*Univerzitet u Beogradu, Elektrotehnički fakultet*

*Odsek za Softversko Inženjerstvo*

*13S113SS*

Projekat iz Sistemskog Softvera

Asembler

Aleksandar Gradinac 0568/15

Ovaj dokument predstavlja kratko uputstvo za upotrebu projekta iz Sistemskog Softvera – asemblera, kao i kratak pregled rešenja. Asembler je napisan za pojednostavljenu arhitekturu procesora sa instrukcijskim setom veličine 16,

U cilju lakšeg prevođenja programa, obezbeđen je makefile. Program se provodi pokretanjem komande make u direktorijumu sa makefile-om. Kao rezultat prevođenja dobija se izvrši program nazvan assembler. Za prevođenje je neophodno da su instalirani paketi flex i bison. Ukoliko je potreban ručni prevod, potrebno je prvo generisati lekser i parser komandama flex I bison, a nakon toga prevesti fajl zajedno sa izlazom prethodne dve komande.

Asembler se sada može koristiti na sledeći način: assembler file addr [file addr ...], gde file predstavlja naziv fajla sa asemblerskim kodom, a addr predstavlja adresu početka smeštanja sekcija fajla. Asembler generiše dva izlazna fajla za svaki fajl koji se prevodi: file.o I file.o.text. file.o predstavlja fajl sa binarnim sadržajem koji je spreman da se učita u emulator radi izvršavanja. File.o.text predstavlja fajl sa tekstualnim sadržajem gde se može videti pregled sadržaja prevedenog fajla.

U sklopu predate arhive napisan je jedan test koji obuhvata 4 fajla: fileA.asm, fileB.asm, fileC.asm I system.asm, namenjen za testiranje celokupnog asemblera odjednom. U realizaciji datog programa korišćene su sve instrukcije i svi načini adresiranja.

Asembler je napisan u C-u i kompajluje se uz C99 standard i više. Samo parsiranje ulaznog teksta nije realizovano “od nule”, već se koriste generatori leksera I parsera koji za potrebe asemblera generisu lekser I parser. Gramatika je napisana tako da se bavi isključivo parsiranjem fajla I popunjavanjem odgovarajućih struktura, koje se dalje prosleđuju programu.. Asembler ima nedostatak da je većina stanja raštrkana po globalnim promenljivama. Sve funkcije imaju prefiks asm\_, funkcije koje se pozivaju u prvom prolazu imaju prefiks fp\_, a u drugom sp\_.