

ЗАДАТАК

У оквиру пројектног задатка потребно је реализовати МАНН преводац који преводи програме са вишег асемблерског језика на основни MIPS 32bit асемблерски језик. Преводац треба да подржава детекцију лексичких, синтаксних и семантичких грешака као и генерисање одговарајућих извештаја о евентуалним грешкама. Излаз из преводиоца треба да садржи коректан асемблерски код који је могуће извршавати на MIPS 32bit архитектури (симулатору).

Уз пројекат је неопходно испоручити и пропратну документацију која садржи:

- Анализу проблема
- Концепт решења
- Програмско решење
- Верификацију

МАНН – ПРЕВОДИЛАЦ

МАНН (Мипс Асемблер Високог Нивоа) је алат који преводи програм написан на вишем MIPS 32bit асемблерском језику на основни асемблерски језик. Виши MIPS 32bit асемблерски језик служи лакшем асемблерском програмирању јер уводи концепт **регистарске** променљиве. Регистарске променљиве омогућавају програмерима да приликом писања инструкција користе променљиве уместо правих ресурса. Ово знатно олакшава програмирање јер програмер не мора да води рачуна о коришћеним регистрима и њиховом садржају. Ради једноставности пројекта у инструкцијама не смеју да се користе прави ресурсе већ искључиво променљиве.

МАНН језик подржава 10 MIPS инструкција, а то су:

add – (addition) сабирање

addi – (addition immediate) сабирање са константом

b – (unconditional branch) безусловни скок

bltz – (branch on less than zero) скок ако је регистар мањи од нуле

la – (load address) учитавање адресе у регистар

li – (load immediate) учитавање константе у регистар

lw – (load word) читавање једне меморијске речи

nop – (no operation) инструкција без операције

sub – (subtraction) одузимање

sw – (store word) упис једне меморијске речи

Осим ових 10 инструкција реализовати још барем ТРИ произвољне инструкције из документа *MIPS-instructions.pdf*, од којих барем једна није из аритметичко-логичког скупа инструкција. Наравно, потребно је проширити синтаксу језика која је наведена у наставку.

Синтакса MABH језика описана је граматиком:

$Q \rightarrow S ; L$	$S \rightarrow _mem \ mid \ num$	$L \rightarrow eof$	$E \rightarrow add \ rid, \ rid, \ rid$
	$S \rightarrow _reg \ rid$	$L \rightarrow Q$	$E \rightarrow addi \ rid, \ rid, \ num$
	$S \rightarrow _func \ id$		$E \rightarrow sub \ rid, \ rid, \ rid$
	$S \rightarrow id : E$		$E \rightarrow la \ rid, \ mid$
	$S \rightarrow E$		$E \rightarrow lw \ rid, \ num(rid)$
			$E \rightarrow li \ rid, \ num$
			$E \rightarrow sw \ rid, \ num(rid)$
			$E \rightarrow b \ id$
			$E \rightarrow bltz \ rid, \ id$
			$E \rightarrow nop$

Терминални симболи MABH језика су:

;;, ()

`_mem`, `_reg`, `_func`, `num` `id`, `rid`, `mid`, `eof`, `add`, `addi`, `sub`, `la`, `lw`, `li`, `sw`, `b`, `bltz`, `nop`

Овде се напомиње на који начин се врши декларисање функција, меморијских и регистарских променљивих:

- Декларација функције:

`_func funcName`

funcName – мора почети словом, у наставку може бити било који низ слова и бројева.

Пример:

```
_func foo
```

- Декларација меморијске променљиве:

`_mem varName value`

varName – мора почети малим словом *m* у наставку може бити било који број.

Пример:

```
_mem m4 55  
_mem m78 12
```

- Декларација регистарске променљиве:

`_reg varName`

varName – мора почети малим словом *r* у наставку може бити било који број.

Пример:

```
_reg r1  
_reg r9
```

За реализацију пројекта могу се користити приложене датотеке (директоријум *src*) за лексичку анализу као и предлог реализације променљивих и инструкција (*IR.h*). У прилогу се налазе и примери улазних програма (директоријум *examples*). За почетак користити датотеку: *simple.mavn*, а накнадно целокупно решење проверити са датотеком: *multiply.mavn*.

У наставку је дат пример полазног програма и њему одговарајућег асемблерског излаза:

Програм на мавн програмском језику:

```
_mem m1 6;  
_mem m2 5;  
_reg r1;  
_reg r2;  
_reg r3;  
_reg r4;  
_reg r5;  
_func main;  
    la            r4,m1;  
    lw            r1, 0(r4);  
    la            r5, m2;  
    lw            r2, 0(r5);  
    add           r3, r1, r2;
```

Програм преведен на MIPS 32bit асемблерски језик:

```
.globl main  
  
.data  
m1:          .word 6  
m2:          .word 5  
  
.text  
main:  
    la         $t0, m1  
    lw         $t1, 0($t0)  
    la         $t0, m2  
    lw         $t2, 0($t0)  
    add        $t0, $t1, $t2
```

Задаци

Задатак МАНН преводиоца је да уčitану улазну датотеку писану на МАНН језику преведе на MIPS 32bit асемблер. Ограничити се на једну улазну датотеку. Користити екстензију “.man” за улазну датотеку која садржи програм на МАНН језику.

Преводилац треба да приликом преводјења:

1. Додели ресурсе за регистарске променљиве – ограничити се на 4 регистра: t0, t1, t2 и t3 из MIPS архитектуре
2. Све меморијске променљиве генерише у секцију за податке - .data водећи рачуна о синтакси асемблерског језика
3. Све инструкције смести у програмску секцију - .text
4. Име функције генерише као глобални симбол - .globl и као лабелу на прву њену инструкцију
5. Генерише излазну датотеку са екстензијом “.s” која садржи преведен и коректан MIPS 32bit асемблерски језик полазног програма

НАПОМЕНЕ:

1. Приликом преводјења омогућити детекцију и реаговање на лексичке, синтаксне и семантичке грешке.
2. Уколико се полазни програм успешно преведе на MIPS 32bit асемблерски језик, извршавање програма је могуће проверити коришћењем *QtSpim* симулатора који је доступан на [линку](#).