

ΕΡΓΑΣΤΗΡΙΟ ΑΡΧΙΤΕΚΤΟΝΙΚΗ Η/Υ

3Η ΕΡΓΑΣΙΑ ΕΡΓΑΣΤΗΡΙΟΥ

**Arduino +
x32 MIPS ASSEMBLY**

Ορφέας- Άγγελος Νικολάου

ΑΜ: 2792

E-MAIL: int02792@uoi.gr

Άρτα, 2023 / 11 / 11

Table of contents

1η Ασκηση – Arduino.....	3
Περίληψη.....	3
Υλοποίηση.....	3
Διάγραμμα.....	3
Κώδικας.....	3
Screenshots.....	6
2η Ασκηση – MIPS (μέρος 1o).....	7
Περίληψη.....	7
Υλοποίηση.....	7
Ερώτηση 1.....	7
Ερώτηση 2.....	8
Int.....	8
Float.....	8
String.....	9
Ερώτηση 3.....	9
Ερώτηση 4.....	9
2η Ασκηση – MIPS (μέρος 2o).....	10
Περίληψη.....	10
Υλοποίηση.....	10
Κώδικας.....	10
Screenshots.....	10
2η Ασκηση – MIPS (μέρος 3o).....	11
Περίληψη.....	11
Υλοποίηση.....	11
Κώδικας.....	11
Screenshots.....	11

1η Έσκηση - Arduino

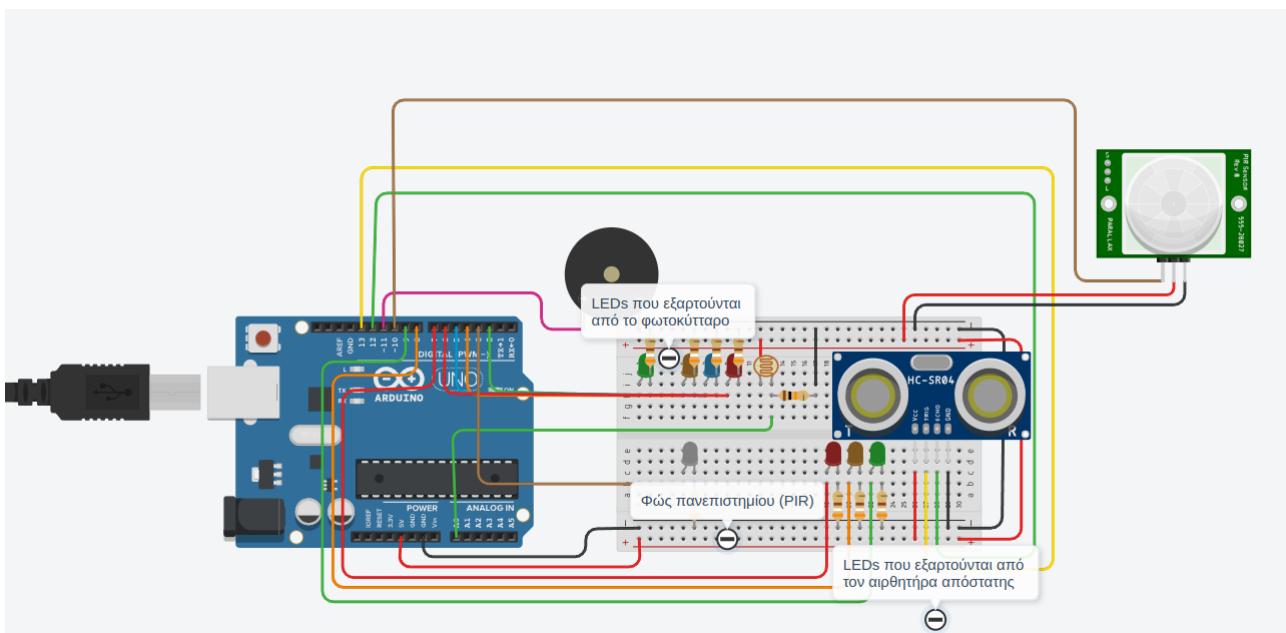
Περίληψη

Σε αυτή την άσκηση ζητιέται η υλοποίηση ένα μέρος μιας έξυπνης πόλης με:

- Αισθητήρες ανίχνευσης φωτός
 - Αισθητήρα απόστασης μαζί με σύστημα ήχου για warning
 - Αισθητήρα PIR.

Υλοποίηση

Διάγραμμα



(Είχα κάτσει για μισή ώρα με το analog LED στο A5 και δεν δούλευε,
τελικά είναι μόνο για input αυτά...)

Κώδικας

GitHub mirror

Pastebin mirror Password: 12341

Φυσικά υπάρχει και μέσα στον φάκελο Askhsh_1.

```

12 #define LED1 2
11 #define LED3 4
10 #define LED4 5
9 #define LED5 6
8 #define SONIC_TRGR 13
7 #define SONIC_ECHO 12
6 #define RED 7
5 #define ORANGE 8
4 #define GREEN 9
3 #define BUZZER 11
2 #define PIR 10
1 #define PIR_LED 3
13
1 /* START TIME */
2 struct time{
3     size_t h;
4     size_t m;
5     size_t s;
6 };
7 struct time set_time(void); // set time to struct time
8 size_t get_hours(struct time const); // return h field of struct time
9 size_t get_minutes(struct time const); // return m field of struct time
10 size_t get_seconds(struct time const); // return s field of struct time
11 bool is_valid(struct time const); // is valid for distance sensor to enable
12 /* END TIME */
13
14 void toggle_5_leds(int const); // Usage: 0/LOW for off, 1/HIGH for on
15 void handle_photoresistor(void); // enables 5 leds if val>930, else shuts them off
16 int activate_sonic_sensor(size_t const); //parameter is activation in um
17 void handle_distance(int const);
18
19 struct time current_time = {0}; // no reason to = 0 but good practice
20
21 void handle_pir(void);
22
23 #define ENABLE_PHOTORESISTOR 1
24 #define ENABLE_ULTRASONIC 1
25 #define ENABLE_PIR 1
26
27 void setup()
28 {
29     Serial.begin(9600);
30     /* group of 5 leds */
31     pinMode(LED1, OUTPUT);
32     pinMode(LED3, OUTPUT);
33     pinMode(LED4, OUTPUT);

```

```

22     pinMode(LED1, OUTPUT);
21     pinMode(LED3, OUTPUT);
20     pinMode(LED4, OUTPUT);
19     pinMode(LED5, OUTPUT);
18     /* init sonic sensor */
17     pinMode(SONIC_TRGR, OUTPUT);
16     pinMode(SONIC_ECHO, INPUT);
15     /* init sonic sensor leds */
14     pinMode(GREEN, OUTPUT);
13     pinMode(ORANGE, OUTPUT);
12     pinMode(RED, OUTPUT);
11     /* set sonic sensor leds to low juust in case */
10     digitalWrite(GREEN, 0);
9     digitalWrite(ORANGE, 0);
8     digitalWrite(RED, 0);
7     /* set group of 5 leds to low */
6     toggle_5_leds(0);
5
4     digitalWrite(SONIC_TRGR, 0);
3     #if ENABLE_ULTRASONIC
2         current_time = set_time(); // για να μην ρωταιει καθε φορα την ωρα
1     #endif //ENABLE_ULTRASONIC
66     pinMode(BUZZER, OUTPUT);
1
2     pinMode(PIR, INPUT);
3     pinMode(PIR_LED, OUTPUT);
4     analogWrite(PIR_LED, 0);
5
6 }
7
8 void loop(){
9     // photoresistor
10    #if ENABLE_PHOTORESISTOR
11        handle_photoresistor();
12    #endif //ENABLE_PHOTORESISTOR
13
14    // ultrasonic + piezo
15    #if ENABLE_ULTRASONIC
16        if (is_valid(current_time)){ // is valid when [23:00:00,06:00:00]
17            int dist = activate_sonic_sensor(10); // 10 microseconds pulse
18            handle_distance(dist, false);
19            // true flag είναι αρκετά ενωχλητικό διότι εμφανίζει μύνημα
20            // κάθε φορα που ολλάζει η θέση του, και δεν είναι καν
21            // ο σκοπός του sensor αυτού
22            //handle_distance(dist, true);
23        }

```

```

22     // κάθε φορα που ολλάζει η θέση του, και δεν είναι καν
21     // ο σκοπός του sensor αυτού
20     //handle_distance(dist, true);
19 }
18 #endif //ENABLE_ULTRASONIC
17
16 // pir
15 #if ENABLE_PIR
14     handle_pir();
13 #endif //ENABLE_PIR
12 }

11 struct time set_time(void){
10     struct time ret;
9     int x; // input
8
7     // set hours + range check
6     do{
5         Serial.println("Enter hours (0-23): ");
4         while(Serial.available() == 0) { delay(300); } // infinite loop while nothing
3         x = Serial.parseInt();
2     }while(x < 0 || x > 23);
1     ret.h = x;
108
1     // set minutes + range check
2     do{
3         Serial.println("Enter minutes (0-59): ");
4         while(Serial.available() == 0) {delay(300);} // infinite loop while nothing
5         x = Serial.parseInt();
6     }while(x < 0 || x > 59);
7     ret.m = x;
8
9     // set seconds + range check
10    do{
11         Serial.println("Enter seconds (0-59): ");
12         while(Serial.available() == 0) {delay(300);} // infinite loop while nothing
13         x = Serial.parseInt();
14     }while(x < 0 || x > 59);
15     ret.s = x;
16
17     char message[100];
18     // tell user current time
19     sprintf(message,
20         "Time was set to: %02u:%02u:%02u\n",
21         ret.h, ret.m, ret.s);
22     Serial.print(message);
23

```

```

22         ret.h, ret.m, ret.s);
21     Serial.print(message);
20
19 // tell user if time is valid or not for proximity sensor
18 if (is_valid(ret)){
17     Serial.println("Time is in the 23:00:00--06:00:00 range.");
16     Serial.println("Activating proximity seensor.");
15 }
14 else{
13     Serial.println("Time is not in the 23:00:00--06:00:00 range.");
12     Serial.println("Proximity sensor will NOT be activated.");
11 }
10     return ret;
9 }
8 size_t get_hours(struct time const t){ return t.h; }
7 size_t get_minutes(struct time const t){ return t.m; }
6 size_t get_seconds(struct time const t) { return t.s; }
5 bool is_valid(struct time const t){
4
3     if (get_hours(t) < 23 && get_hours(t) > 6) { return false; }
2     if (get_hours(t) == 6 && (get_minutes(t) > 0 || get_seconds(t) >0)) { return false; }
1     return true;
151
1 }
2 void toggle_5_leds(int const t){
3     // LEDx = pin number + 1
4     // τύπω που στο σκεύωμα η συνάρτηση θα μπορούσε να μην
5     // δέχεται τιμοτα και να κάνει όντως toggle κάθε φορά που
6     // ολλάζεται
7     digitalWrite(LED1, t);
8     digitalWrite(LED3, t);
9     digitalWrite(LED4, t);
10    digitalWrite(LED5, t);
11 }
12 void handle_photoresistor(void){
13     int val = analogRead(A0);
14     static bool on = false; //για να μην εκτυπώνει κάθε φορά
15         //τα μύνημα
16     if (val < 930 && !on){
17         // if it's dark and they're off, turn them on
18         toggle_5_leds(1);
19         Serial.println("Low light levels detected, powering on leds");
20         on = true;
21     }
22     else if (val >=930 && on){
23         // if it's bright and they're on, turn them off

```

```

22 }
21 else if (val >=930 && on){
20 // if it's bright and they're on, turn them off
19 toggle_5_leds(0);
18 Serial.println("High light levels detected, powering off leds");
17 on = false;
16 }
15 }
14 int activate_sonic_sensor(size_t const d){
13
12 digitalWrite(SONIC_TRGR, 1);
11 delayMicroseconds(d);
10 digitalWrite(SONIC_TRGR, 0);
9 double duration = pulseIn(SONIC_ECHO, 1);
8 // scary magic
7 int distance = duration * 0.034 / 2;
6 return distance;
5 }
4 void handle_distance(int const d, bool const flag){
3
2 // previous --> για να μην εκτυπώνει κάθε φορά μόνημα
1 // select--> για να μην λειτουργήσει άλλαξε κάθε φορά το LED
194 // tolerance--> και να μην άλλαξε η τιμή θα έμπινε μέσο
1 static int previous = 0;
2 int const tolerance = 1;
3 static int select[3] = {0};
4 static int freqKHz = 0;
5 if (abs(d-previous) > tolerance){
6 previous = d;
7 // Annoying message but may be useful
8 if (flag) {Serial.println("Proximity sensor message: object changed place");}
9 // Dangerous distance
10 if (d<= 25 && select[0] == 0){
11 digitalWrite(GREEN, 0);
12 digitalWrite(ORANGE, 0);
13 digitalWrite(RED, 1);
14 Serial.println("Proximity sensor message: distance <=25");
15 Serial.println("Proximity sensor message: Light: RED");
16 freqKHz = 4000;
17 select[0]=1;select[1]=0;select[2]=0;
18 }
19 // Warning distance
20 if (d >= 25 && d <= 50 && select[1] == 0){
21 digitalWrite(GREEN, 0);
22 digitalWrite(ORANGE, 1);
23 digitalWrite(RED, 0);

```

```

45 if (d > 50 && select[2] == 0){
44 digitalWrite(GREEN, 1);
43 digitalWrite(ORANGE, 0);
42 digitalWrite(RED, 0);
41 Serial.println("Proximity sensor message: 50 < distance");
40 Serial.println("Proximity sensor message: Light: GREEN");
39 freqKHz = 1000;
38 select[0]=0;select[1]=0;select[2]=1;
37 }
36 }
35 tone(BUZZER, freqKHz);
34
33 }
32
31 void handle_pir(void){
30 // προνομεύειν καταστάση για να μην εμφανίζεται μυνημάτων καθε φορά
29 static int PIR_STATE = 0;
28 static int val = 0;
27 static int brightness = 0;
26 val = digitalRead(PIR);
25
24 // if something moved
23 if (val == 1){
22 //switch on lights
21 brightness = 255;
20 analogWrite(PIR_LED, brightness);
19 // if was off and now on, show msg
18 if (PIR_STATE == 0) {
17 Serial.println("PIR message: Motion detected!");
16 PIR_STATE = 1;
15 }
14 }
13 else {
12 // slowly turn down the light
11 brightness -= 5;
10 // in case negative values
9 if (brightness < 0) { brightness = 0;};
8 analogWrite(PIR_LED, brightness);
7 // if it was on and now off, show msg
6 if (PIR_STATE == 1){
5 Serial.println("PIR message: Motion stopped!");
4 PIR_STATE = 0;
3 }
2 }
1 delay(100);
269

```

Αντί για ήχο με πιο γρήγορο ρυθμό, το έβαλα με υψηλότερη συχνότητα όσο αυξάνεται ο κίνδυνος, αφού ακούγεται πιο καθαρά η διαφορά (κατά την γνώμη μου).

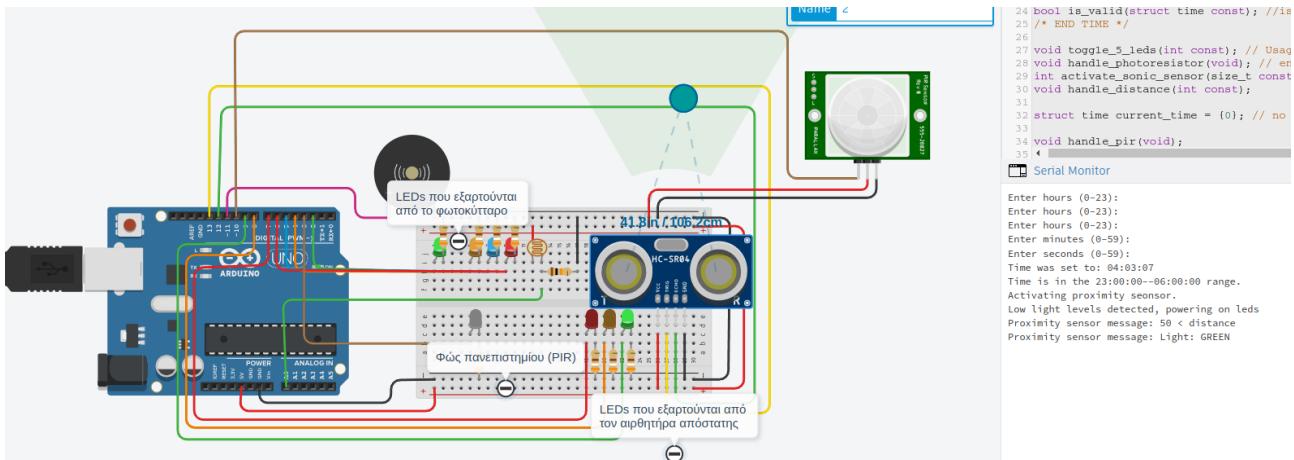
ΠΡΟΣΟΧΗ το % ήχου στο σύστημα πρέπει να είναι υπερβολικά χαμηλό, αλλιώς σπάει τα αυτιά και δεν βρήκα άλλον τρόπο να το χαμηλώσω.

Στις γραμμές 36, 37, 38 έχει defines για disable/enable features (το να εισάγω ώρα κάθε φορά που ήθελα να δοκιμάσω κάτι είχε καταντήσει κουραστικό, και είπα να το βάλω για όλα αφού το έβαλα για τον ήχο).

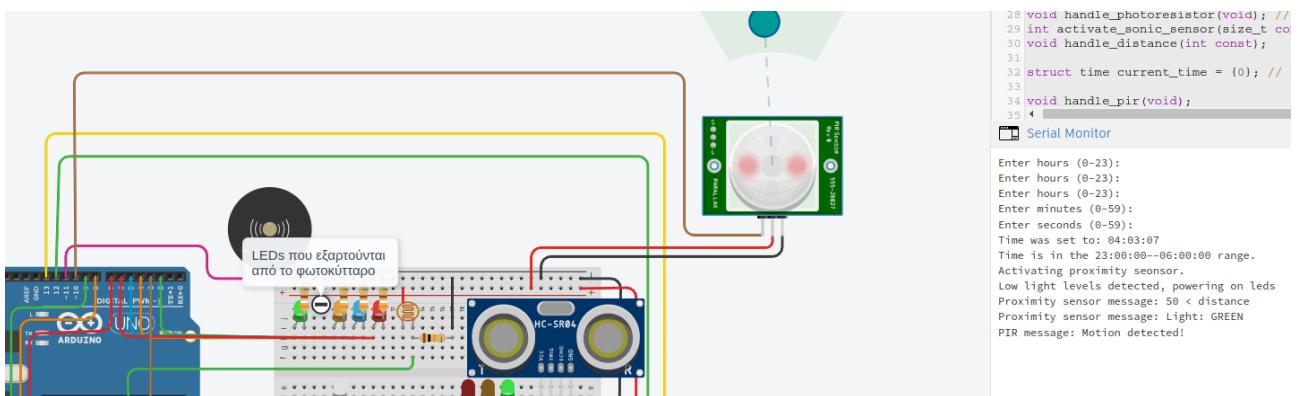
Το LED πανεπιστημίου το έκανα αναλογικό διότι μου ήταν λογικό να το βάλω έτσι, αν και δεν είναι εύκολο να το βγάλω σε screenshots.

Screenshots

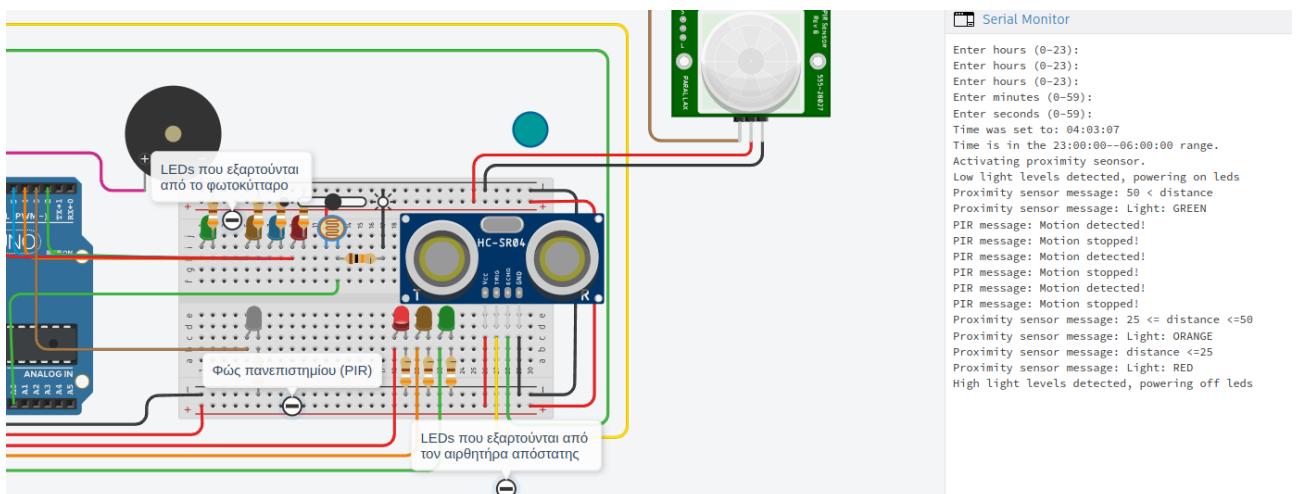
(Όλα #define ENABLE_X 1)



Έγκυρη ώρα



PIR detection.



Κίνδυνος και υψηλά επίπεδα φωτεινότητας.

Δεν βάζω μη-έγκυρη ώρα επειδή είναι βαρετό, απλά το ultrasonic και piezo δεν λειτουργούν.

2η Άσκηση – MIPS (μέρος 1ο)

Περίληψη

Απαντήσεις ερωτήσεων.

Υλοποίηση

Ερώτηση 1

Το εξής πρόγραμμα είναι ο ελάχιστος απαιτούμενος κώδικας που χρειάζεται για να εμφανίσουμε ένα μήνυμα προς την χρήστη.

Σε περίπτωση που έχουμε ένα πλήρης πρόγραμμα, το μόνο που χρειάζεται είναι:

1. Πεδία στο .data τύπου .asciiz (είχα δοκιμάσει κάποτε με .ascii και στο τέλος των χαρακτήρα \0, δυστυχώς δεν δούλεψε).
2. Μέσα στην main να φορτώσουμε την διεύθυνση του αλφαριθμητικού στο \$a0 με την εντολή la (load adress) όπου παίρνει &str[0].
3. \$v0 ορίζει τι syscall καλούμε, αφού θέλουμε printf("%s") κάνουμε li (load immediate) το syscall 4.

(Βήματα 2 και 3 μπορούμε με όποια σειρά θέλουμε).

4. syscall

```
8 .data
7
6  onoma_alfarithmitikou: .asciiz "this is an example"
5
4 .text
3 .globl main
2 main:
1
    la $a0, onoma_alfarithmitikou # $a0 parametros mias syscall (an xreiazetai)      grammh 9
1                                # ektos an h syscall exei sxesh me floats          grammh 10
2
3  li $v0, 4                   # syscall --> print string me arguments thn $a0      grammh 11
4  syscall                      # kaloume thn syscall 4                           grammh 12
5
6
7 #EXIT
8 li $v0, 10
9 syscall
```

ΠΗΓΗ

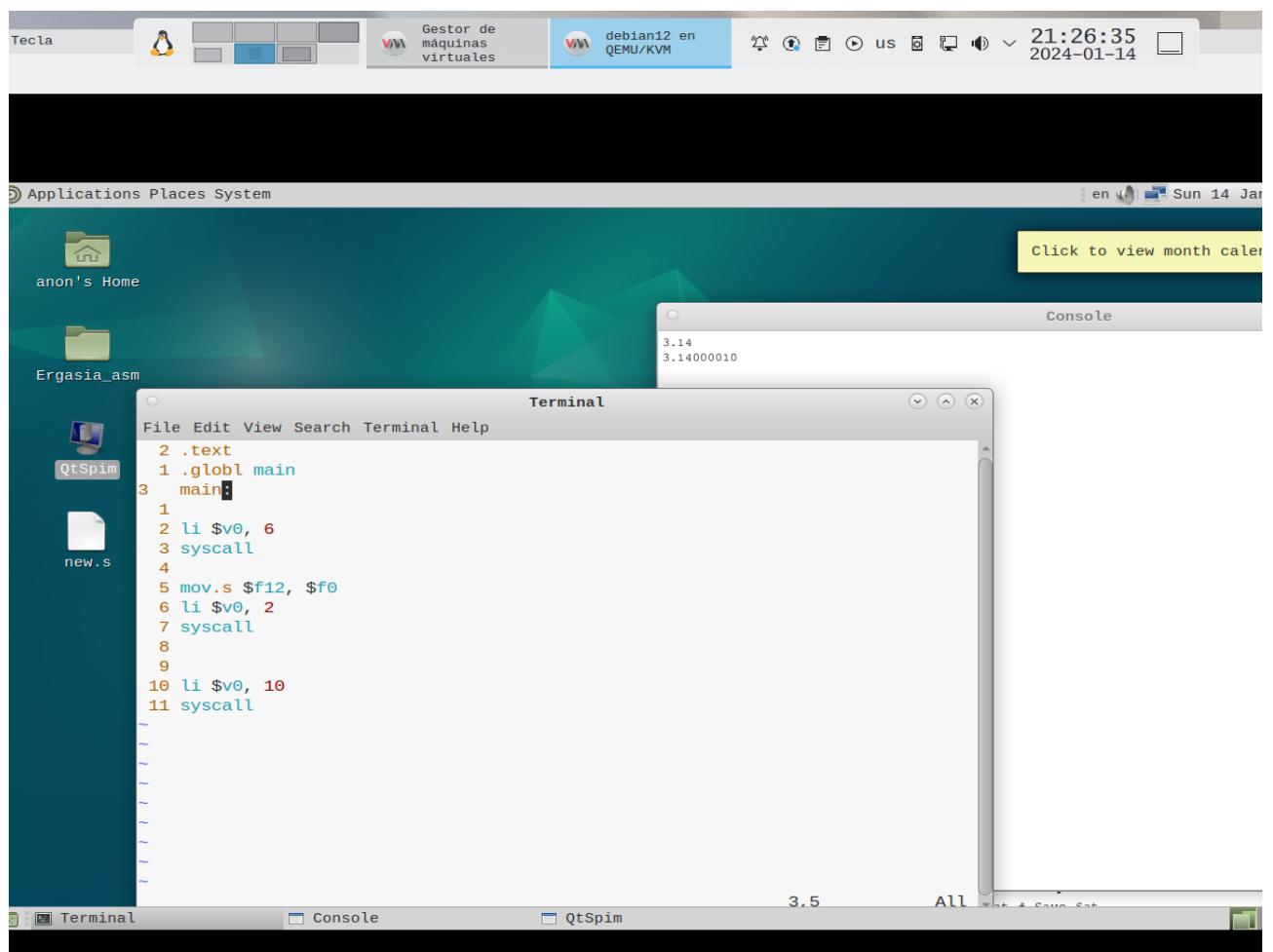
Ερώτηση 2

Αναλόγως τι τype θέλουμε αλλάζει η διεργασία για εισαγωγή στοιχείων.

Int

```
8 .text
7 .globl main
6 main:
5  li $v0, 5      # syscall read int
4  syscall        # call
3
2
1
10 # den xreiazetai apla to ebala gia na to kanei print gia na sigoureftw oti to egrapsa swsta
1 move $a0, $v0
2 li $v0, 1
3 syscall
4
5 #EXIT
6 li $v0, 10
7 syscall
```

Float



String

```

10 .data
 9   buffer: .space 1024 # max mege8os buffer
 8
 7 .text
 6 .globl main
 5
 4 main:
 3 li $v0, 8          #syscall read str
 2 la $a0, buffer    # dieu8hnsh tou buffer
 1 li $a1, 1024      # mhkos buffer
11 syscall
 1
 2
 3
 4
 5
 6 li $v0, 4
 7 la $a0, buffer
 8 syscall

```

Ερώτηση 3

read integer	5		\$v0 contains integer read
read float	6		\$f0 contains float read
read double	7		\$f0 contains double read
read string	8	\$a0 = address of input buffer \$a1 = maximum number of characters to read	See note below table

Αναλόγως τι syscall εκτελούμε. Για read str είναι η ίδια διεύθυνση του τύπου .space που φορτώσαμε ως adress στο \$a0. Άρα η προτελευταία γραμμή στην ερώτηση 2 δεν χρειάζεται, αλλά μπορεί σε ένα κανονικό πρόγραμμα να αποθηκευτεί κάτι άλλο στο \$a0, για αυτό άφησα πολλά κενά και ξανά φόρτωσα την διεύθυνση.

Ερώτηση 4

Service	Code in \$v0	Arguments	Πηγή
print integer	1	\$a0 = integer to print	
print float	2	\$f12 = float to print	
print double	3	\$f12 = double to print	
print string	4	\$a0 = address of null-terminated string to print	

2η Άσκηση – MIPS (μέρος 2o)

Περίληψη

Από δυαδικό αριθμό σε δεκαδικό με έλεγχο για μη έγκυρες εισόδους.

Υλοποίηση

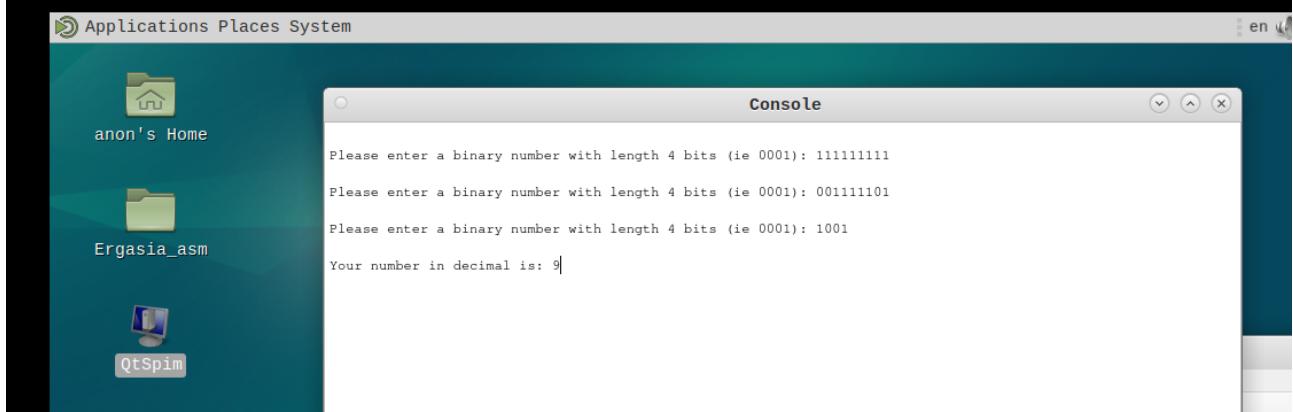
Κώδικας

[GitHub mirror](#)

[Pastebin mirror](#) Password: 12341

Φυσικά υπάρχει και μέσα στα αρχεία της εργασίας.

Screenshots



2η Άσκηση – MIPS (μέρος 3ο)

Περίληψη

Ζητάει ονοματεπώνυμο, ΑΜ, εξάμηνο. Αριθμομηχανή με λειτουργία πρόσθεση, αφαίρεση, πολλαπλασιασμό, διαιρέση, $\wedge 2$. Αν εισαχτεί το ΑΜ μου (2792) τελειώνει το πρόγραμμα.

Υλοποίηση

Kώδικας

[GitHub mirror](#)

[Pastebin mirror](#) Password: 12341

Φυσικά υπάρχει και μέσα στα αρχεία της εργασίας.

Screenshots

```
Enter full name: sdfsd fsdf

Enter rolling number: 279123123

Enter semester: 3

Enter first number: 5

Enter second number: -5

Enter valid operation (+, -, *, /, p [power of two]): p
Result: 25
Result: 25
Enter first number: 5

Enter second number: -5

Enter valid operation (+, -, *, /, p [power of two]): +
Result: 0
Enter first number: 5

Enter second number: -5

Enter valid operation (+, -, *, /, p [power of two]): -
Result: 10
Enter first number: 5

Enter second number: 3

Enter valid operation (+, -, *, /, p [power of two]): *
Result: 15
```

```
Enter first number: 5  
Enter second number: /2  
  
Enter valid operation (+, -, *, /, p [power of two]): /  
Result: 15  
Enter first number: 5  
  
Enter second number: 2  
  
Enter valid operation (+, -, *, /, p [power of two]): /  
Result: 2  
Enter first number: 2  
  
Enter second number: 2792
```
