SISTEMSKI SOFTVER SI3SS,
SISTEMSKO PROGRAMIRANJE IR3SP

DOMAĆI ZADATAK ZA ISPITNI ROK JUN-JUL 2016.

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), 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 bazirati na školskoj varijanti elf formata (tekstualni fajl kakav je korišćen u zadatku 9 u prezentaciji V3_Konstrukcija asemblera.ppt) i predložiti izmene u formatu u skladu sa potrebama ciljne arhitekture (nove sekcije, novi tipovi zapisa o relokacijama, dodatna polja u postojećim tipovima zapisa i sl.). Prilikom generisanja izlaznog fajla voditi se principima koje koristi GNU asembler.

Sintaksa asemblera i ostali zahtjevi:
- u jednoj liniji može biti najviše jedna komanda,
- na početku svake linije može da stoji labela koja se završava dvotačkom,

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

- simboli mogu da se izvoze tako što se na početku fajla navede direktiva:

.public <ime globalnog simbola>,...

- simboli mogu da se uvoze tako što se na početku fajla navede direktiva:

.extern <ime uveženog simbola>,...

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

- izvorni kod je podeljen u sekcije:

    - .text - sekcije koje sadrže mašinski kod,
    - .data - sekcije koje sadrže inicijalizovane podatke,
    - .bss - sekcije koje sadrže neinicijalizovane podatke,

- svaka od sekcija iza naziva može da ima i podnaziv odvojen tačkom. Osnovni naziv sekcije određuje tip sekcije, dok podnaziv omogućava pravljenje više različitih sekcija istog tipa (npr. .text.prvideo, .text.drugideo, .text su 3 različite sekcije istog tipa)

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

- pored mnemonika koji su definisani treba obezbediti direktive .char, .word, .long, .align i .skip sa istim funkcionalnostima kao u GNU asembleru,

- ostatak sintakse definisati po sopstvenom nahođenju.

Poširenje zadatka za preostalih 15 bodova

Npraviti interpretativni emulator za računar opisan u prilogu. Ulaz emulatora je izlaz asemblera, pri čemu je moguće zadati veći broj predmetnih programa koje je potrebno učitati, povezati i pokrenuti. Nazivi predmetnih programa se zadaju kao argumenti komandne linije. Emulacija je moguća samo ukoliko nakon učitavanja i povezivanja nema nerazrešenih simbola. Prilikom pokretanja emulacije, kao prvi argument komandne linije zadaje se i fajl kojim se određuje raspored sekcija prilikom punjenja programa u memoriju (skript). Format fajla za upravljanje punjenjem i postupak punjenja opisani su u nastavku. U svakom redu datoteke zapisan je naziv jedne sekcije ili naredba za dodelu vrednosti. Ponavljanje naziva sekcija nije dozvoljeno. Ukoliko se naredbama za dodelu vrednosti ne uradi drugačije, sekcije treba smestiti redosledom po kojem su navedene u skriptu, jednu za drugom. Sekcije koje nisu navedene u skriptu smestiti posle poslednje sekcije navedene u skruptu i to po abecednom redosledu. U skriptu je moguće koristiti i poseban simbol (tačka, ".") koji predstavlja tekuću adresu. Na početku rada, vrednost ovog simbola je 0. Kada se u skriptu naiđe na sekciju, sekcija se smešta počev od tekuće adresa, a tekuća adresa se uvećava za veličinu pomenute sekcije. Tekuću adresu je moguće promeniti i time promeniti raspored sekcija u memoriji. Tekuća adresa se menja tako što se simbol tačka pojavi sa leve strane dodele vrednosti ("="). Nova vrednost ne sme biti manja od prethodne (u suprotnom prijaviti grešku i prekinuti proces). Vrednost je moguće dodeliti i drugim simbolima, tako što se naziv simbola pojavi sa leve strane dodele vrednosti. Tako definisan simbol je moguće koristiti ravnopravno sa ostalim globalnim simbolima definisanim u predmetnim programima koji se pokreću (nisu dozvoljene višestruke definicije). Vrednost koja se pojavljuje sa desne strane je prost izraz u kojem učestvuju konstante i drugi do tada definisani simboli. Operacije koje su dozvoljene su sabiranje, oduzimanje i poravnanje.

Poravnanje se zapisuje poput poziva funkcije koja prihvata dva argumenta. Prvi argument je polazna adresa, dok je drugi broj delilac koji se koristi za poravnanje. Operacija ALIGN vraća najmanju adresu deljivu zadatim deliocem koja nije manja od adrese zadate prvim argumentom.