AR - Vežbe 7 - Uvod u asembler

Mladen Vidović
mvidovic@singidunum.ac.rs

Univerzitet Singidunum
Centar Novi Sad

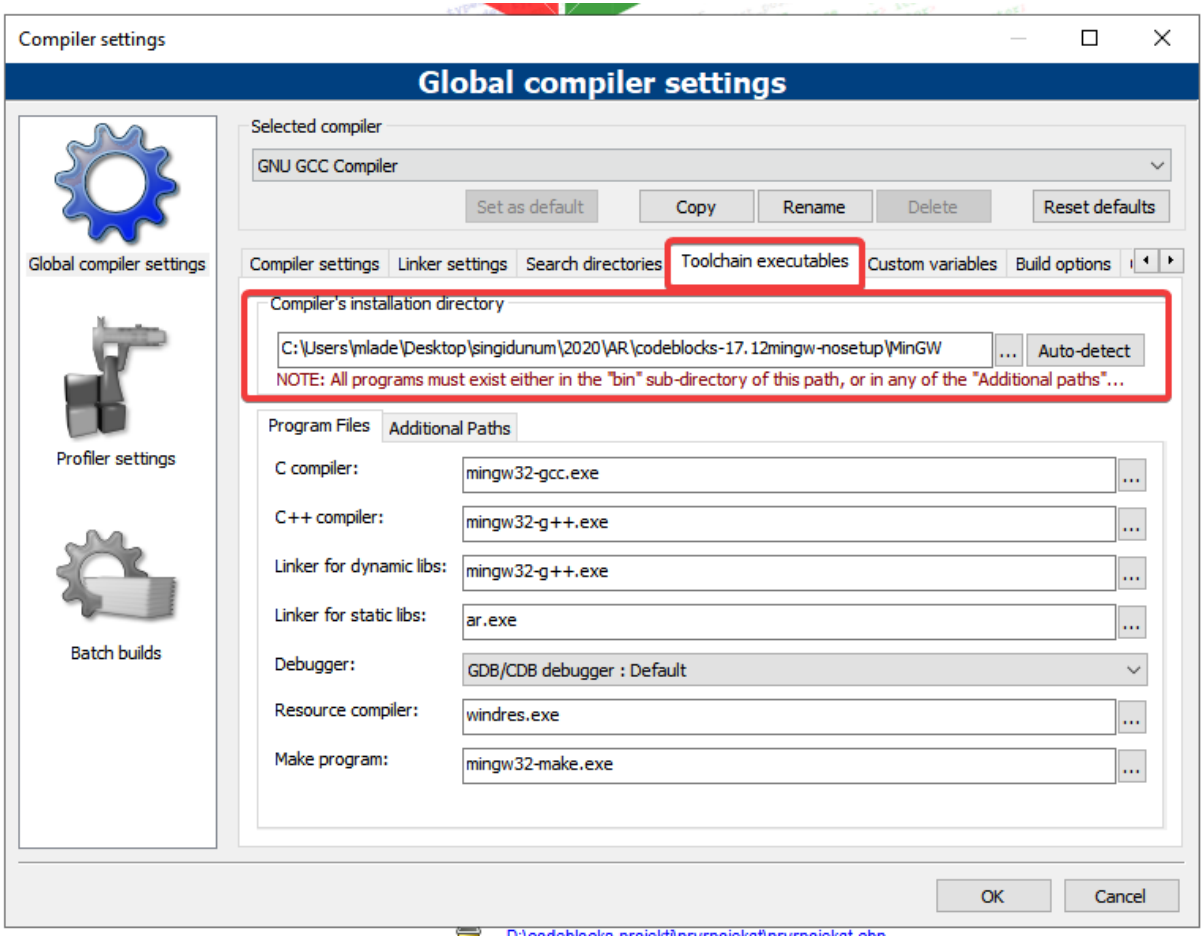
5. novembar 2024.

Codeblocks

- Preuzeti codeblocks na sledećoj adresi: `https://sourceforge.net/projects/codeblocks/files/Binaries/17.12/Windows/`
- Odabrati mingw, nosetup verziju
- Raspakovati na proizvoljnu lokaciju, pa pokrenuti

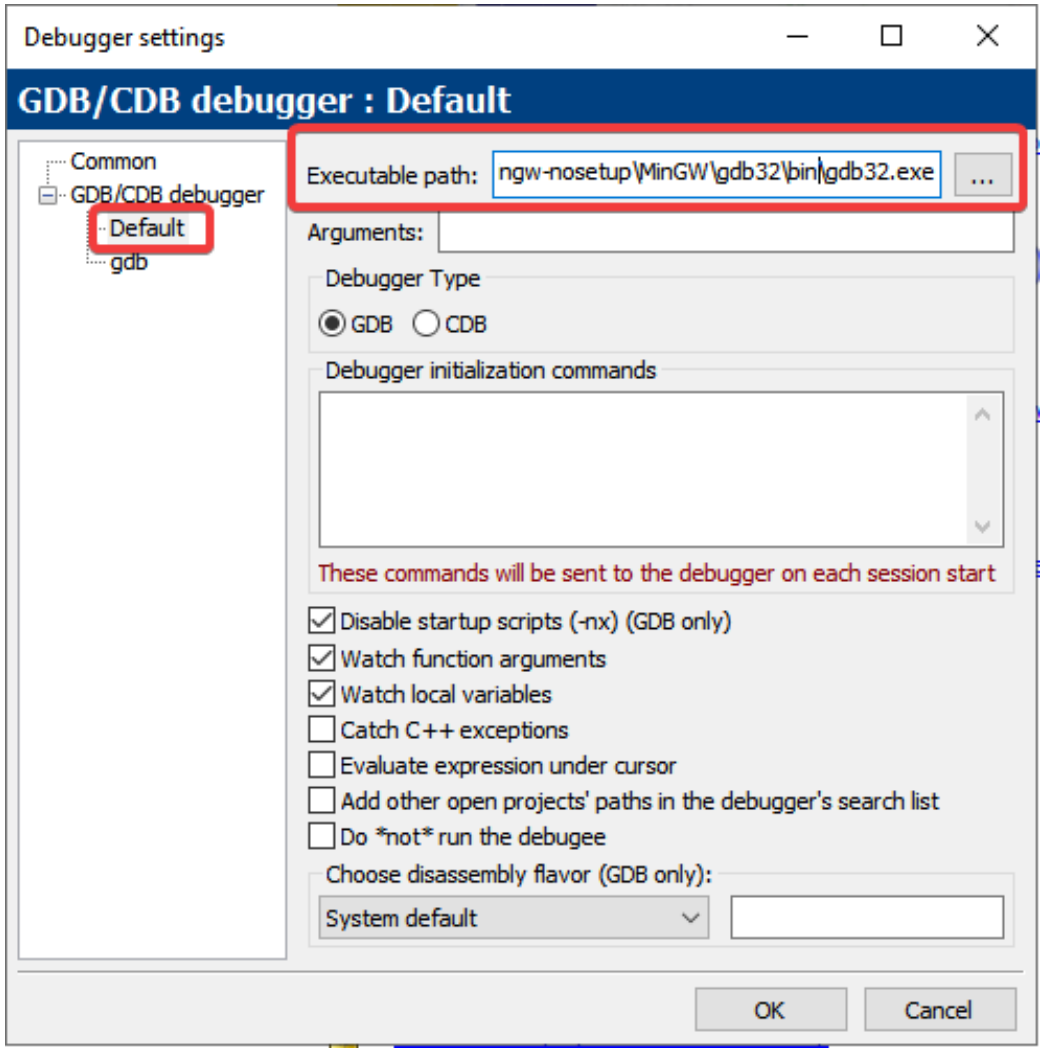
Podešavanje kompajlera

- Settings $->$ Compiler...
- Podesiti u Toolchain executables putanju do codeblocks direktorijuma/MinGW



Podešavanje dibagera

- Settings $->$ Debugger...
- Kreirati novu konfiguraciju, ili za default konfiguraciju podesiti putanju do codeblocks lokacije/MinGW/gdb32/bin/gdb32.exe



Kreiranje projekta i priprema za rad

- File $->$ New $->$ Project
- Odabrati Empty project kao template
- Kreirati novu datoteku za projekat File $->$ New $->$ Empty file
- Dodati datoteku u projekat (dijalog) i sačuvati obavezno sa ekstenzijom .s

Primer

```
#prvi primer
.section .data
.section .text
.global _main
_main:
movl $10, %eax
xorl %eax, %eax
ret
```

Algoritamsko rešavanje problema

- draw.io https://app.diagrams.net/
- Početak
- Kraj
- Iskaz
- Input
- Grananje

Zadaci

- Napisati program koji poredi vrednosti promenljivih A i B.
- Ukoliko je A veće od B, B se uveća za 10.
- Ukoliko je A manje od B, A se uveća za 15.
- Ukoliko su A i B jednaki, oba broja se uvećaju za 5.

Zadaci

- Kreirati program koji poredi vrednosti promenljivih A i B. Sve dok je vrednost promenljive A manja od promenljive B, inkrementirati vrednost promenljive A. U promenljivu count upisati broj inkrementacija promenljive A.

Uvod u sintaksu

- Section - deo programa
- .section < naziv sekcije >
- .section .data - sekcija za definisanje podataka
- .section .text - sekcija sa programskim kodom

Zadaci

- Implementirati množenje dve promenljive preko sabiranja. Rezultat množenja zapisati u promenljivu proizvod.

Nizovi

- niz: .long 15, 21, 24, 56, 32
- niz2: .long 4, 14, 52, 37, 212
- Napisati program koji prolazi kroz oba niza, sabira vrednosti, i zbir zapisuje u novi izlazni niz

Nizovi

- niz: .long 5, 12, 23, 45, 1, 156, 18
- Napisati program koji prebrojava koliko parnih brojeva ima u nizu. Smestiti rezultat u promenljivu broj.

Uvod u sintaksu

- Labele služe za označavanje dela koda, odnosno specifične linije koda.
- <naziv labele>:
- primer:
- _main labela je specijalna labela, i preko komande .global _main se deklariše da je to početak izvršavanja programa

Uvod u sintaksu

- Opšti oblik komande
- mnemonik(sufiks) operand, operand
- Neke instrukcije imaju jedan operand, neke nemaju nijedan
- Ako instrukcija ima dva operanda, prvi je izvorišni operand, a drugi je i izvor i odredište
- pročitaju se vrednosti oba operanda, izvrši se instrukcija, i rezultat izvršavanja instrukcije se smesti u drugi operand
- mov operand, operand - smešta se vrednost prvog operanda u drugi operand
- add operand, operand - vrednosti oba operanda se saberu, i rezultat se smesti u drugi operand
- sub operand, operand - od drugog operanda se oduzme prvi, i rezultat se smesti u drugi operand

Uvod u sintaksu

- Sufiks se dodaje na kraj mnemonika i označava veličinu operanada sa kojim se radi
- b - byte 8 bita
- w - word 16 bita
- l - long 32 bita
- movb - premesti vrednost veličine 1 bajt
- addw - saberi vrednosti veličine 2 bajta
- subl - oduzmi vrednosti veličine 4 bajta

Uvod u sintaksu

- Adresiranje operanada - preko prefiksa (ili nedostatka istog)
- % - registarsko adresiranje, nakon ove oznake sledi ime registra kojem se pristupa
- Registri su 32-bitne lokacije u koje možemo da zapisujemo i iz kojih možemo da čitamo podatke
- Ima više registara, u okviru zadataka ćemo se ograničiti na 4: eax, ebx, ecx i edx
- $ - neposredno adresiranje - konkretne vrednosti
- $20 - decimalni broj 20
- $0b1101 - binarni broj 1101 (0b je prefiks za binarne brojeve-)

Primer

```
#prvi primer
.section .data
.section .text
.global _main
_main:
movl $10, %eax
xorl %eax, %eax
ret
```

Uvod u sintaksu

- Napisati program za sabiranje i oduzimanje 2 broja.
- Prvi broj smestiti u registar eax
- Drugi broj smestiti u registar ebx
- Zbir smestiti u registar ecx
- Razliku (prvi - drugi) smestiti u registar edx

Uvod u sintaksu

- Napisati program za računanje sledećeg izraza
- (A+B) - (C+D) - (E+F)
- proizvoljno odabrati brojeve

Uvod u sintaksu

- Memorijske promenljive - adresa u memoriji na kojoj se nalazi vrednost
- oznaka: .tip vrednost
- broj: .long 25
- definišu se u .data segmentu programa
- da bismo pročitali vrednost promenljive, navodimo oznaku bez prefiksa

Uvod u sintaksu

```
#drugi primer
.section .data
a: .long 20
.section .text
.global _main
_main:
movl a, %eax
xorl %eax, %eax
ret
```

Uvod u sintaksu

- Napisati program za sabiranje i oduzimanje 2 broja.
- Prvi broj smestiti u promenljivu a
- Drugi broj smestiti u promenljivu b
- Zbir smestiti u promenljivu zbir
- Razliku (prvi - drugi) smestiti u promenljivu razlika

Zadaci

- Napisati program koji predstavlja logički kalkulator. Promenljive A i B predstavljaju 8bitne binarne brojeve. Izračunati rezultat operacija I, ILI, NI, NILI i EKSKLUZIVNO ILI ova dva broja, i rezultat smestiti u odgovarajuće promenljive.
- Mnemonici za logičke operacije su and, or, not i xor.
- Obratiti pažnju na sufikse zbog veličine operanada.