RI 201
Arhitektura računara

Uvod u GDB

Organizacija

- Auditorne i laboratorijske vježbe: 35 bodova
- Testovi na laboratorijskim vježbama: 20 bodova
- Završni projekat: 15 bodova

Kompajleri i ELLCC

- Proces prevođenja koda iz jezika visokog nivoa u jezik nižeg nivoa
- Osnovni zadaci kompajler toolchain-a su:
  - Prevođenje izvornog koda (npr. C, C++) u asemblerski kod za datu arhitekturu
  - Otkrivanje sintaksnih i semantičkih grešaka u izvornom kodu
  - Asembliranje i uvezivanje programa

- ELLCC predstavlja kompajlerski lanac koji omogućava kompajliranje koda za veliki broj procesorskih arhitektura
- Na ovom kursu ćemo koristiti ELLCC za MIPS arhitekturu

Analiziranje grešaka u kodu

- Tipovi grešaka:
  - Sintaksne greške
  - Semantičke greške
  - Logičke greške

Sintaksne greške

```
int a = 3
x = 5 * (2 + 3;
```

Semantičke greške

```
int a;
int b = 10 / a;
```

(1)

```
int a[4];
a[4] = 6;
```

(2)

```
int saberi(int a, int b) {
    return a + c;
}
```

(3)

Logičke greške

```
int saberi(int a, int b) {
    return a - b;
}
```

Analiziranje grešaka u kodu

- Kompajliranjem možemo otkriti sintaksne i neke semantičke greške u kodu
- Za većinu semantičkih grešaka kompajleri izdaju **upozorenja**

```
test2.c: In function ‘main’:
test2.c:5:11: warning: overflow in implicit constant conversion [-Woverflow]
   int a = 0xFFFFFFFFFFF;
           ^
```

Analiziranje grešaka u kodu

- Semantičke greške koje kompajler ne prepozna, kao i logičke greške otklanjamo alatom koji se naziva **debugger**
- Debugger je alat koji služi za dinamičku analizu programa
- Na ovom kursu koristićemo debugger iz GNU projekta pod nazivom GDB (GNU Debugger)

GDB

- Kompajliranje programa za debugiranje:

```
$ gcc -g test.c -o test
```

test.c

```
#include <stdio.h>
Int main() {
        int a = 5;
        while(a > 0) {
                printf(“a = %d\n”, a);
                ++a;
        }
        return 0;
}
```

- Nakon kompajliranja možemo pokrenuti debugger komandom:

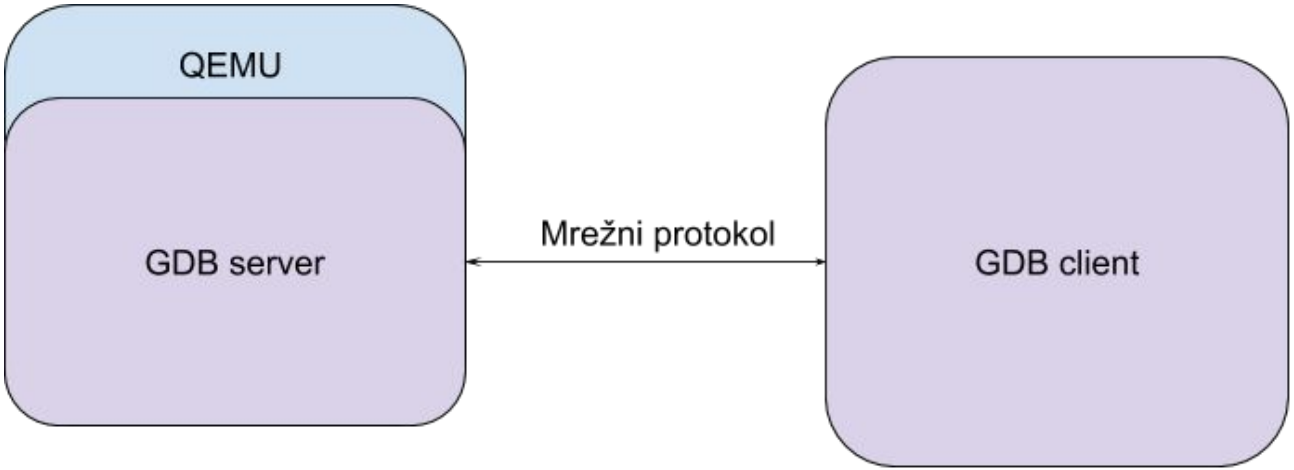
```
$ gdb test
```

GDB

```
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from debug...done.
(gdb)
```

QEMU i GDB server

- Za pokretanje izvršnih fajlova koji targetiraju MIPS platformu koristimu QEMU
- Osim emulacije date procesorske arhitekture, QEMU omogućava i debugiranje aplikacija
- Za debugiranje aplikacija QEMU koristi GDB server



Osnovne GDB komande

- `target remote host:port`
  - GDB klijent se spaja na GDB server kojeg očekuje da sluša na navedenoj IP adresi i portu. Ukoliko se host izostavi, podrazumijeva se localhost
- `break adresa`
  - Postavlja break point na navedenoj adresi ili simbolu
- `si/ni`
  - si (step instruction) izvršava trenutnu instrukciju na koju pokazuje programski brojač
  - ni (next instruction) slično si komandi, stim što u slučaju da je sljedeća instrukcija skok na neku funkciju, si ulazi u funkciju, dok ni izvršava kompletnu funkciju i nastavlja izvršenje nakon povratka
- `x/NFU`
  - Gdje N predstavlja broj memorijskih jedinica za prikaz, F predstavlja format ispisa (x -heksadecimalni ispis, i-ispis instrukcija, s-null terminirani string...) i gdje U predstavlja veličinu memorijske jedinice za prikaz (b - byte, h - half word, w - word, g - giant)
- `list, info registers, print simbol, r, c`