvršavanju programa.

## 4.1 Programiranje ugrađenih sistema

Ugrađeni sistem je kombinacija računarskog hardvera i softvera kao i drugih mehaničkih i električnih delova koji ima neku određenu funkciju. Za razliku od ličnih računara, koji imaju mnoštvo različitih funkcija i primena, ugrađeni sistemi obično imaju jednu konkretnu svrhu često kao deo nekog većeg sistema. Primeri su mnogobrojni uređaji u automobilima (npr. čipovi koji kontrolišu rad ABS sistema), jednostavni digitalni časovnici, razni uređaji u domaćinstvu i drugi. Budući da su mogućnosti takvih sistema često jako ograničene u smislu memorije i procesorske snage, pre svega radi uštede na ceni i dimenzijama uređaja, efikasnost programa je jako bitna. Upravo zbog toga je C pogodan za tu namenu više nego i jedan drugi jezik. Osim C-a se često se koriste i assembler i C++, međutim prednosti koje C++ donosi često nisu vredne gubitaka na efikasnosti osim kod jako velikih razvojnih timova.

## 4.2 Sistemsko programiranje

Sistemsko programiranje, odnosno programiranje operativnih sistema takođe zahteva jezik blizak hardveru. C programski jezik je upravo dizajniran za potrebe sistemskog programiranja i to u okviru operativnog sistema Unix. Među modernim operativnim sistemima C se najviše vezuje za Linux operativni sistem, ali većina popularnih operativnih sistema je barem značajnim delom pisana u jeziku C. O značaju C programskog jezika u sistemskom programiranju govori podatak da je u Arch Linux distribuciji Linux operativnog sistema (poznatoj po minimalnoj instalaciji), preko 97% programskog koda jezgra napisano u jeziku C kao i najveći deo od 27 osnovnih paketa koji omogućavaju osnovnu instalaciju sistema.

# 5 Zaključak

Zaključak

## Literatura

- [1] J. Laski and W. Stanley. *Software Verification and Analysis*. Springer-Verlag, London, 2009.
- [2] Free Software Foundation. GNU gcc, 2013. on-line at: <http://gcc.gnu.org/>.
- [3] A. M. Turing. *On Computable Numbers, with an application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society, 2(42):230–265, 1936.

## A Dodatak

Ovde pišem dodatne stvari, ukoliko za time ima potrebe. Ovde pišem dodatne stvari, ukoliko za time ima potrebe. Ovde pišem dodatne stvari,

ukoliko za time ima potrebe. Ovde pišem dodatne stvari, ukoliko za time ima potrebe. Ovde pišem dodatne stvari, ukoliko za time ima potrebe.