

# Debugovanje u LLDB-u

Seminarski rad u okviru kursa  
Metodologija stručnog i naučnog rada  
Matematički fakultet

Momir Adžemovic, Miloš Miković, Marko Spasić,  
Mladen Dobrašinović

momir.adzemovic@gmail.com, spaskeasm@gmail.com,  
milos.mikovicpos@gmail.com, dobrasinovic.mladen@gmail.com

1. april 2020.

## Sažetak

U ovom tekstu je ukratko prikazana osnovna forma seminarskog rada. Obratite pažnju da je pored ove .pdf datoteke, u prilogu i odgovarajuća .tex datoteka, kao i .bib datoteka korišćena za generisanje literature. Na prvoj strani seminarskog rada su naslov, apstrakt i sadržaj, i to sve mora da stane na prvu stranu! Kako bi Vaš seminarski zadovoljio standarde i očekivanja, koristite uputstva i materijale sa predavanja na temu pisanja seminarskih radova. Ovo je samo šablon koji se odnosi na fizički izgled seminarskog rada (šablon koji *morate* da koristite!) kao i par tehničkih pomoćnih uputstava. Pročitajte tekst pažljivo jer on sadrži i važne informacije vezane za zahteve obima i karakteristika seminarskog rada.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>3</b>
<b>2</b>	<b>Osnovna uputstva</b>	<b>3</b>
<b>3</b>	<b>Upoznavanje sa LLDB-om</b>	<b>3</b>
3.1	LLDB interfejs komandne linije . . . . .	4
<b>4</b>	<b>Engleski termini i citiranje</b>	<b>4</b>
<b>5</b>	<b>Slike i tabele</b>	<b>5</b>
<b>6</b>	<b>Kôd i paket listings</b>	<b>6</b>
<b>7</b>	<b>Prvi naslov</b>	<b>6</b>
7.1	Prvi podnaslov . . . . .	6
7.2	Drugi podnaslov . . . . .	6
7.3	... podnaslov . . . . .	6
<b>8</b>	<b>Poređenje sa drugim popularnim debagerima</b>	<b>6</b>
8.1	Poređenje: GDB i LLDB . . . . .	7
8.2	Visual Studio Debugger i LLDB . . . . .	7

<b>9 Zaključak</b>	<b>7</b>
<b>Literatura</b>	<b>7</b>
<b>A Dodatak</b>	<b>8</b>

## 1 Uvod

Kada budete predavali seminarski rad, imenujete datoteke tako da sadrže redni broj teme, temu seminarskog rada, kao i prezimena članova grupe. Precizna uputstva na temu imenovnja će biti data na formi za predaju seminarskog rada. Predaja seminarskih radova biće isključivo preko veb forme, a NE slanjem mejla. Link na formu će biti dat u okviru obaveštenja na strani kursa. Vodite računa da prilikom predavanja seminarskog rada predate samo one fajlove koji su neophodni za ponovno generisanje pdf datoteke. To znači da pomoćne fajlove, kao što su .log, .out, .blg, .toc, .aux i slično, **ne treba predavati**.

## 2 Osnovna uputstva

Vaš seminarski rad mora da sadrži najmanje jednu **sliku**, najmanje jednu **tabelu** i najmanje **sedam referenci** u spisku literature. Najmanje jedna slika treba da bude originalna i da predstavlja neke podatke koje ste Vi osmislili da treba da prezentujete u svom radu. Isto važi i za najmanje jednu tabelu. Od referenci, neophodno je imati bar jednu **knjigu**, bar jedan **naučni članak** iz odgovarajućeg časopisa i bar jednu adekvatnu **veb adresu**.

**Dužina seminarskog rada treba da bude od 10 do 12 strana.** Svako prekoračenje ili potkoračenje biće kažnjeno sa odgovarajućim brojem poena. Eventualno, nakon strane 12, može se javiti samo tekst poglavlja **Dodatak** koji sadrži nekakav dodatni kôd, ali je svakako potrebno da rad može da se pročita i razume i bez čitanja tog dodatka.

Ко жели, може да пише рад ћирилицом. У том случају, неопходно је да су инсталирани одговарајући пакети: texlive-fonts-extra, texlive-latex-extra, texlive-lang-cyrillic, texlive-lang-other.

Nemojte koristiti stari način pisanja slova, tj ovo:

```
\v{s} i \v{c} i \’c ...
```

Koristite direktno naša slova:

```
š i č i ć ...
```

## 3 Upoznavanje sa LLDB-om

LLDB podržava standardne funkcije debugovanja preko komandne linije i može se koristiti kao debager u interaktivnom razvojnom okruženju. Konkretno, sa debagerom pokrenutim nad programom prevedenim sa debug opcijama omogućava se [4]:

- Aktiviranje procesa programa sa određenim argumentima komandne linije (eng. command line arguments).
- Korišćenje breakpoint-a (određenog reda ili funkcije u izvornom kodu pri kojima debager zaustavlja izvršavanje programa kada se stigne do odgovarajućeg dela izvršnog koda).
- Korišćenje watchpoint-a (određene promenljive, takve da debager zaustavlja proces ili nit kada se njeno stanje promeni).
- Korišćenje dodatnih uslova nad breakpoint-ovima i watchpoint-ovima.
  - Nastavljanje ili pokretanje programa.
- Pokretanje procesa red po red (sa „ulaženjem” u funkciju ili bez).

- Istraživanje promenljivih ili memorije procesa.
- Izvršavanje proizvoljnog izraza nad stanjem procesa (npr. menjanje neke promenljive na steku).
- Istraživanje steka okvira poziva.
- Izvršavanje drugih naprednih i raznih funkcija.

Ono što ističe LLDB je omogućavanje korišćenja eksternih skripti za debagovanje preko javnog API-a za Python, izvršavanje proizvoljnog Python koda unutar debagera [9] (preko ugnježdenog interpretatora (eng. embedded interpreter) i omogućavanje REPL (Read-Evaluate-Print-Loop) funkcija za programske jezike zajedno sa mogućnostima debagovanja [2].

### 3.1 LLDB interfejs komandne linije

LLDB interfejs komandne linije (eng. command line interface) se aktivira pozivom `lldb` u ljustici (eng. shell) sa programom koji želimo debagovati kao argumentom. Program komandne linije `lldb` se odlikuje strukturisanom sintaksom osnovnih komandi koja je sledećeg oblika [10]:

#### Primer 3.1

```
<imenica> <glagol> [-opcije [vrednost-opcije]] [argument [argument...]]
```

U ovakvom obliku, imenica se zove i komanda, a glagol potkomanda. Postoje i skraćenice (eng. alias) za komande koje mogu odstupati od ovog oblika. Upravo zato što je ovaj format komandi jako strukturisan mogu nam biti pogodni skraćeni oblici komandi koji mogu biti sličniji onome što je poznato korisnicima drugih debagera [3]. U tabeli 1 su date neke od osnovnih komandi kao primer korišćenja interfejsa i reprezentativni prikaz širokog skupa mogućnosti LLDB-a koji nije naveden u potpunosti u ovom radu. Posebno se ističu komande `help` i `apropos`, koje mogu biti korisne početnicima u korišćenju ovog alata.

## 4 Engleski termini i citiranje

Na svakom mestu u tekstu naglasiti odakle tačno potiču informacije. Uz sve novouvedene termine u zagradi naglasiti od koje engleske reči termin potiče.

Naredni primeri ilustruju način uvođenja engleskih termina kao i citiranje.

**Primer 4.1** *Problem zaustavljanja (eng. halting problem) je neodlučiv.*

**Primer 4.2** *Za prevođenje programa napisanih u programskom jeziku C može se koristiti GCC kompajler.*

**Primer 4.3** *Da bi se ispitivala ispravnost softvera, najpre je potrebno precizno definisati njegovo ponašanje.*

Reference koje se koriste u ovom tekstu zadate su u datoteci *seminarski.bib*. Prevođenje u pdf format u Linux okruženju može se uraditi na sledeći način:

```
pdflatex TemaImePrezime.tex
bibtex TemaImePrezime.aux
pdflatex TemaImePrezime.tex
pdflatex TemaImePrezime.tex
```

Tabela 1: Upotreba interfejsa komandne linije LLDB-a [4][10].

<code>process launch -- &lt;argumenti&gt;</code>	Pokreće izabrani program sa datim argumentima.
<code>thread step-in</code>	U trenutnoj niti nastavlja izvršavanje programa sledeće instrukcije izvornog koda, ulazeći u pozive funkcija.
<code>thread step-inst-over</code>	U trenutnoj niti nastavlja izvršavanje programa sledeće instrukcije izvršnog koda, ne ulazeći u pozive funkcija.
<code>breakpoint set --file 1.c --line 42</code>	Postavlja breakpoint na red 42 u izvornom kodu programa 1.c.
<code>breakpoint list</code>	Ispisuje postojeće breakpoint-ove debagera.
<code>breakpoint disable 1</code>	Deaktivira breakpoint 1.
<code>apropos &lt;ključna_reč&gt;</code>	Traži u pomoći za upotrebu komandi (eng. command help) datu ključnu reč.
<code>help</code>	Štampa pomoć za komande. (help se može koristiti i za nalaženje pomoći za upotrebu potkomandi određene komande [3]).

Prvo latexovanje je neophodno da bi se generisao *.aux* fajl. *bibtex* proizvodi odgovarajući *.bbl* fajl koji se koristi za generisanje literature. Potrebna su dva prolaza (dva puta pdflatex) da bi se reference ubacile u tekst (tj da ne bi ostali znakovi pitanja umesto referenci). Dodavanjem novih referenci potrebno je ponoviti ceo postupak.

Broj naslova i podnaslova je proizvoljan. Neophodni su samo Uvod i Zaključak. Na poglavlja unutar teksta referisati se po potrebi.

**Primer 4.4** U odeljku 7 precizirani su osnovni pojmovi, dok su zaključci dati u odeljku 9.

Još jednom da napomenem da nema razloga da pišete:

`\v{s} i \v{c} i \v{c} ...`

Možete koristiti srpska slova

`š i č i ć ...`

## 5 Slike i tabele

Slike i tabele treba da budu u svom okruženju, sa odgovarajućim naslovima, obeležene labelom da koje omogućava referenciranje.

**Primer 5.1** Ovako se ubacuje slika. Obratiti pažnju da je dodato i `\usepackage{graphicx}`

Na svaku sliku neophodno je referisati se negde u tekstu. Na primer, na slici ?? prikazane su pande.

**Primer 5.2** I tabele treba da budu u svom okruženju, i na njih je neophodno referisati se u tekstu. Na primer, u tabeli 2 su prikazana različita poravnanja u tabelama.

Tabela 2: Razlčita poravnanja u okviru iste tabele ne treba koristiti jer su nepregledna.

centralno poravnanje	levo poravnanje	desno poravnanje
a	b	c
d	e	f

## 6 Kôd i paket listings

Za ubacivanje koda koristite paket **listings**: [https://en.wikibooks.org/wiki/LaTeX/Source\\_Code\\_Listings](https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings)

**Primer 6.1** *Primer ubacivanja koda za programski jezik Python dat je kroz listing 1. Za neki drugi programski jezik, treba podesiti odgovarajući programski jezik u okviru defnisanja stila.*

```

1000 # This program adds up integers in the command line
import sys
1002 try:
    total = sum(int(arg) for arg in sys.argv[1:])
1004     print 'sum =', total
except ValueError:
1006     print 'Please supply integer arguments'

```

Listing 1: Primer ubacivanja koda u tekst

## 7 Prvi naslov

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

### 7.1 Prvi podnaslov

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

### 7.2 Drugi podnaslov

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

### 7.3 ... podnaslov

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

## 8 Poredjenje sa drugim popularnim debagerima

Potrebno je naglasiti da pri poređenju različitih debagera ne možemo objektivno odrediti koji je debager najbolji, jer to dosta zavisi koji se operativni sistem koristi, a i samih preferenci korisnika. Visual Studio

Code, jedan od popularnijih editora, koristi LLDB, GDB i VSD za C++ u zavisnosti od od operativnog sistema na kojem je instaliran [8]:

- **Linux:** GDB
- **macOS:** LLDB or GDB
- **Windows:** the Visual Studio Windows Debugger or GDB (using Cygwin or MinGW)

	LLDB	GDB	Visual Studio Debugger
Podrška za programske jezike	C, C++, Objective C	C, C++, Objective C, Java, Fortran etc.	C#, C++, Visual Basic, JavaScript etc.
Implementacija	C++	C	C++/C#
Podrška za operativne sisteme	Unix, Windows, MacOS	Unix, Windows, MacOS	Windows
Razvijen od strane	GNU Project	LLVM developer group	Microsoft
UI	TUI	TUI	GUI

Slika 1: LLDB, GDB, Visual Studio Debugger [1][5][7]

## 8.1 Poređenje: GDB i LLDB

Debugger GDB predstavlja standard za GNU sisteme (ne striktno samo za GNU) [1]. Ako se proverava kvalitet debagera LLDB, onda u potpunosti ima smisla upoređivati ga prvo sa GDB debagerom kao jednim od najpopularnijih debagera. Debager LLDB u debugovanju velikih programa pokazuje bolje performanse od GDB debagera i ima dobar korisnički interfejs [6]. Način korišćenje ova dva debagera je veoma sličan i skup komandi se većinom poklapa. Postoji zvaničan rečnik koji prevodi komande iz GDB u LLDB [4]. Novije verzije GDB podržavaju MacOS, ali u proteklih par godina se pretežno koristio LLDB kao glavni debager za MacOS.

## 8.2 Visual Studio Debugger i LLDB

Visual studio debugger je takođe jedan od poznatijih debagera koji možemo da upoređujemo sa LLDB-om. Prednost VSD u odnosu na LLDB je u tome što VSD nudi grafički point-and-click korisnički interfejs, a prednost LLDB je u broju operativnih sistema koji za koju ima podršku [7].

# 9 Zaključak

Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak.

## Literatura

- [1] Free Software Foundation. GDB: The GNU Project Debugger, 2020. on-line at: <https://www.gnu.org/software/gdb/>.
- [2] Apple Inc. REPL and Debugger, 2020. on-line at: <https://swift.org/lldb/#why-combine-the-repl-and-debugger>.
- [3] Apple Inc. Understanding LLDB Command Syntax, 2020. on-line at: <https://developer.apple.com/library/archive/documentation/General/Conceptual/lldb-guide/chapters/C2-Understanding-LLDB-Command-Syntax.html>.
- [4] LLVM. GDB to LLDB command map, 2020. on-line at: <https://lldb.llvm.org/use/map.html>.
- [5] LLVM. The LLDB Debugger, 2020. on-line at: <https://lldb.llvm.org>.
- [6] LLVM. The LLDB Debugger, 2020. on-line at: <http://blog.llvm.org/2010/06/new-lldb-debugger.html>.
- [7] Microsoft. Visual Studio Debugger Suport, 2019. <https://docs.microsoft.com/en-us/visualstudio/debugger/debugger-feature-tour?view=vs-2019>.
- [8] Microsoft. Debug C++ in Visual Studio Code, 2020. <https://code.visualstudio.com/docs/cpp/cpp-debug>.
- [9] The LLDB Team. Python Scripting, 2020. on-line at: <https://lldb.llvm.org/use/python.html>.
- [10] The LLDB Team. Tutorial, 2020. on-line at: <https://lldb.llvm.org/use/tutorial.html>.

## 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.