Întrebări și Răspunsuri – Compilație

Am strâns aici toate întrebările care mi-au fost puse, până acum, legat de tema voastră și ce soluții sau „chichițe” am remarcat uitându-mă peste proiectul vostru. Înainte să începem (citiți tot ce e cu roșu, e pt toată lumea):

**WARNING**: Știu că am mai spus-o și o să mă repet până mor, dar **EU NU VĂ AJUT LA MIPS!!!** (teoretic, vorbind, pt. că laborantul vostru m-a rugat să nu intervin peste el și să nu facem MIPS la tutoriat, dar… here I am. Deci, shhhhhhhh...)

* Pt problema de **nota 7**, cum să rețin matricea și să o manipulez în MIPS? 🡪 Folosim tot vectori, dar manipulăm indicii ca să se potrivească. Spre exemplu:

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |
| 9 | 10 | 11 |
| 12 | 13 | 14 |

Elementul 5 cu i(linia)=1 și j(coloana)=2 corespunde elementului i\*nr\_coloane+j și tot așa.

**Pt Problema de nota 10**:

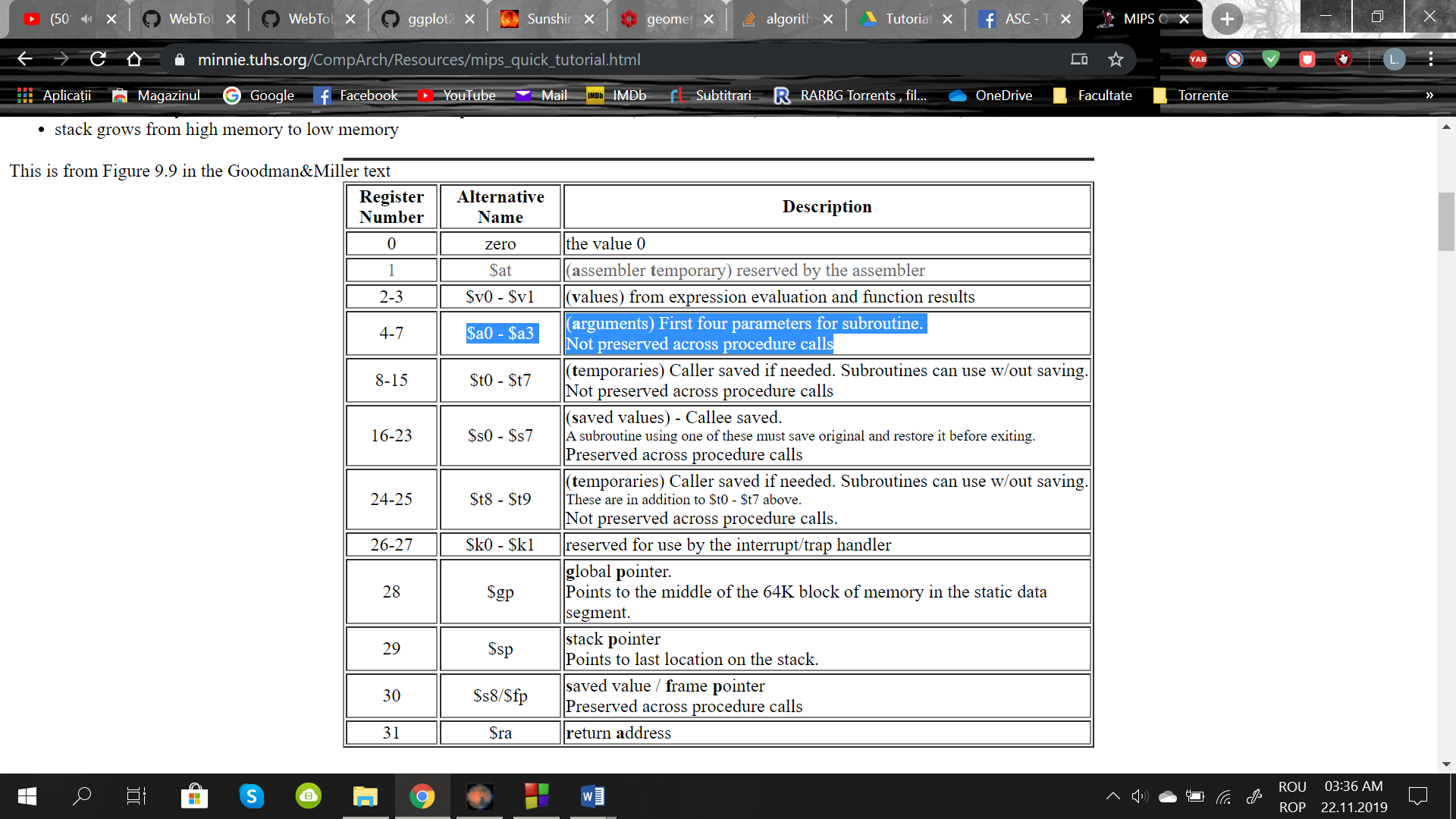
Sfaturi Generale:

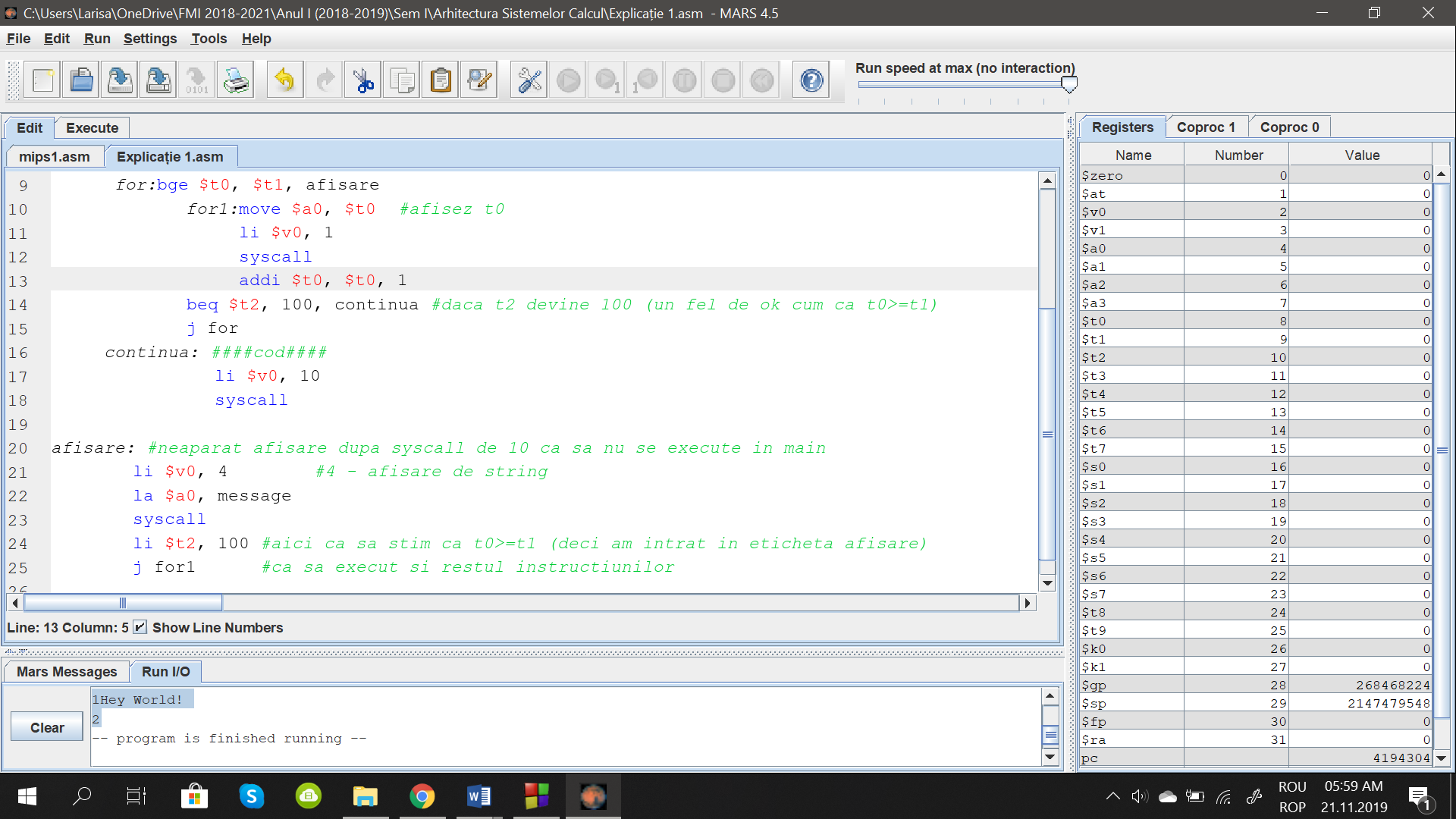
* Mai întâi faceți problema în C++, cât mai optim, clar, concis și fără să recurgeți la structuri de date foarte complicate. Spre exemplu, cineva verifica cu 2 for-uri ca să vadă dacă sunt 2 de 1 și atunci nr respectiv nu era generator. Așa ne complicăm. Mai ușor, în timp ce calculăm vectorul de generator, putem să avem o condiție în C++ de genul *if(generator[j]==1 && j!=p-1) ok=false;* Și exemplele pot continua (idea e să nu vă complicați cu foarte multe etichete și condiții și blah).
* După ce aveți programul în C++, vă puteți apuca de convertirea în MIPS. Recomandare: spargeți problema asta mare, în probleme atomice, de sine stătătoare, și faceți verificări în continuu ca să vedeți dacă aveți greșeli din timp (după ce se adună mult cod, nu veți mai știi pe unde se găsește și va dura mult timp ca să o identificați). Recomandare 2: folosiți MARS (În **NICIUN CAZ PCSPIM**, orice altceva) pentru că puteți face debug pe pași și puteți vedea și cum se modifică regiștrii, dar și memoria de pe stivă, adică vedeți live unde se schimbă valorile și unde apare o posibilă greșeală (vedeți că vă scrie în MARS unde se face afișarea, numele erorii și puteți să o căutați pe net).
* Fiți odihniți și alocați-vă câteva ore în fața calculatorului, fără să fiți deranjați. Este o problemă muncitorească; nu e imposibilă, dar durează foarte mult să o implementați. Volumul de cod e cam de 3 ori mai mare decât în C++.
* Puteți să îmi dați mesaj în privat, dar, vă rog, veniți cu întrebări clare, nu de genul „De ce nu merge programul?” pentru că durează foarte mult să mă uit pe codul tutoror și aș aprecia dacă mi-ați spune „Am eroare pe linia X. De ce?”. Adică descrieți-mi puțin ce încercați să faceți și de ce nu merge. Înțelegeți voi… pt că nu am timp să mă ocup de voi toți, altfel.

Întrebări:

1. Cum afișez o valoare int în MARS? 🡪 Nu rețineți adresa de memorie a int-ului în $a0, ci valoarea efectivă a int-ului. Apoi $v0=1 și syscall.
2. Pot folosi regiștrii $a0, $a1 etc dacă rămân fără $t? 🡪 Teoretic, poți, dar convenția spune că nu prea ai voie (pt că sunt folosiți de subrutine ca argumente, aka în ei punem argumentele funcțiilor… cum ar fi în C++ *int ceva(int a, int b)* și a și b sunt, de fapt, $a0, $a1). La nevoie, aia e, acum nu te-o omorî. Ai mai jos explicații pentru fiecare registru:

<https://minnie.tuhs.org/CompArch/Resources/mips_quick_tutorial.html>



1. Am o etichetă de genul *for: bge …, …, afiș*. Pot să fac cumva ca după ce se execută eticheta afiș, să mă întorc înapoi în for și să termine execuția? 🡪 Da. Este cam peste mână și nu știu de ce v-ar trebui așa ceva, dar vă arăt un exemplu rudimentar.

Cod:

.data

message: .asciiz "Hey World! \n"

.text

main:

li $t0, 1

li $t1, 2

li $t2, 0

for:bge $t0, $t1, afisare

for1:move $a0, $t0 #afisez t0

li $v0, 1

syscall

addi $t0, $t0, 1

beq $t2, 100, continua #daca t2 devine 100 (un fel de ok cum ca t0>=t1)

j for

continua: ####cod####

li $v0, 10

syscall

afisare: #neaparat afisare dupa syscall de 10 ca sa nu se execute in main

li $v0, 4 #4 - afisare de string

la $a0, message

syscall

li $t2, 100 #aici ca sa stim ca t0>=t1 (deci am intrat in eticheta afisare)

j for1 #ca sa execut si restul instructiunilor

#######Uite și ce afișează programul####### 1Hey World!2

1. Primim următoarea eroare: *Runtime exception at 0x0040007c: store address not aligned on word boundary 0x1001001*, dată de instrucțiunea: sw $t0, ($s0) #gen[0]=1. Iar codul nostru are declarat ca date de intrare (în această ordine):

*.data*

*p: .space 4*

*mesaj: .asciiz "Nu e Prim"*

*gen: .space 100000*

De fapt, ce se întâmplă? Noi lucrăm cu biții în memorie. Int ocupă 4 biți, char ocupă 1 bit etc… În memorie noi trebuie să așezăm structurile de date, astfel încât poziția pe care o ocupă în memorie să dividă nr de biți pe care structura respectivă o are. Acest lucru este făcut automat de limbaje de programare avansate (C++), dar în MIPS trebuie să o facem no, manual. Pe exemplul nostru. P este .space 4, adică ocupă 4 biți. Mesaj este un string și va ocupa 10 biți. 10+4=14. Gen, noi vrem să îl inițializăm cu un word (sw $t0, ($s0) #gen[0]=1) și word ocupă 4 biți (este un int), deci el tb să stea în memorie pe o poziție divizibilă cu 4 (0, 4, 8, 12, 16…). Noi suntem la bitul 15, deci gen nu se poate inițializa cu un word. Cum rezolvăm problema? Declarăm gen înaintea lui p (p și mesaj sunt ambele char-uri, deci nu contează pe ce poziție se află în memorie) și gen va începe la adresa 0 (0 divide 4 și am scăpat de eroare). **Pentru viața voastră de programatori**: **este important să declarați variabilele de la cele care ocupă cel mai mare nr de biți, la cele care ocupă mai puțini**. De ce? Deși C++ vă va poziționa corect la adresa din memorie, voi efectiv mâncați din memoria sistemului (C++ va umple acei biți liberi ca să poată să ajungă la poziția care se divide) și vă treziți cu programe cu nush ce dimensiune inumană.

1. Lucrul cu vectori 🡪 aveți mai multe posibilități: <http://courses.cs.vt.edu/~cs2505/summer2011/Notes/pdf/T23.MIPSArrays.pdf>
2. Cum se afișează un string în MIPS? → Trebuie reținută adresa string-ului în $a0 (Ex: la $a0, mesaj) și $v0=4.
3. De ce se afișează un caracter suplimentar la finalul string-ului, după ce fac codarea? → Pt că voi transformați și ultimul caracter al string-ului (acel 0), deci trebuie să luați asta în considerare când faceți codarea.