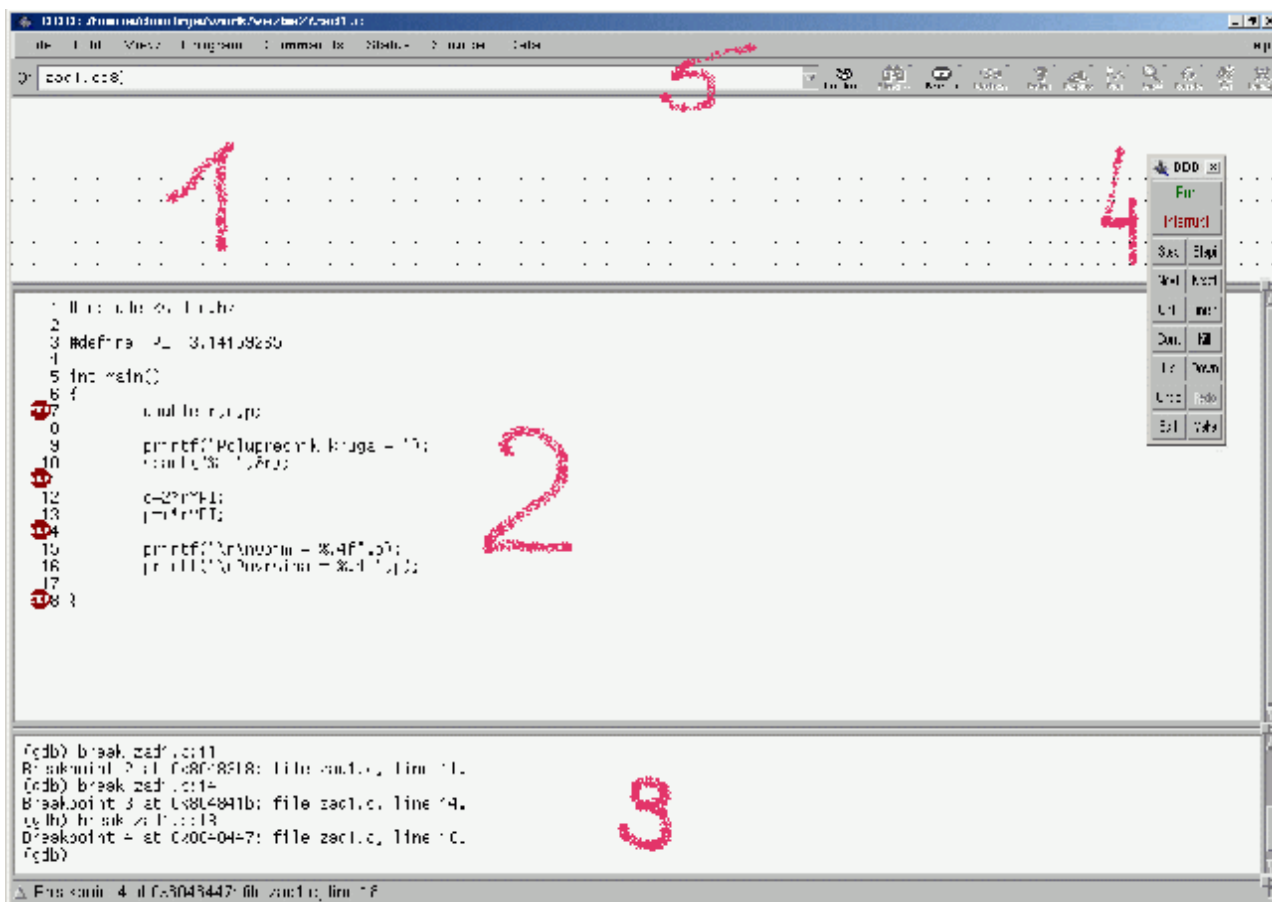


# DEBAGIRANJE UPOTREBOM DDD ALATA KROZ PRIMER

Debager je alat za kontrolisano izvršavanje programa i unutar njega se mogu stalno pratiti i menjati sadržaji promenljivih i postavljati prekidne tačke. Debager u Linuxu se zove DDD (Data Display Debugger) i predstavlja grafičko okruženje za **gdb** (GNU Debugger), osnovni debager koji radi iz komandne linije.

Upotreba ddd alata biće prikazana kroz primer dat u zad1.c.

1. Iz komandne linije prevesti izvornu datoteku zad1.c pozivom naredbe: `gcc -g zad1.c -o zad1`. Opcija `-g` specificira da se u izvrsnoj datoteci dodaju informacije koje debager koristi za kontrolisano izvršavanje.
2. Iz komandne linije pozivamo ddd debager za program zad1 na jedan od sledeća tri načina:  
`$ ddd zad1 &`  
`$ ddd zad1`  
`$ ddd`
3. Na slici 1.1 dat je prikaz ddd okruženja neposredno nakon poziva prethodne komande.



Okruženje na slici se sastoji iz:

Data Window (1) – panel koji može sadržati promenljive programa

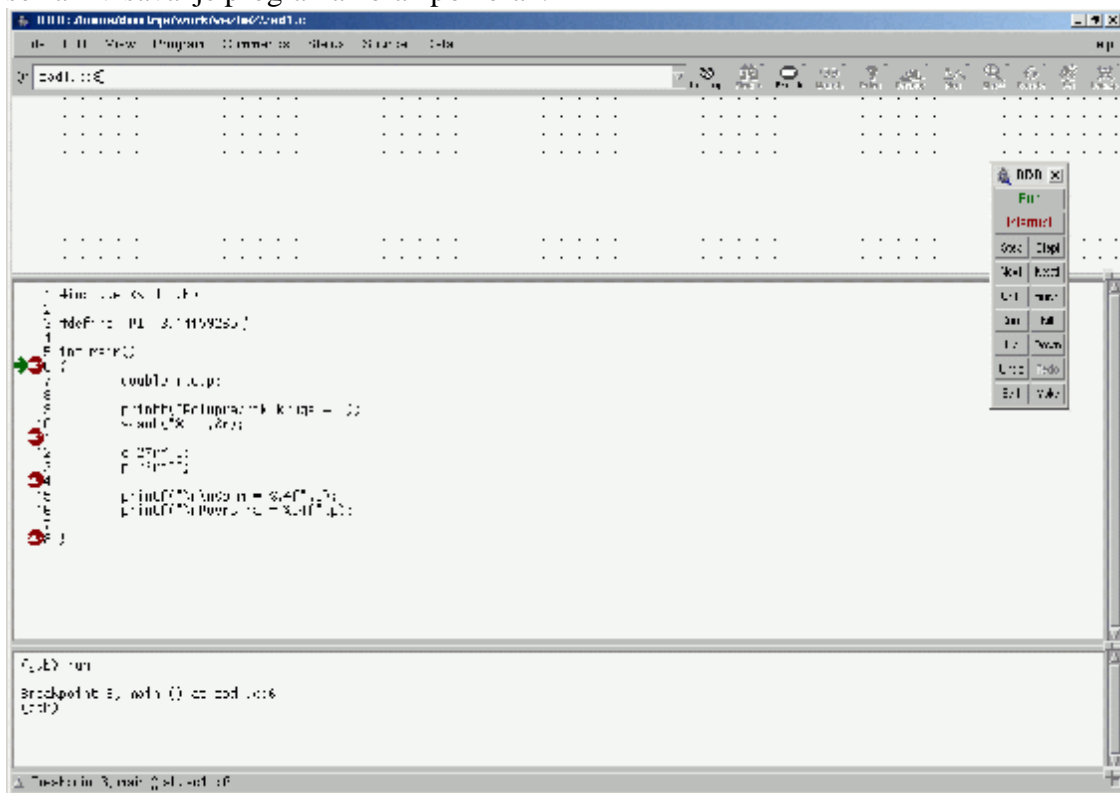
Source Window (2) – source programa koji se debugira  
Debugger Console (3) – vizuelizacija kroz pozive gdb komandi  
Command Tool (4) - prečice komandi koje se mogu naći u glavnom meniju  
Tool Bar (5) –

4. Dodati tačke prekida na linijama 7, 11, 14, 18. Tačka prekida se dodaje na jedan od tri načina:

- Kursor se postavi na pocetak linije, pa se mišem pritisne na dugme Break u gornjem meniju. Ako u toj liniji vec postoji tacka prekida, dugme se menja u Clear.
- Pritisne se desnim dugmetom miša nad rednim brojem linije i iz pop-up menija koji se pojavi izabere opcija "Set breakpoint". Tačka prekida, ako postoji uklanja se "Delete breakpoint" komandom.
- Dvoklik mišem na početak linije

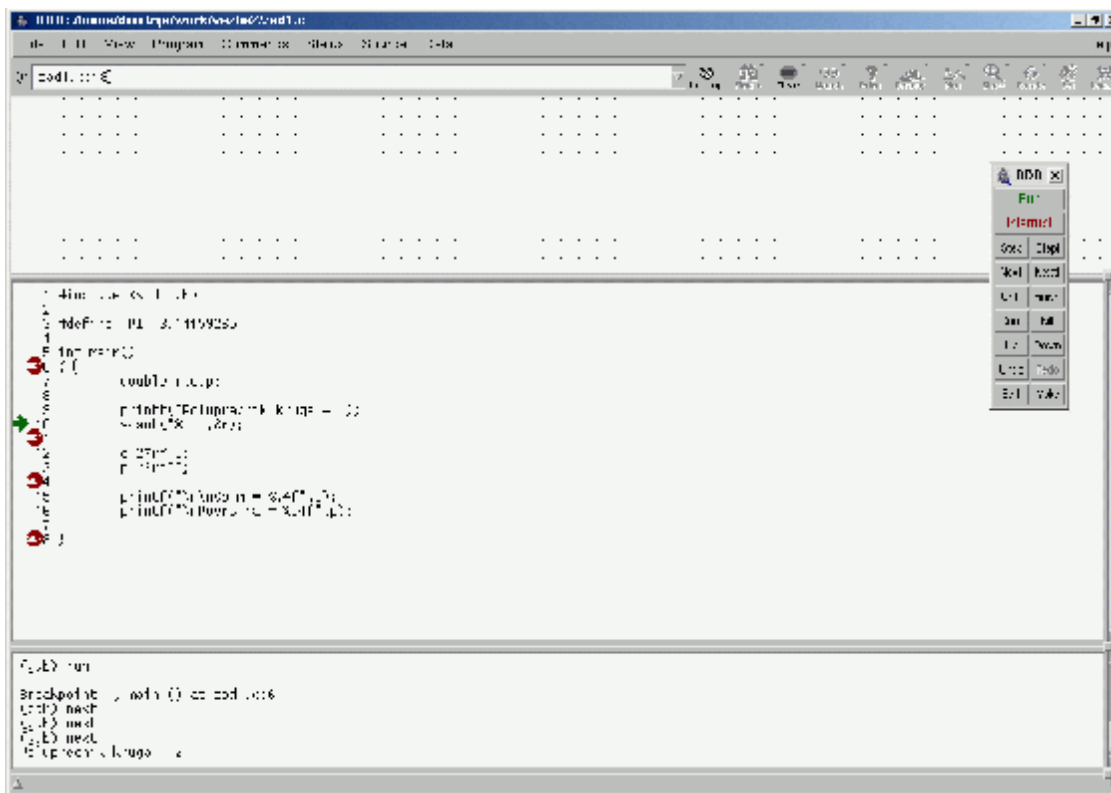
5.

Pokretanjem programa pozivom Program→Run(F2), program se izvršava do prve tačke prekida. Za kretanje u programu na raspolaganju su komande, **Next**(F5), **Step**(F6) i **Cont**(F9). **Step**i izvršava tekuću naredbu i nakon toga ponovo zaustavlja izvršavanje programa. Koriste se za izvršavanje programa korak po korak.

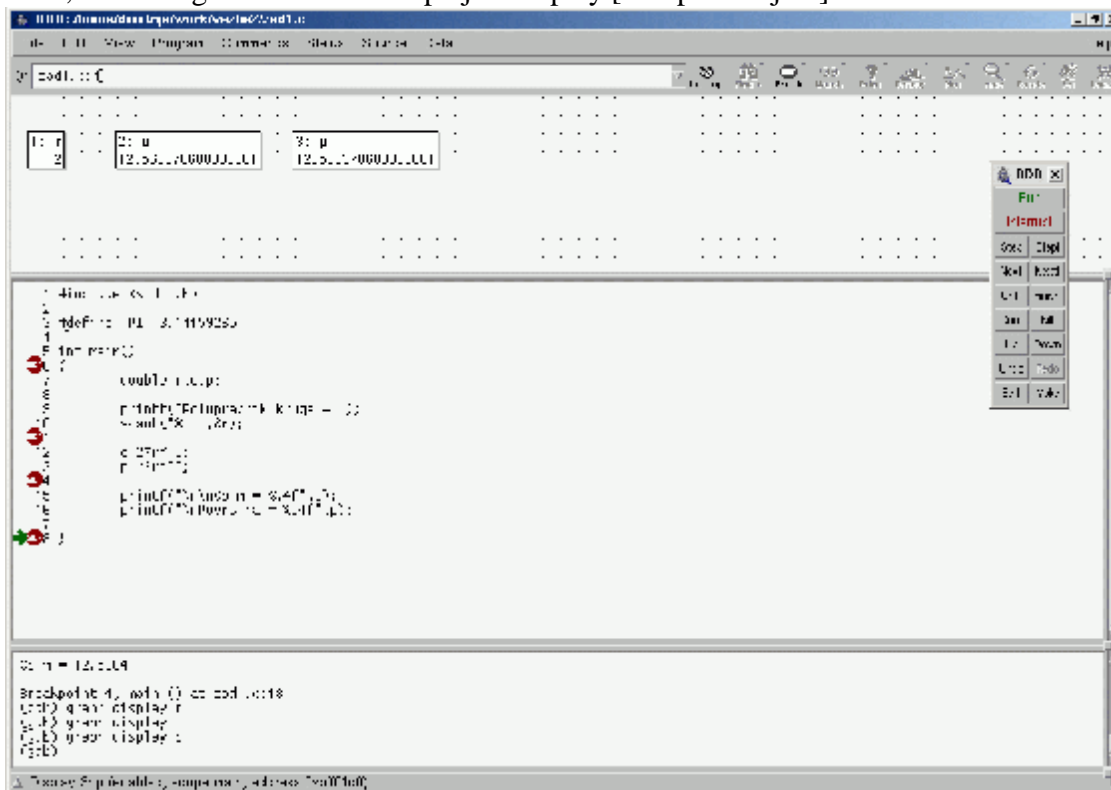


Razlika između ove dve komande se ogleda kod poziva potprograma: step izvršava potprogram korak po korak, dok **Next** izvrši ceo potprogram i zaustavlja se na naredbi koja sledi naredbu poziva potprograma.

**Cont** nastavlja izvršavanje programa do sledeće tačke prekida. U našem slučaju, nakon poziva komande **Run**, program će se izvršiti do linije 6, kao na slici 1.2 gde pokazuje zelena strelica. Daljim kretanjem pozivom komandi **Step** ili **Next**, program će stati u liniji 10, gde očekuje unos poluprečnika od strane korisnika u prozoru dbg-a.



6. Prilikom debugovanja omogućen je prikaz sadržaja promenljivih u gornjem panelu. Dodavanje promenljive se postiže pozicioniranjem kursora miša nad promenljivom, i pritiskom na desni taster miša, nakon čega treba odabrati opciju "Display [ime promenljive]"



## 7. Za prekid izvršavanja programa dostupna je komanda Kill(F4)

U nastavku je dat prikaz komandi za rad sa debugiranim programom, koje su dostupne iz Programs podmenija glavnom menija, ili iz Command Tool panela.

Run	Početak izvršavanja programa, sa zahtevom za zadavanje argumenata programu
Run Again	Početak izvršavanja programa sa argumentima programu zadatim u prethodnom izvršavanju.
Run in Execution Window	Ako je omogućeno, počinje izvršavanja drugog programa u posebnom prozoru.
Step	Izvršava program do naredne linije koda, i potom se zaustavlja i vraća kontrolu ddd-u.
Step Instruction (Stepi)	Izvršava jednu mašinsku instrukciju, potom se zaustavlja I vraća kontrolu ddd-u.
Next	Nastavlja izvršavanja na sledećoj linij koda. Ovo je slično "Step" komandi, ali se pri pozivu funkcije u liniji koda, ne ulazi u izvršavanje naredbi unutar funkcije.
Next Instruction (Nexti)	Izvršava jednu mašinsku instrukciju, a ako je u pitanju poziv funkcije, izvršava se dok se ne dobije povratna vrednost.
Until	
Finish	
Continue	Nastavlja izvršavanje programa do pojave naredne tačke prekida.
Continue Without Signal	Nastavak izvršavanja bez slanja signala.
Kill	Završava proces debugiranog programa.
Interrupt	Prekida izvršavanje programa. Ovo je ekvivalentno slanju signala prekida procesu.
Abort	Prekida izvršavanje programa. Ekvivalentno je slanju signala SIGBART procesu.