Uputstvo za korišćenje GDB-a

GNU debager ili **GDB** je standardan debager za GNU operativni sistem. To je prenosiv (*portable*) debager koji radi na mnogim uniksolikim sistemima i koristi se za mnoge programske jezike. GDB se isporučuje kao sastavni deo većine GNU/Linux distribucija. Ukoliko GDB ne postoji na sistemu, može se instalirati koristeći sledeće dve naredbe:

sudo apt-get update

sudo apt-get install gdb

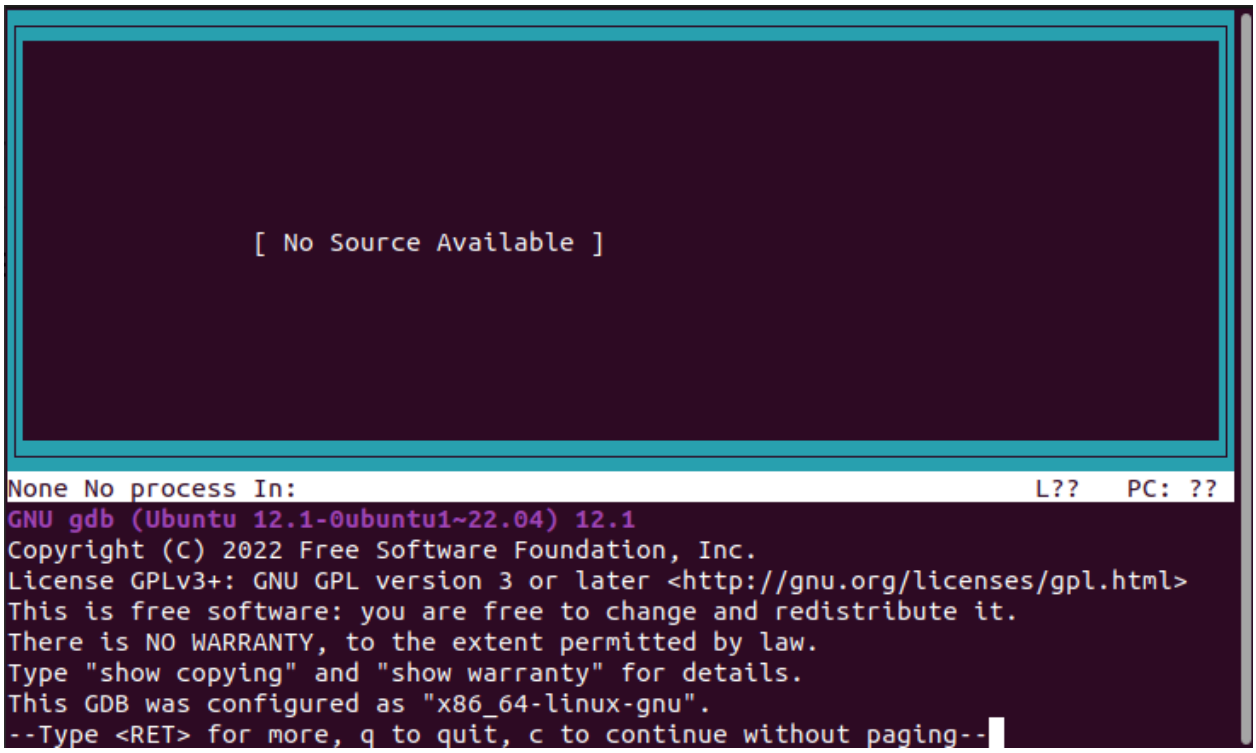
Pre samog korišćenja GDB-a obavezno je kompajliranje programskog koda sa parametrom **-g** iz foldera gde se nalazi fajl sa kodom. (Savet: otvoriti terminal preko celog ekrana zbog veće preglednosti.) U slučaju asemblerskog koda naredba za kompajliranje je:

gcc -m32 -g naziv_fajla.S

Kompajliranjem se kreira izvršni fajl pod nazivnom a.out (u slučaju da prilikom kompajliranja nije naveden željeni naziv izvršnog fajla). Da bi se započelo korišćenje GDB-a sa njegovim korisničkim interfejsom potrebno je pokrenuti sledeću naredbu:

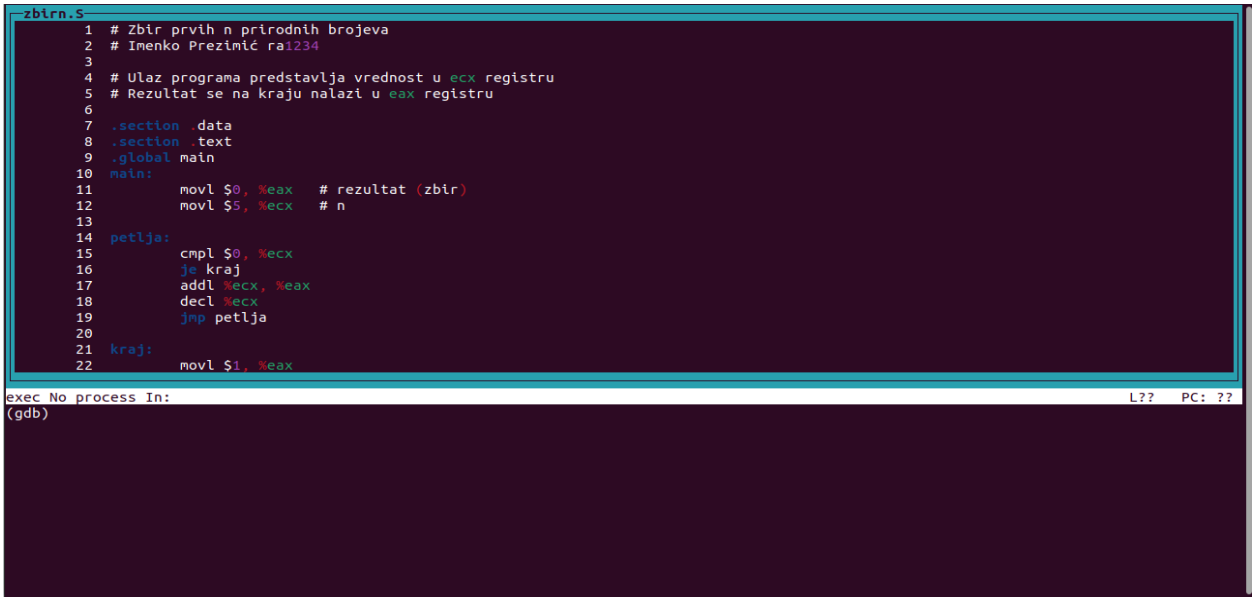
gdb -tui a.out

Ako se pri pokretanju GDB-a ispiše poruka *[No Source Available]* (slika 1) potrebno je kliknuti taster **ENTER**.



Slika 1

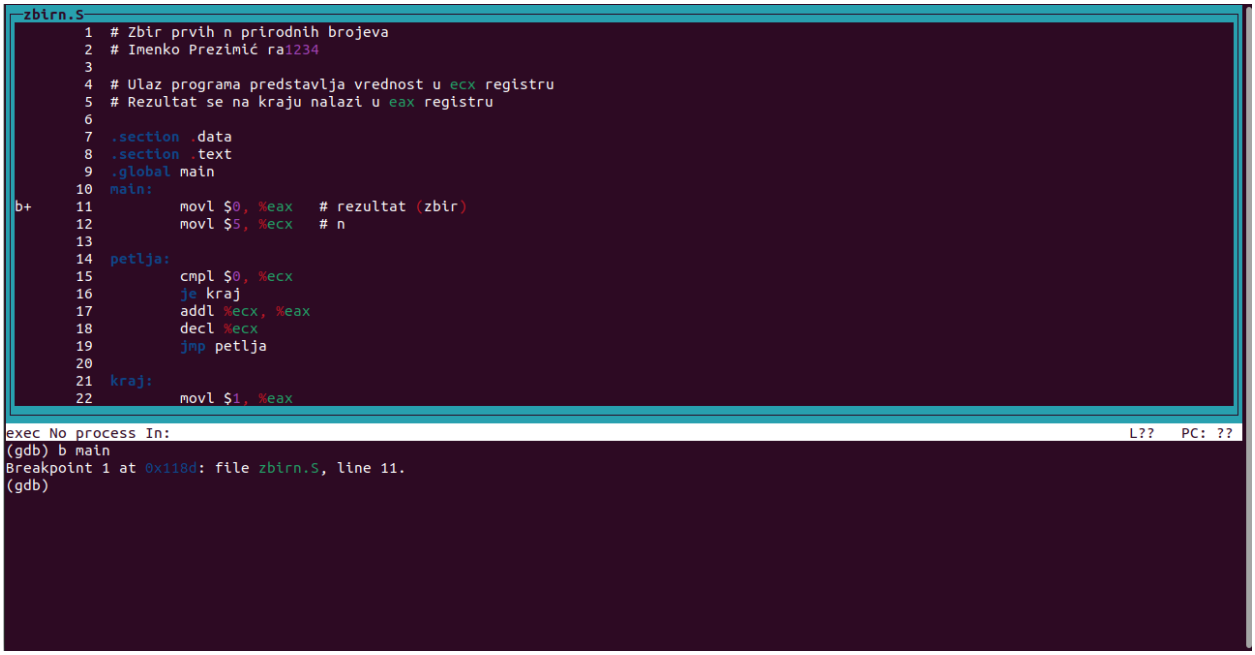
Kada je GDB uspešno pokrenut, u gornjem delu terminala trebalo bi da bude vidljiv kod programa, dok bi u donjem delu (prostor predviđen za komande) trebalo da u poslednjoj liniji piše ***(gdb)*** što znači da je GDB spreman da prima naredbe (slika 2).



Slika 2

Da bi se počelo sa debagovanjem koda potrebno je postaviti *breakpoint* na početak *main*-a (slika 3), što je moguće naredbom:

break main ili **b main**

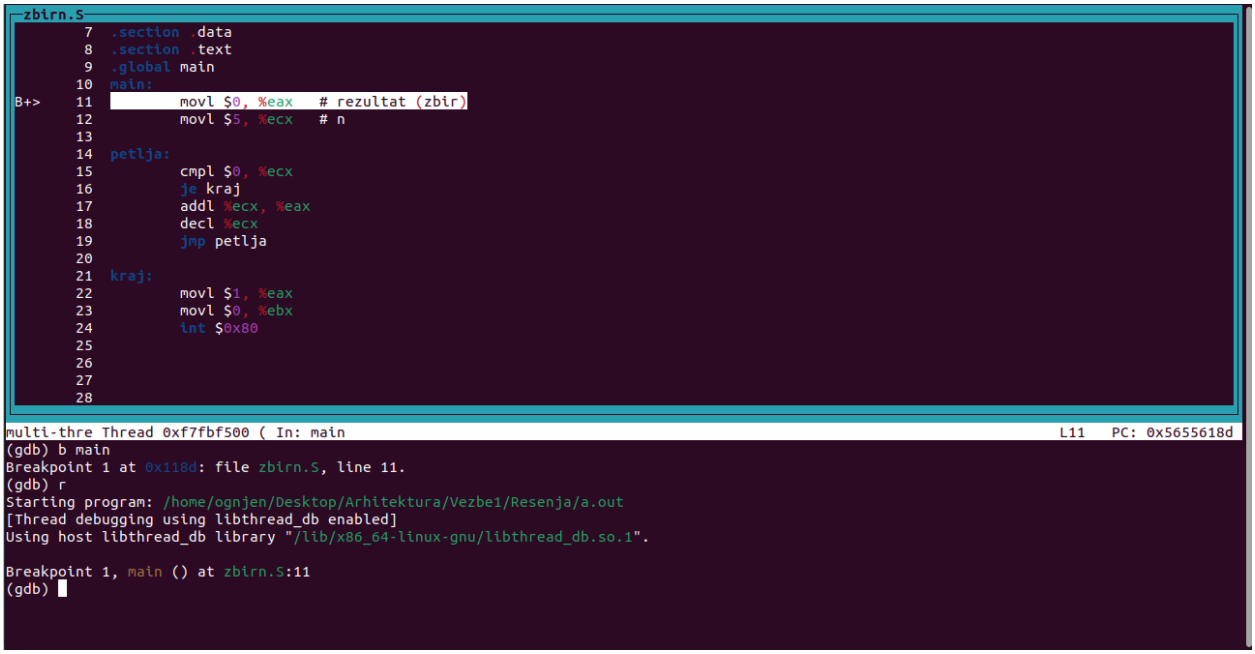


Slika 3

Naredba za pokretanje izvršavanja programa:

run ili **r**

Kada se ova naredba izvrši, u prozoru u kom je prikazan kod programa linija koja treba da bude izvršena sledeća će biti osenčena (slika 4).

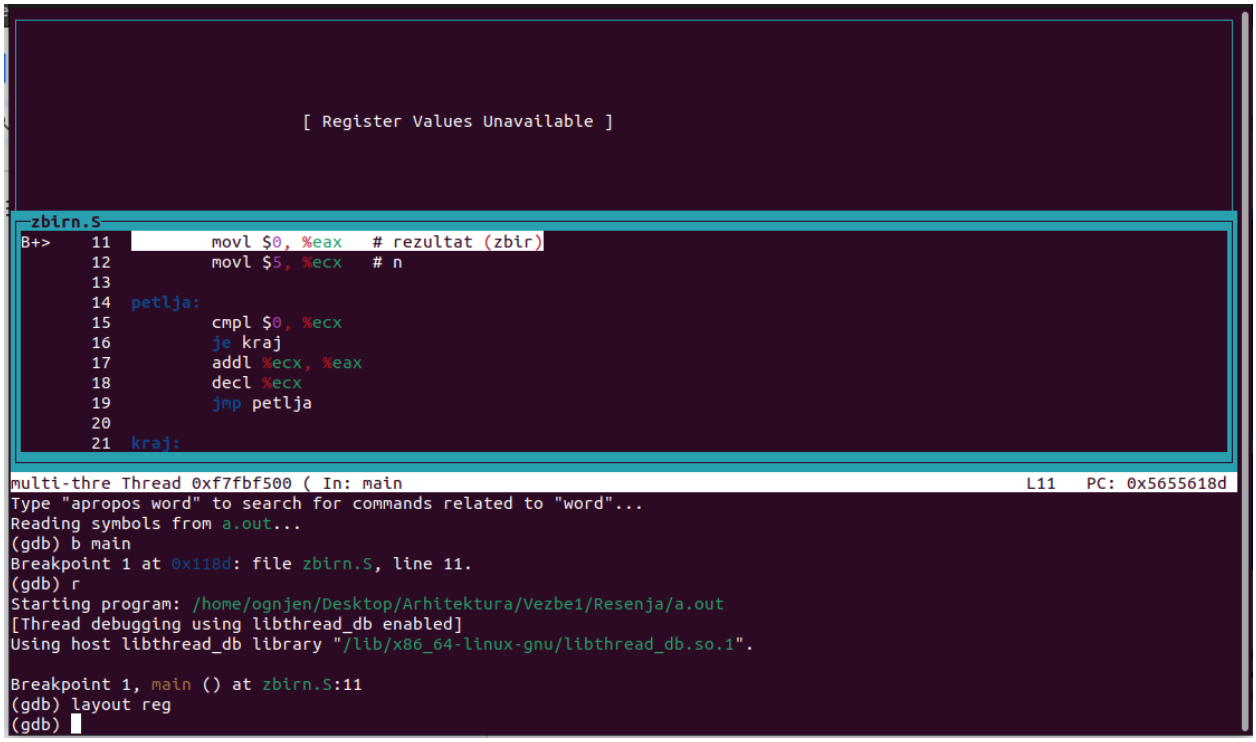


Slika 4

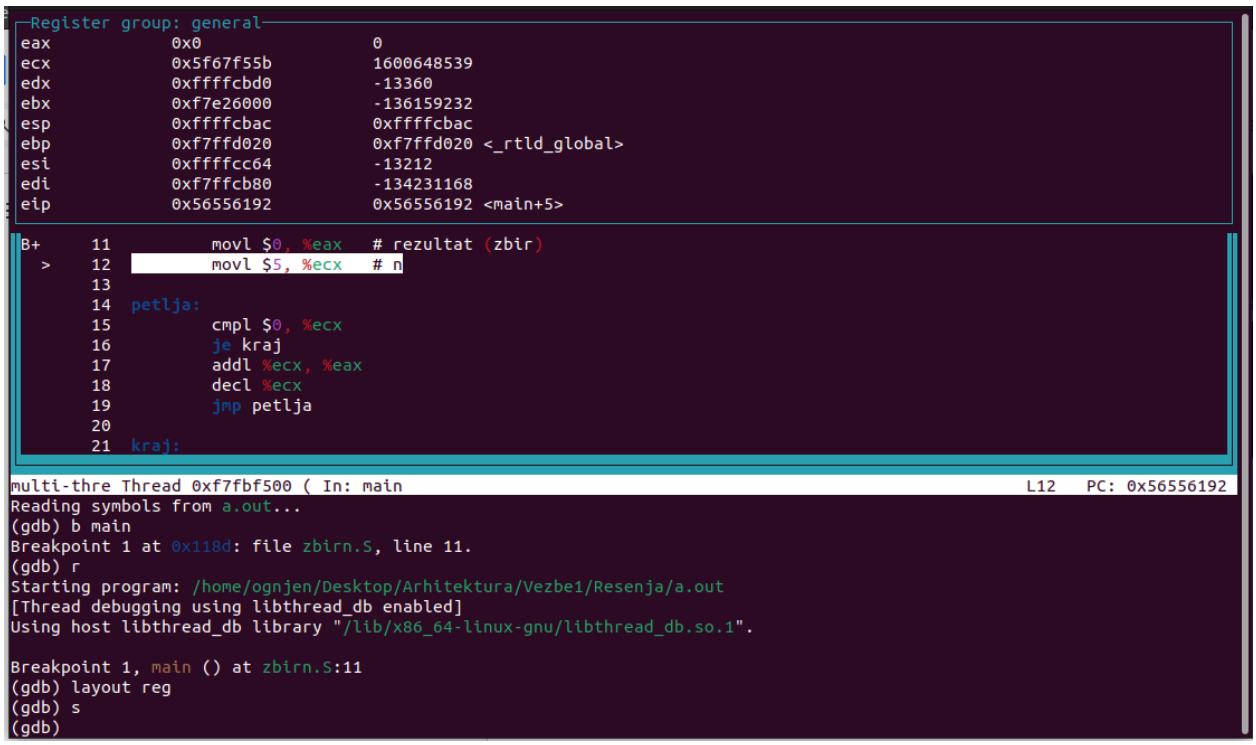
Otvaranje prozora sa vrednostima registara:

layout reg

Nakon ove naredbe u gornjem delu terminala se otvara novi prozor kroz koji je moguće pratiti vrednosti registrara (slika 6). U većini slučajeva kada se ovaj prozor otvori u njemu bude ispisana poruka [Registers Values Unavailable] (slika 5), sve što je potrebno uraditi u tom slučaju je pomeriti se na sledeću liniju koda što se obavlja naredbom **step**.



Slika 5



Slika 6

Ako je u nekom trenutku potrebno posebno ispisati vrednost određenog registra to je moguće naredbom:

print $eax ili **p $eax**

- podrazumevano je da se vrednost smeštena u registru prikazuje u decimalnom i heksadecimalnom obliku, ako je potrebno prikazati vrednost u binarnom obliku koristi se sledeća naredba:

  print /t $eax ili **p /t $eax**

- takođe, za ispis vrednosti registrara manjeg kapaciteta koristi se ista naredba:

  print $al ili **p $al**

Naredbe za kretanje kroz kod:

- izvršavanje sledeće naredbe bez ulaska u funkciju: **next** ili **n**
- izvršavanje slešeće naredbe sa ulaskom u funkciju: **step** ili **s**
- postavljanje *breakpoint*-a na početak određenje funkcije (labele):

  break <naziv_labele> ili **b <naziv_labela>**

- postavljanje *breakpoint*-a na određenu liniju koda:

  break <broj_linije> ili **b <broj_linije>**

- nastavak izvršavanja programa do sledećeg *breakpoint*-a: **continue** ili **c**
- brisanje svih postavljenih *breakpoint*-a: **clear**
- kada poslednja izvršena naredba treba ponovo da se izvrši dovoljno je samo pritisnuti taster **ENTER**, bez potrebe za ponovnim kucanjem naredbe

Rad sa promenljivama:

- prikaz vrednosti promenljive: **x &<naziv_promenljive>**
  - ako je potrebno konvertovati vrednost promenljive u određeni brojni sistem:

    x /<brojni_sistem> &<naziv_promenljive>

    - **/t** - za binarni brojni sistem
    - **/d** - za decimalni brojni sistem

- **/x** - za heksadecimalni brojni sistem
- **/o** - za oktalni brojni sistem
- **/c** - za prikaz karaktera koji odgovara *ascii* vrednosti smeštene u promenljivoj
- primer: Prikaz vrednosti promenljive *a* u decimalnom obliku
    - x /d &a

- prikaz niza:

  x/<broj_elemenata_niza><veličina_elementa><brojni_sistem> &<naziv_niza>

    - moguće veličine elemenata niza:
        - bytes (1 bajt): **/b**
        - halfwords (2 bajta): **/h**
        - words (4 bajta): **/w**
        - giants (8 bajta): **/g**
    - primer: Prikaz vrednosti elemenata niza pod nazivom *niz* koji sadrži 10 elemenata gde je svaki veličine 4 bajta (words - w) u heksadecimalnom obliku (x)
        - x/10wx &niz

- kada se radi sa nizovima često je potrebno prikazati vrednost trenutnog elementa niza
    - indirektno adresiranje (ako je adresa niza smeštena u registru eax)

      x/<veličina_elementa_niza><brojni_sistem> $eax

        - primer: Prikaz vrednosti trenutnog elementa niza (ako je adresa niza smeštena u registru eax), gde je svaki element veličine 1 bajt (bytes) u binarnom obliku (t)
            - x/bt $eax
    - indeksno adresiranje (gde je indeks smešten u registru eax)

      x ((char *)&*niz*+$*eax4)**

- prikaz stringa: **x/s &<naziv_stringa>**
    - prikaz stringa kao niza karaktera:

x /<broj_karaktera>c &<naziv_stringa>

Kod rada sa potprogramima često je potrebno prikazati vrednost prosleđenih argumenata potprograma pomoću registra ebp

- pošto se vrednosti prvog paratra potrogram pristupa kao 8(%ebp), njegovu vrednost je moguće prikazati pomoću naredbe:

x /<brojni_sistem> $ebp+8

- za prikaz ostalih parametara potrebno je pratiti šablon:

x $ebp+broj_bajtova_na_steku_u_odnosu_na_ebp

- na isti način se pristupa i lokalnim promenljivama, pa tako se vrednost prve lokalne promenljive dobija kao: **x $ebp-4**

Prilikom rada sa gdb fokus je postavljen na deo sa kodom i nije moguće pristupiti prethodnim naredbama pritiskom strelice na gore. Da bi se fokus prebacio na komandnu liniju pokrenuti komandu:

focus cmd

Da bi vratili fokus nazad na kod pokrenuti komandu:

focus src

Izlaz iz GDB-a: **quit** ili **q**