

Проектна задача бр. 1

Пишување на асемблерски програми со MIPS инструкциското множество

Правила за работа

1. Решението на проектната задача треба да биде датотека со екстензија **.asm** (предефинирана екстензија за асемблерски код во симулаторот MARS) која ќе биде прикачена на поставениот линк на курсот.
2. Датотеката треба да биде **именувана според вашиот индекс** (пример 123456.asm).
3. Кодот треба задолжително да содржи **коментари** напишани **со латинско писмо**.
4. Првиот ред на кодот треба да биде коментар кој ќе го содржи вашиот **индекс, презиме и име** со латинско писмо.
5. **Доколку не биде исполнет барем еден од претходните услови, проектната задача нема да биде прегледана.**
6. **Студентите кои ќе имаат макар делумни плагијати на кодот ќе бидат казнети со 0 поени и забрана за полагање во јунскиот испитен рок. Казната ќе биде применета за сите студенти кои имаат слични кодови, без да се води сметка кој од кого го преземал кодот.**
7. Кодот кој нема успешно да се компајлира ќе биде оценет со 0 поени.
8. Студентите кои нема да успеат да ја решат целата задача, но ќе понудат точно делумно решение, ќе добијат соодветен број на поени.
9. Проектната задача не е задолжителна, но има удел во вкупната оцена.
10. Проектната задача се изработува во симулаторот MARS.
11. Дозволено е користење на псевдо-инструкции.

Задача

Да се напише асемблерска програма во MIPS која за дадени два вектори, како низи од цели броеви, ќе ги пресмета нивниот збир и нивниот скаларен производ. Збир на два вектори е вектор чији што елементи на дадени позиции се збир на елементите од двата вектори на истите позиции. Скаларен производ на два вектори е скалар кој се добива како збир од производите на посебните елементи на двата вектори.

Пример:

$$v1 = (1; 3; -7)$$

$$v2 = (4; -2; -1)$$

$$\text{збир на вектори} \rightarrow v1 + v2 = [1 + 4; 3 + (-2); (-7) + (-1)] = (5; 1; -8)$$

$$\text{скаларен производ} \rightarrow v1 * v2 = 1 * 4 + 3 * (-2) + (-7) * (-1) = 4 - 6 + 7 = 5$$

Векторите (низите) се читаат од стандарден влез. На стандарден влез најпрво се задава почетната мемориска адреса за зачувување на збирот, потоа следува големината на векторите (двата вектори имаат иста големина), па следно се задаваат двата вектори, односно се читаат нивните елементи. Влезните низи се состојат од најмногу 20 цели броеви. Низите треба да се зачуваат во меморијата за што е потребно да се резервира простор од по 20 збора за двете низи (*помош*: со наредбата `.space n` се алоцираат n последователни *бајти* во меморијата).

Програмата треба да искористи процедура за пресметување на скаларен производ и збир на вектори. Во процедурата се пресметуваат скаларниот производ и збирот на векторите, така што збирот на векторите директно се зачувува во меморијата на очекуваната адреса.

Програмата на стандарден излез треба го да испечати скаларниот производ. Збирот на двата вектори треба да остане запишан како низа на мемориска локација која е зададена на стандарден влез. Дополнително, во регистарот `$s1` зачувајте ја вредноста на последните 16 бита од вредноста на вашиот индекс (користете маска за пресметување на вредноста). Бројот на индекс го внесувате преку податочниот сегмент во програмата и во кодот ги пресметувате последните 16 бита.

Забелешки:

- Должината на низите ќе биде зададена на стандарден влез, пред да се внесат елементите.
- Програмата на стандарден излез **не треба да печати** текст од типот „внесете низи“, „внеси големина на низите“ или „скаларниот производ е 5“ и сл.
- Пресметката на скаларниот производ и збирот на вектори се врши во процедура. Пресметките по модул со број на индекс, како и влезот и излезот не треба да бидат во процедури.
- Решенијата кои нема да користат процедура за соодветниот дел ќе бидат оценети со помал број на поени.
- Проверете ја исправноста на вашето решение и со други низи бидејќи евалуацијата на кодот ќе се изврши со различни низи од примерот.

Пример (за број на индекс 123456):

Влез: 268501280 3 1 3 -7 4 -2 -1

мемориска адреса за зачувување на векторот на збирот	големина на векторите	вектор 1	вектор 2
268501280	3	1, 3, -7	4, -2, -1

Излез:

5

Дополнително, збирниот вектор (5, 1, -8) е зачуван во меморијата почнувајќи од мемориска локација 268501280, а вредноста во регистарот \$s1 е 57920 (вредност на последни 16 бита од 123456).