

RA domača naloga 2024/2025

Za domačo nalogo pri predmetu Računalniška arhitektura mora Peter Zmeda v zbirnem jeziku ARM implementirati prevajalnik za zbirni jezik preprostega 16-bitnega računalnika Mini MiMo. Model preprostega računalnika vsebuje 4 16-bitne registre R0, R1, R2 in R3. Ukazi in pomnilniške bsede so prav tako 16-bitne. Aritmetično logični ukazi so 2-operandni. Naslovi računalnika Mini MiMo se prično pri 0x0000.

Ker Peter ni več programiranja vas prosi za pomoč. Napišite program, ki bo za program podan v ascii nizu izvorna_koda izračunal tabelo oznak za program računalnika Mini MiMo. Vsaka nova vrstica je v nizu označena z ascii znakom '\n' oziroma LF. Direktiva .var definira 16-bitno nepredznačeno ali predznačeno število. Vsako celo število ali ukaz je lahko označen samo z eno oznako. Pred oznako so dovoljeni presledki ali tabulatorji. Oznaka se mora zaključiti z znakom ':'. Predpostavite lahko, da je program sintaktično pravilen.

Program realizirajte v večih korakih.

V **prvem** koraku prepisite niz izvorna_koda v niz izvorna_koda_pocisceno, tako da boste odstranili komentarje in odvečne presledke. Dovoljeni so enovrstični komentarji, ki se prično z znakom '@'.

V **drugem** koraku odstranite odvečne vrstice in niz izvorna_koda_pocisceno prepisite nazaj v izvorna_koda.

Za primer vzemimo program v zbirnem jeziku za Mini MiMo. Vrstice programa so označene z zaporedno številko.

```
1.)
2.)
3.)      stev1: .var 0xf123          @ komentar 1
4.) @prazna vrstica
5.)      stev2: .var  15
6.) stev3: .var 128
7.)_start:
8.) mov r1, #5 @v r1 premakni 5
9.)mov r2, #1
10.)ukaz3: add r1, #1
11.)b _start
```

Po končanem prvem in drugem koraku, mora niz izvorna_koda vsebovati počiščeno izvorno kodo.

```
1.)stev1: .var 0xf123
2.)stev2: .var 15
3.)stev3: .var 128
4.)_start: mov r1, #5
5.)mov r2, #1
6.)ukaz3: add r1, #1
7.)b _start
```

V zadnjem, **tretjem**, koraku ponovno iterirajte čez seznam izvorna_koda in izračunajte tabelo oznak.

Tabela oznak je realizirana s seznamom. Vsaka oznaka je predstavljena z asciz nizom, ki mu sledi 16 bitni naslov. Če naslov v tabeli oznak slučajno ni poravnan potem v tabelo oznak zapišite dodaten bajt z vrednostjo 0. Če je vrednost registra sodo ali liho število lahko preverite z ukazom tst.

```
tst r1, #1
bne liho @skok se izvede, če je v r1 liho število
```

Po zaključitvi programa mora tabela_oznak vsebovati sledečo vsebino. Pozorni bodite na dodatne ničle zaradi zaključenih nizov in poravnanih naslovov.

```
'stev1' 00 00 00 00 'stev2' 00 01 00 'stev3' 00 02 00 '_start' 00 00 03 00 'ukaz3' 00 05 00
```

Domače naloge ni potrebno opraviti v celoti, temveč lahko naredite le posamezne podprobleme. Za vaše rešitve je potrebno napisati poročilo na maksimalno dveh straneh in opraviti tudi ustni zagovor. V poročilu jasno prikažite rezultate delovanja vašega programa.

Naloga je sestavljena iz dveh delov. Prvi je obvezen, z drugim (neobveznim delom) lahko dobite nekaj dodatnih točk za vašo oceno LAB vaj. Obvezna naloga bo štela kot opravljena, če dokončate **eno od naslednjih možnosti**:

- sprogramirate **prvi** in **drugi** del, kjer iz izvorne kode počistite odvečne presledke, tabulatorje, komentarje in odvečne prazne vrstice;
- sprogramirate samo **tretji** del, kjer predpostavite, da je izvorna koda že počiščena in izračunate ter zapišete tabelo oznak v pomnilnik.

Neobvezna naloga bo štela kot opravljena, če dokončate eno od naslednjih možnosti:

- opravite obvezno nalogo v celoti (**prvi, drugi, tretji** del);
- napišete kateri koli drug bolj kompleksen zapleten program v zbirniku na podlagi lastne ideje;
- napišete in izvedete program na podlagi lastne ideje na modelu MiniMiMo CPE ali prispevajte na kateri koli drug način (spremenite model, ustvarite svojo lastno CPE, napišete zbirnik itd.).

```
.text
.org 0x20
izvorna_koda: .asciz "          \n\n      stev1: .var 0xf123          @ komentar 1\n @prazna vrstica \n      stev2: .var 15
\nstev3: .var 128\n_start:\n mov r1, #5 @v r1 premakni 5\nmov r2, #1\nukaz3: add r1, #1\nb _start"
izvorna_koda_pocisceno: .space 120
tabela_oznak: .space 100
.align
.global _start
_start:
        @vas program napisite tu!
_end: b _end
```