**SISTEMSKI SOFTVER**

Anja Lukic, 2016/146

#### Uputstvo za prevodjenje i pokretanje sistema

Za prevodjenje programa koristi se g++ prevodilac, verzija 4.9.4.

Instalacija odgovarajuce verzije g++ prevodioca:

*sudo add-apt-repository ppa:ubuntu-toolchain-r/test*

*sudo apt-get update*

*sudo apt-get install gcc-4.9*

*sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.9 50*

*sudo apt-get install g++-4.9*

*sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.9 50*

U okviru foldera src u projektu, pokretanjem komande :

*make -f Makefile*

generise se program asm, koji se pokrece u istom folderu komandom:

*./asm -o outputname.o inputname.s*

#### Opis resenja

Asembler implementiran u ovom projektu je dvoprolazan i radjen po uzoru na onaj radjen na vezbama iz predmeta Sistemski softver.

U prvom prolazu asembler prolazi kroz ulazni fajl i upisuje labele i sekcije koje nalazi u tabelu simbola. Pri tome u svakoj sekciji vodi racuna o location counter-u, koji uvecava svaki put kada naidje na direktivu koja generise podatke u memoriji ili na instrukciju, koju mora da koduje dovoljno da zna koliko bajtova ima instrukcija i njeni operandi, da bi tu velicinu dodao na location counter.

Nakon prvog prolaza, asembler ima delimicno formiranu tabelu simbola, u kojoj su svi simboli (sem onih uvezenih direktivom *.extern*) lokalni.

U drugom prolazu, asembler generise sekcije i njihove tabele relokacionih zapisa.U ovom prolazu on ignorise definicije labela, pa obradjuje samo direktive koje generisu podatke u memoriji i instrukcije, koje koduje i upisuje u sekcije. Takodje, pri nailasku na direktivu *.global,* redefinise simbol u tabeli kao globalan.

Na kraju, asembler generise izlazni fajl u koji upisuje tabelu simbola i sve sekcije sa njihovim tabelama relokacionih zapisa.