

GN GDB debager

Kristina Pantelić, 91/2016, kristinapantelic@gmail.com

Ivana Cvetkoski, 65/2016, ivana.cvetkoski@gmail.com

Bojana Ristanović, 45/2016, bojanaristanovic97@gmail.com

Nikola Stamenić, 177/2016, nikola.stamenic@hotmail.com

18. april 2020

Sadržaj

Glava 1

Recenzent — ocena:

1.1 O čemu rad govori?

Rad prvenstveno govori o GDB debageru, načinu njegove upotrebe u konzolnim i grafičkim korisničkim okruženjima, poređenju sa drugim popularnim debagerima, osnovnim operacijama koje pruža korisnicima i metodama upravljanja samim debagerom. Na početku rada, dat je i kratak pregled o osnovnim pojmovima debugovanja i upoznavanje sa debagerom kao alatom za pronalaženje grešaka.

1.2 Krupne primedbe i sugestije

U poglavlju “Uvod” mislim da je dobar deo teksta bolje prebaciti kao uvod u poglavlje “Debugovanje”, ovaj tekst zameniti tekstom koji će malo više zainteresovati čitaoca za temu rada a ne za temu debugovanja. **Napisan je potpuno novi uvod.** U poglavlju “Debugovanje” umesto detaljnog objašnjavanja klasičnog načina debugovanja, u tom delu mogu se dodati i drugi načini debugovanja i ukratko se uporediti. Takodje, u poglavlju “Debugovanje” za rečenicu “Odlike koje poseduju uspešni debageri su kreativnost, logičko zaključivanje...” na prvi pogled stiže se utisak da se opisuje alat a ne čovek koji vrši debugovanje. **Rečenica je preformulisana i spojena sa prethodnom i sada glasi:** “Debugovanje je jedan od najkreativnijih aspekata programiranja, zahteva od programera iskustvo, inteligenciju, razmišljanje “van kutije”, sagledavanje problema sa različitih strana, logičko zaključivanje, odlučnost; ali može biti i jedan od najzahtevnijih aspekata programiranja.” Dalje, u poglavlju “Debugovanje” u pasusu koji počinje rečenicom “Suprotno tome, sa grafičkim alatima za uklanjanje bagova...” stiže se utisak da svi debageri koriste grafički korisnički interfejs i da svi imaju opisanu funkcionalnost, jer suprotno klasičnom načinu debugovanja je korišćenje debagera a ne debagera sa grafičkim korisničkim interfejsom. **Rečenica je izbačena jer je dodat deo gde su opisani različiti načini debugovanja.** U poglavlju “Debager”, rečenicu “Kada program usled бага ili netačnog podatka ne može da nastavi normalno sa radom, debager pokazuje lokaciju problema u originalnom kodu” treba preformulisati, jer nema svaki debager ovu funkcionalnost, zavisi od interfejsa debagera i podrške za konkretan debager. **Umesto „debager“ dodatno je „napredniji debager“.**

Mislim da je poglavlje “Udaljeno debugovanje” previše detaljno opisano i treba ga skratiti i napisati samo najbitnije odlike ovog načina debugovanja. [Deo o udaljenom debugovanju je skraćen i jasnije napisan](#). U poglavlju “Grafičko okruženje gdbgui” mislim da je preopširno opisan korisnički interfejs, umesto toga, možda bi bilo bolje opisati zašto je ovo okruženje dobro, po čemu se izdvaja od drugih i kada ga treba koristiti ili u kojim situacijama se najčeće koristi. Slično, za poglavlje “GDB u Qt Creator razvojnom okruženju”, mislim da je preopširno opisan način upotrebe GDB debagera u ovom okruženju, možda bi bilo bolje odgovoriti na pitanja napisana u prethodnoj rečenici, nepotrebno su opisane informacije koje sadrži tačka prekida jer postoji slika na kojoj se one vide, takodje slike “4” i “5” nalaze se u poglavlju “6”, mislim da je bolje prebaciti ih iznad, jer nemaju veze sa poglavljem u kojem se nalaze. [Poglavlja o gdbgui-u i Qt Creator-u su izmenjena tako da ne govore o samom korišćenju i objašnjavanju interfejsa, već koje mogućnosti poseduje dati alat](#). [Slike 4 i 5 su zajedno sa upotrebom alata Qt Creator prebačene u dodatak, ali su se pre toga nalazile u poglavlju 6 iz razloga što sam L^AT_EX raspoređuje figure tako da celokupni dokument zauzme što manje prostora](#).

Tekst ispod poglavlja “Poređenje sa drugim popularnim debagerima” govori o stvarima na koje treba obratiti pažnju prilikom izbora debagera i nema veze sa naslovom poglavlja, možda je bolje ispod poglavlja “Debager” napraviti podpoglavljje sa ovim tekstom. Kod poređenja GDB debagera sa LLDB i Valgrind debagerom, možda ne bi bilo loše dodati tabelu na kojoj će biti prikazano poređenje GDB debagera sa ostalim debagerima na primer po pitanju podrške za programske jezike, implementacije samih debagera, korisničkih interfejsa i slično, time se dobija na preglednosti.

Zaključak, govori o bagovima, debugovanju i zašto je debager koristan alat, a jako malo o GDB debageru, treba ga preformulisati tako da sumira sve pozitivne i negativne stvari vezane za GDB debager, dati čitaocu pregled situacija (na primer jezika, operativnih sistema, razvojnih okruženja) kada GDB pruža svoj maksimalni potencijal, ali i dati prostora čitaocu da sam odluči da li je GDB debager za njega.

1.3 Sitne primedbe

Nisam naišao na veliki broj štamparskih grešaka, ali posoji nekolicina koje sam zapazio. U poglavlju 3 u rečenici “takođe zavisi, u određenoj meri, i od programskog jezika...” ne treba pisati slovo “i” posle zareza. [Ispravljeno je u “ali takođe, u određenoj meri, zavisi i od programskog jezika koji se koristi”](#) U poglavlju 3 u prvoj rečenici treba napraviti razmak “meta program”. [Ispravno je i "metaprogram", "meta-program", kao i "meta program", stoga ostavljeno je "metaprogram"](#). U poglavlju 4.1.2 možda je lepše u zagradi napisati “počevši od vrha steka” umesto pisanja crtice, takodje u istom poglavlju treba napraviti razmak kod “šetamo kroz stek pozive”. [Crtica je zamenjena zarezom, jer smatramo da tako lepše izgleda nego u zagradama, ali je konstatacija na mestu](#). [Razmak nije pdf-u zbog latex-a, jer on tako ostavlja, ali je i to rešeno](#). U poglavlju 5.1 jedna od rečenica glasi “Tako se tu mogu naći..”, lepše je napisati “Tu se mogu naći..”, na više mesta u ovom poglavlju nije napravljen razmak na mestima gde je potrebno kao na primer “Load Binary(slika 1.1)” ili “konzole(slika 1.5)”. [Razmaci su rešeni, s obzirom na naknadno dodavanje tih delova, nije uočeno da](#)

nedostaju razmaci.

Videćemo šta ćemo za deo pre toga, za tu rečenicu. Globalno je možda lepše, kada referišemo na sliku, tu referencu staviti u zagradu, na primer (1) umesto 1. Stilovi pisanja su prilično dobro usklađeni. Ovo treba da se dogovorimo

1.4 Provera sadržajnosti i forme seminarskog rada

1. Da li rad dobro odgovara na zadatu temu?
Rad je u velikoj meri odgovorio na ključna pitanja vezana za ovu temu koja se nalaze u opisu teme. Potrebno je samo, po mom mišljenju, neke delove skratiti, a neke dopuniti kao što sam opisao u odeljku “Krupne primedbe i sugestije”.
2. Da li je nešto važno propušteno?
Mislim da nije propuštena nijedna ključna komponenta, ali se seminarski može poboljšati u svakom slučaju.
3. Da li ima suštinskih grešaka i propusta?
Smatram da se za zaključak može reći da je prilično pogrešan i treba ga ispraviti, čitajući samo njega čitalac stiče utisak da rad govori o debagovanju i debagerima, a tema rada je GDB debager, slično važi i za uvod. [Napisan je novi zaključak kao i uvod](#)
4. Da li je naslov rada dobro izabran?
Naslov rada je dobro izabran i sadržaj je u velikoj meri usklađen sa njim. Izdvojio bih samo poglavlje “6” i tekst ispod njega, koji nema veze sa naslovom. [I ovo](#)
5. Da li sažetak sadrži prave podatke o radu?
Sažetak daje prave podatke o radu i mislim da je ukratko opisao sve teme kojima se rad bavi.
6. Da li je rad lak-težak za čitanje?
Rad je pretežno lak za čitanje, osim u delovima gde se opisuje rad GDB debagera u QT Creator razvojnom okruženju, gdbgui i udaljeno debagovanje gde rad postaje suvoparan.
7. Da li je za razumevanje teksta potrebno predznanje i u kolikoj meri?
Za razumevanje teksta nije potrebno preveliko predznanje, jer je dosta pojmova detaljno objašnjeno u tekstu, ipak tekst se dosta bolje razume ako čitalac ima predznanja na temu debagovanja i rada sa debagerom.
8. Da li je u radu navedena odgovarajuća literatura?
Literatura jeste dobro navedena sem reference 4 <https://www.hpc-europa.org/>. za koju nisam pronašao literaturu vezanu za rad. Ostatak literature sadrži dosta knjiga vezanih za temu debagera što je pohvalno. Određeni delovi korišćeni su iz literature o debagovanju profesora Saše Malkova, bilo bi lepo, navesti i ovaj izvor literature. [Bojana treba da reši hpc-europa, ako se ne varam ona je dodala to, a ja ne mogu da se snađem :D Baltazara \(https://konkursiregiona.net/wp-content/uploads/2016/05/prof.-baltazar-253x300.jpg\) treba da citiramo.](#)

9. Da li su u radu reference korektno navedene?
Sa reference pod rednim brojem 4, <https://www.hpc-europa.org/>. nisam pronašao podatke koji stoje u tekstu, ostale reference jesu dobro navedene.
10. Da li je struktura rada adekvatna?
Struktura rada jeste dobra, rad sadrži poglavlja i podpoglavlja, ima sažetak, uvod i zaključak, sadržaj i deo sa literaturom. Pohvalio bih to što se tabela najbitnih komandi nalazi u dodatku, a ne u tekstu, rad je dosta pregledniji na ovaj način.
11. Da li rad sadrži sve elemente propisane uslovom seminarskog rada (slike, tabele, broj strana...)?
Rad sadrži slike kao i tabele, broj strana je zadovoljen kao i broj referenci. U skup referenci ukljčeno je tri master rada kao i knjige na temu debugovanja.
12. Da li su slike i tabele funkcionalne i adekvatne?
Tabele i slike su u velikoj meri funkcionalne, samo nije bilo potrebno detaljno objašnjavati u tekstu sadržaje sa nekih slika. Sliku 1 bilo bi lepo povećati. [Slika 1 je uvećana.](#)

1.5 Ocenite sebe

U oblast debugovanja i debagera, mogu da kažem, da nisam preterano dobro upućen, rekao bih da sam srednje upućen. GDB debager, jesam koristio, ali vrlo malo. Za razumevanje rada, mislim da nije potrebno preveliko znanje iz ovih oblasti, stoga smatram da sam radi razumeo u potpunosti. Moje zamerke najviše su se odnosile na čitljivost i preglednost rada, kao i strukturanost pojedinih celina, neke tehničke stvari koje sam uvideo kao greške već sam obrazložio, za ostale podatke napisane u tekstu, nisam dovoljno stručan da potvrdim ispravnost.

Glava 2

Recenzent — ocena:

2.1 O čemu rad govori?

Ovaj rad prelazi opšte koncepte бага i debugovanja, nakon čega te iste koncepte prikazuje na GNU GDB debageru. Obrađuje njegovu osnovnu funkcionalnost, prednosti i mane, kao i podržane varijante grafičkih okruženja za debugovanje posredstvom GDB debagera. Nakon čega poredi njegovu funkcionalnost, portabilnost i prilagodljivost sa drugim debagerima i profajlerima.

2.2 Krupne primedbe i sugestije

1. U tekstu se navodi da Mac OS X nije uniksoliki operativni sistem, a u stvari jeste. Može donekle biti zbunjujući naziv jezgra "XNU - X Is not Unix" koji nije dodeljen zato što je operativni sistem neuniksolik. [Greška je ispravljena. Obrisan je Mac OS X iz pomenute rečenice, jer on spada u uniksolike operativne sisteme.](#)
2. Pri obradi QT Creator-a i njegovog okruženja za debugovanje je napomenuto da pri zaustavljanju programa nema nikakve indikacije gde se desila fatalna greška, što je netačno, jer je u svakom trenutku dostupan ceo stek sa pozivima funkcija sa trenutnim brojevima linija, kao i trenutno stanje promenljivih, ovo će biti dostupno čak iako se ne postavi tačka zaustavljanja na mesto greške. [Rešen je problem, određeni prozor u kom se to ispisuje je bio isključen, samim tim nije uočen ispis čak iako je bio tražen s obzirom da u literaturi piše da ima ispisa, ali se kod autora nije pojavljivao, pa neke neproverene stvari autor nije hteo da stavlja u rad.](#)
3. U C++ kodu iz listinga 1 ima sintaksička greška, kose crte za komentare su pogrešnom smeru. [Rešena primedba, pogrešna strana je ostala zbog latex-a i oznake za novi red, pa je autor povučen time napravio grešku.](#)
4. Pri kraju se u sekciji "Poređenje sa drugim popularnim debagerima", GDB poredi sa VALGRIND-om, što u kontekstu podnaslova nema mnogo smisla, s obzirom da VALGRIND nije debager već skup alata za, pre svega, profajliranje. Čak i da se ograničimo na Memcheck, i on je isključivo

memorijski debager, što po definiciji znači da služi za dosta različit skup problema od regularnih debagera, pa mi se čini da nije dobro porediti ih. Moje mišljenje je da treba ili obraditi drugi debager ili se pozabaviti pitanjem: "Kada treba koristiti debager, a kada profajler?"

5. Navedeno je da je DDD debager, što stvara konfuziju u rečenici gde se istovremeno uvodi kao debager i kao softver koji podržava debagere, po meni je najbolje adresirati ga kao "grafički interfejs za debagere".
Naveden deo je preformulisan na sledeći način: "...zbog popularnosti grafičkih korisničkih interfejsa (*eng.* GUI) razvijen je veliki broj GUI zasnovanih (*eng.* GUI-based) debagera koji rade pod Unix sistemom. Većina njih su grafički interfejsi za debagere. Jedan od najpoznatijih grafičkih interfejsa za debagere je DDD (*eng.* Data Display Debugger)".
6. Referenca broj 4 u literaturi je dosta nekonkretna ili je pogrešno postavljena. Nisam uspeo na obezbeđenom linku da nađem navedene činjenice odnosno literaturu. [Kao i gore, hpc-europa moramo da rešimo](#)

2.3 Sitne primedbe

1. Diskutabilno je korišćenje rečenice "jedno od najintelektualnijih aspekata programiranja" u 3. sekciji. U redu je naglasiti da zahteva razmišljanje "van kutije" i da je zbog toga potrebna određena doza kreativnosti, iskustva, i inteligencije, ali među velikim bogatstvom računarskih nauka, po mom mišljenju, ima mnogo fascinantnijih oblasti. Da se ne bi dolazilo u ovakve subjektivne konflikte, autori bi možda trebalo da izbegavaju subjektivne tvrdnje.
Rečenica je preformulisana: "...zahteva od programera iskustvo, inteligenciju, razmišljanje "van kutije", sagledavanje problema sa različitih strana; ali može biti i jedan od najzahtevnijih aspekata programiranja".
2. "Veština debagovanja programera može biti bitan faktor u debagovanju" mnogo zvuči kao tautologija.
Rečenica je preformulisana: "Iskustvo i veština debagovanja programera su bitni faktori u procesu debagovanja..."
3. Štamparska greška na 11. strani, Veliki deo komadi → Veliki deo komandi
[Rešeno](#)
4. Štamparska greška u sekciji 4.1.4, u prvom pasusu, specifičnost → specifičnost
[Rešeno](#)
5. Nisam nigde našao u literaturi navedenu podelu na početku 6. poglavlja. Ako jeste originalni deo teksta ovog rada, preporučujem da se napomene da je ovo na osnovu prethodnih zapažanja autora.
Neke od stvari na koje treba obratiti pažnju prilikom izbora debagera → Na osnovu prethodnog teksta, neke od stvari na koje treba obratiti pažnju prilikom izbora debagera
Ili pak ako nije originalno, da se doda izvor. [Ovo treba da rešimo, ja iskreno ne znam je li neko sam pisao ili citirao, ali valjalo bi rešiti.](#)
[Ni Kristina ne zna odakle je ovo, nije ona pisala.](#)

6. Ni u jednom trenutku se ne pokazuje izgled GDB u terminalu. Bilo bi mnogo jasnije kako funkcionise GDB, posebno rad sa stekom, kad bi se ubacila barem jedna slika programa pokrenutog sa terminalskom verzijom istog.
Prikazan je primer jednog jako jednostavnog koda, koji ima grešku već u funkciju koju poziva main funkcija, tako da dubina steka nije značajno velika.

2.4 Provera sadržajnosti i forme seminarskog rada

1. Da li rad dobro odgovara na zadatu temu?
Da, tema je GDB debager, čiji svi aspekti su obrađeni u okviru rada.
2. Da li je nešto važno propušteno?
Sve neophodno je obrađeno, ali kada su u pitanju teme sa debugovanjem uvek je dobra ideja dati malo realnije ili barem raznolike primere koji će zaista dočarati čitaocu koliko je moćan dotični debager. Preporučujem da se konkretno doda primer gde se koristi stek poziva funkcija.
3. Da li ima suštinskih grešaka i propusta?
Osim krupnih primedbi u sekciji iznad smatram da nema propusta.
4. Da li je naslov rada dobro izabran?
Koristi zadatu ključnu reč u naslovu, što je odlično, jedino možda nije toliko primamljiv jer više zvuči kao naslov u udžbeniku. [Možda da prome-](#)
[nimo, ali da ne budemo ovako "šaljivi" kao recenzent](#) Možda bi bilo bolje nazvati ga "GNU GDB debager pod mikroskopom" ili nešto u sličnom duhu.
5. Da li sažetak sadrži prave podatke o radu?
Da, sažeto, precizno i pravilno opisuje sadržaj rada.
6. Da li je rad lak-težak za čitanje?
Rad je izuzetno lak za čitanje. Nove informacije se uvode umereno, tako da čitalac ni u jednom trenutku nema potrebu da se vraća unatrag, tome doprinosi i što je stil pisanja vrlo pitak.
7. Da li je za razumevanje teksta potrebno predznanje i u kolikoj meri?
Za tekst je potrebno osnovno znanje C++ programskog jezika isključivo zbog jednog primera kao i elementarno znanje o kompajliranju i radu u terminalu.
8. Da li je u radu navedena odgovarajuća literatura?
Obe primedbe su navedene u većim i manjim primedbama. Jedna stavka u literaturi je diskutabilna, dok je u drugom slučaju tekst upitnog izvora. Ali osim toga su navedene sve potrebne stavke u literatur.
9. Da li su u radu reference korektno navedene?
U svim osim jednog gore navedenog slučaja, reference su pravilno iskorišćene. Navedene su na ispravnim mestima i uzete iz proverenih izvora.

10. Da li je struktura rada adekvatna?
Jeste, redosled obrađenih tema je vrlo logičan i prirodan. Svaki put kada uvodi koncept kreće od generalnog i ide ka konkretnom konceptu.
11. Da li rad sadrži sve elemente propisane uslovom seminarskog rada (slike, tabele, broj strana...)?
Da, autori su ispoštovali sve propisane uslove propisane za pisanje seminarskog rada.
12. Da li su slike i tabele funkcionalne i adekvatne?
Izbor korišćenja tabela je vrlo koristan jer bi u suprotnom prezentacija komandi bila nečitljiva. Slike jesu adekvatne, s tim što sam naveo u sitnim primedbama da fali slika GDB dok je pokrenut u terminalu. Takođe bih preporučio autorima da prošire sliku is QT creator-a na kojoj je prikazana greška, baš zbog razloga naveneom u krupnim primedbama.

2.5 Ocenite sebe

Veoma sam upućen u oblast. Radio sam sa GDB debagerom popriličnu količinu vremena i intenzivno sam koristio debugere i u QT kreatoru i u JetBrains orkuženjima, kao na primer CLion, IntelliJ, Pycharm i Android Studio. U navedenim okruženjima sam tražio i popravljao i svoje i tuđe greške raznovrsnih tipova.

Glava 3

Dodatne izmene