SISTEMSKI SOFTVER 13S113SS, 13E113SS

DOMAĆI ZADATAK – VAŽI OD JUNA 2017.

Napomene:
Rok za predaju je 7 dana prije ispita. Način predaje i termin odbrane će naknadno biti objavljeni.
Rešenje mora da sadrži dokumentaciju (Word ili pdf format):
1) opis problema,
2) kratak opis rešenja sa uputstvom za prevođenje i pokretanje programa,
3) nekoliko test programa za testiranje sistema (izvorni kod za svaki test program, odgovarajući predmetni program za svaki test, komande za prevođenje i rezultat emuliranja-standardni ulaz i standardni izlaz), a najmanje 3,
4) izvorni kod rešenja domaćeg zadatka (asembler i emulator),
5) izvršni kod rešenja domaćeg zadatka (asembler i emulator)

Obaveštenje o početku i načinu predaje rešenja domaćeg zadatka, kao i o tačnom terminu odbrane, biće na mailing listi predmeta ir3sp@lists.etf.rs (za SI: si3ss@lists.etf.rs). Na odbrani će biti korišćena verzija domaćeg koja je predata i neće biti dozvoljene naknadne izmene.

Nekompletna dokumentacija ili nepoštovanje pravila za prijavljivanje uzrokuje dobijanje manjeg broja poena.

Opis okruženja

Projekat se radi pod operativnim sistemom Linux na x86 arhitekturi u programskim jezicima C, C++ ili asembleru (moguća je i kombinacija). Potrebi alati su opisani u materijalima za predavanja i vežbe.

Za one koji nemaju instaliran Linux, ponuđena je instalacija u okviru virtuelne mašine VMware Player (program zahteva registraciju, ali je besplatan). Virtuelna mašina se može preuzeti (veličina arhive oko 1.3GB, na lokalnom disku potrebno bar 5GB za potrebe virtuelne mašine) sa particije "fakultet" na racunarima u laboratoriji 26 (III godina\IR3SP-SI3SS\Laboratorija\). U okviru ove virtuelne mašine već su instalirani potrebni alati.

Zadatak (25 bodova)

Napisati dvoprolazni asembler za procesor opisan u prilogu. Ulaz asemblera je tekstualni fajl u skladu sa sintaksom opisanom u nastavku. Izlaz asemblera treba da bude predmetni program zapisan u tekstualnom fajlu. Format predmetnog programa opisan je u nastavku.

Sintaksa asemblera opisana je u prilogu. Dodatni zahtevi i napomene navedeni su u nastavku:

- u jednoj liniji može biti najviše jedna komanda,

- labela može da stoji i u praznoj liniji i tada je njena vrednost jednaka adresi prve sledeće instrukcije,

- simboli mogu da se uvezu ili izvezu direktivom .global. Ukoliko je simbol definisan, izvozi se. U suprotnom se uvozi. Direktiva se može navesti bilo gdje u ualznom kodu. Primer:

.global <ime globalnog simbola>,...

- u jednoj direktivi može da se navede i više globalnih naziva koji su odvojeni zapetama,

- fajl sa izvornim kodom se završava direktivom .end. Ostatak fajla se odbacuje (ne prevodi se),

- tipovi adresiranja označeni su u skladu sa opisom u prilogu,

- pored opisanih tipova adresiranja, u instrukciji može da se pojavi i oznaka koja počinje znakom $. Tada se očekuje da se navedenoj lokaciji pristupi PC relativnim adresiranjem (u mašinskoj instrukciji se generiše registarsko indirektno adresiranje, pri čemu se koristi registar PC i odgovarajući pomeraj). Posle znaka dolar može biti navedena adresa u numeričkom obliku ili labela. Ovakav tip adresiranja može da se pojavi u svim instrukcijama koje podržavaju registarsko indirektno adresiranje sa pomerajem

- ORG direktiva može da se navede samo pred početak sekcije i tada se njome specificira početna adresa naredne sekcije (asembler ovu vrednost koristi kao početnu adresu sekcije)

- sve upotrebe simbola koje asembler može u potpunosti da razreši, asembler razreši i ne ostavlja zapise o relokaciji

- simboli u izrazima mogu da se pojave samo u operacijama sabiranja i oduzimanja (samo se dodaju ili oduzimaju od ostatka izraza), osim u slučaju kada se razlika dva izraza pojavi u zagradama čime se dobija konstanta koja dalje može da se koristi u izrazu proizvoljnog oblika

- skup pravila i napomena može biti dodatno preciziran putem liste, što će povremeno biti ugrađeno u postavku

Primer ulaznog programa:

```
.data
a: DD a
ORG 0x20
.text
LOAD R1, a    ; ucitava sadrzaj memorijske lokacije a u registar R1
LOAD R2, #a ; ucitava adresu lokacije a u registar R2
LOAD R3, $a ; ucitava sadrzaj lokacije a u registar R3 koristeci PC relativno adresiranje
x: JZ R0, x  ; apsolutni skok na lokaciju x
JZ R0, $x ; PC relativni skok na lokaciju x
.end
```

Opis formata predmetnog programa:

Predmetni program se sastoji od tabele simbola u kojoj su opisane i sekcije koje u programu postoje. Potom su navedene sekcije koje u fajlu imaju sadržaj, kao i prateće tabele relokacija. Tabela simbola počinje tekstom „#TabelaSimbola“ (bez navodnika), a zatim su u narednim redovima navedeni simboli, tako da je u svakom redu opisan jedan simbol. Red može biti zapisan po jednom od dva moguća formata, u zavisnosti od toga da li simbol predstavlja naziv sekcije ili drugi definisani simbol. Na početku reda je navedena reč koja opisuje tip reda. Za simbol koji predstavlja sekciju navedena je reč „SEG“, dok se za ostale simbole navodi „SYM“. Red koji počinje sa „SEG“ u nastavku sadrži sledeća polja odvojena sa po jednim blanko znakom: redni broj (neoznačen ceo broj zapisan decimalno), naziv simbola (jedna reč), redni broj sekcije u kojoj je simbol definisan (za sekcije isti kao i redni broj), početna adresa (neoznačen ceo broj zapisan heksadecimalno sa prefiksom 0x), veličina sekcije (neoznačen ceo broj zapisan heksadecimalno sa prefiksom 0x) i flegovi (jedna reč, gde svako slovo predstavlja jedan od uključenih flegova). Red koji počinje sa „SYM“ u nastavku sadrži sledeća polja odvojena sa po jednim blanko znakom: redni broj (neoznačen ceo broj zapisan decimalno), naziv simbola (jedna reč), redni broj sekcije u kojoj je simbol definisan (označen ceo broj zapisan

decimalno, 0 za eksterne simbole, -1 za apsolutne), vrednost simbola (neoznačen ceo broj zapisan heksadecimalno sa prefiksom 0x), i fleg koji govori da li se radi o globalnom ili lokalnom simbolu (slovo 'G' ili 'L'). Za sve simbole koji predstavljaju sekcije se smatra da su lokalni.

Neposredno posle poslednjeg simbola navode se sadržaji sekcija koje imaju sadržaj u fajlu, tako da se za svaku sekciju navede tabela relokacija za sadržaj te sekcije, pa se potom navede i sam sadržaj sekcije. Tabela sa zapisima o relokacijama počinje redom koji sadrži tekst „#rel<naziv sekcije>“. Npr, za sekciju čiji je naziv „.text.10“ tabela sa zapisima o relokacijama počinje tekstom „#rel.text.10“. U narednim redovima su navedeni zapisi o relokaciji, tako da je u svakom redu naveden jedan zapis o relokaciji. Postoje dva tipa relokacija, apsolutna i relativna. U oba slučaja zapis je naveden po sledećem formatu (navedena polja su odvojena sa po jednim blanko znakom): adresa čiji sadržaj je potrebno prepravljati (neoznačen ceo broj zapisan heksadecimalno za prefiksom 0x), tip relokacije (jedno slovo, 'A' za apsolutnu, 'R' za relativnu) i redni broj simbola (ili sekcije u slučaju upotrebe lokalnog simbola) u odnosu na koji se radi relokacija. Adresa se navodi u skladu sa početnom adresom sekcije. Npr. ako sekcija počinje na adresi 0x20 i treba prepravljati drugu duplu reč prve instrukcije u toj sekciji, adresa lokacije čiji sadržaj se prepravlja je 0x24.

Posle poslednjeg zapisa o relokacijama sledi sadržaj sekcije. Sadržaj sekcije počinje redom koji sadrži tekst „<naziv sekcije>“, nakon čega slede redovi sa sadržajem sekcije. Sadržaj sekcije se navodi kao niz bajtova tako da se u jednom redu navede 16 bajtova zapisanih u tekstualnom obliku sa po dve heksadecimalne cifre. Bajtovi su razdvojeni sa po jednim blanko znakom.

Fajl se završava redom koji sadrži „#end“. Svi redovi zaglavlja (redovi koji počinju sa #) ne smiju sadržati drugi sadržaj osim navedenog teksta (# je prvi znak u redu).

Poširenje zadatka za preostalih 15 bodova

Npraviti interpretativni emulator za arhitekturu opisanu u prilogu. Ulaz emulatora je izlaz asemblera koji je moguće učitati i pokrenuti pod uslovom da nema nedefinisanih simbola i da sekcije mogu da se učitaju na predviđene adrese bez preklapanja. Naziv predmetnog programa se zadaje kao argumenti komandne linije. Dodatni preduslov za pokretanje je i da je definisan globalni simbol START od kojeg emulator počinje izvršavanje.