ΨΗΦΙΑΚΟΙ ΥΠΟΛΟΓΙΣΤΕΣ

ΑΝΑΦΟΡΑ ΕΡΓΑΣΤΗΡΙΟΥ 3

ΜΙΧΑΛΗΣ ΓΑΛΑΝΗΣ 2016030036
ΓΙΩΡΓΟΣ ΒΙΡΙΡΑΚΗΣ 2016030035

Προεργασία

Στη 3η εργαστηριακή άσκηση μας ζητήθηκε να κατασκευάσουμε ένα πρόγραμμα σε Clang που αποτελούνταν από 3 λειτουργίες. Ύστερα κληθήκαμε να μετατρέψουμε μέρος του σε γλώσσα Assembly. Για δική μας διευκόλυνση υλοποιήσαμε αρχικά το πρόγραμμα σε γλώσσα C και μετά το μετατρέψαμε στις άλλες δυο.

Περιγραφή Ζητούμενων

Σκοπός αυτού του εργαστηρίου ήταν η βαθύτερη εξοικείωση μας με τη γλώσσα Clang και το πως λειτουργεί η μνήμη. Ήρθαμε επίσης και πρώτη φορά σε επαφή με τη γλώσσα Assembly και μάθαμε τα βασικά χαρακτηριστικά της.

Το πρόγραμμα εκτελούνταν επαναληπτικά και περιείχε 3 συναρτήσεις, μία για κάθε λειτουργία που μας ζητήθηκε. Οι λειτουργίες αυτές είχαν γενικώς να κάνουν με ανάγνωση και εκτύπωση στοιχείων, χρήση επαναληπτικών βρόγχων και πέρασμα πινάκων σε συναρτήσεις μέσω των διευθύνσεών τους.

Περιγραφή της Εκτέλεσης

Το πρόγραμμά μας περιείχε τις συναρτήσεις **main, function_1, function_2** και **function_3.**

- **CLANG**

  Αρχικά, δηλώσαμε τους 2 πίνακες ώς **global** int μεταβλητές καθώς και τους απαραίτητους καταχωρητές στη μνήμη.

```
int R0 = 0, R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19;
int arrayIn[5], arrayOut[5];
```

  Στη main βρισκόταν το menu (διεπαφή με το χρήστη) και για κάθε μία λειτουργία γινόταν η ανάγνωση στοιχείων από τον χρήστη και το πέρασμα τους στις αντίστοιχες συναρτήσεις. Για τη συνάρτηση 1 για παράδειγμα:

```
printf("Please Enter an Integer (N): ");
scanf("%d", &R3);
N = R3;
R2 = function_1(R3);
result = R2;
printf("\nTotal Numbers printed: %d", R2);
```

  Στη πρώτη συνάρτηση χρειάστηκε εμφωλευμένη for loop για να κατασκευάσουμε το ζητούμενο.

Στη δεύτερη συνάρτηση για να αποφύγουμε τη χρήση της συνάρτησης modulo έπρεπε ανάλογα με το πρόσημο του αριθμού να προσθέσουμε ή να αφαιρέσουμε το 2 μέχρι να γίνει 0 ή 1.

```
if (R4 < R0) goto else_label;

While_label_1:
if (R4 <= 1) goto after_loop_1;
R4 = R4 - 2;
goto While_label_1;
after_loop_1:
return R4;

else_label:

//while 2
While_label_2:
if (R4>=-1) goto after_loop_2;
R4 = R4 + 2;
goto While_label_2;
after_loop_2:
return (-R4);
```

Στην τρίτη συνάρτηση, αντί να περάσουμε τους δυο πίνακες στη συνάρτηση, περάσαμε τις διευθύνσεις των δύο πρώτων στοιχείων των πινάκων. Χρησιμοποιήσαμε επίσης for loop για να αποφύγουμε τη πράξη πολλαπλασιασμού.

```
int function_3(int R4, int R5){
    int i;
    R15 = R0;
    i = R15;

    Label_loop_1:
    if (R15 >= 5) goto after_loop_1;

    //another for loop
    int j;
    R16 = R0;
    j = R16;
    Label_loop_2:
    if (R16>=6) goto after_loop_2;
    R17 = arrayOut[R15];
    R18 = arrayIn[R15];
    R19 = R17 + R18;
    arrayOut[R15] = R19;
    R16 = R16 + 1;
    goto Label_loop_2;
    after_loop_2:
    R15 = R15 + 1;
    goto Label_loop_1;

    after_loop_1:
    return (long) arrayOut;
}
```

- ## ASSEMBLY

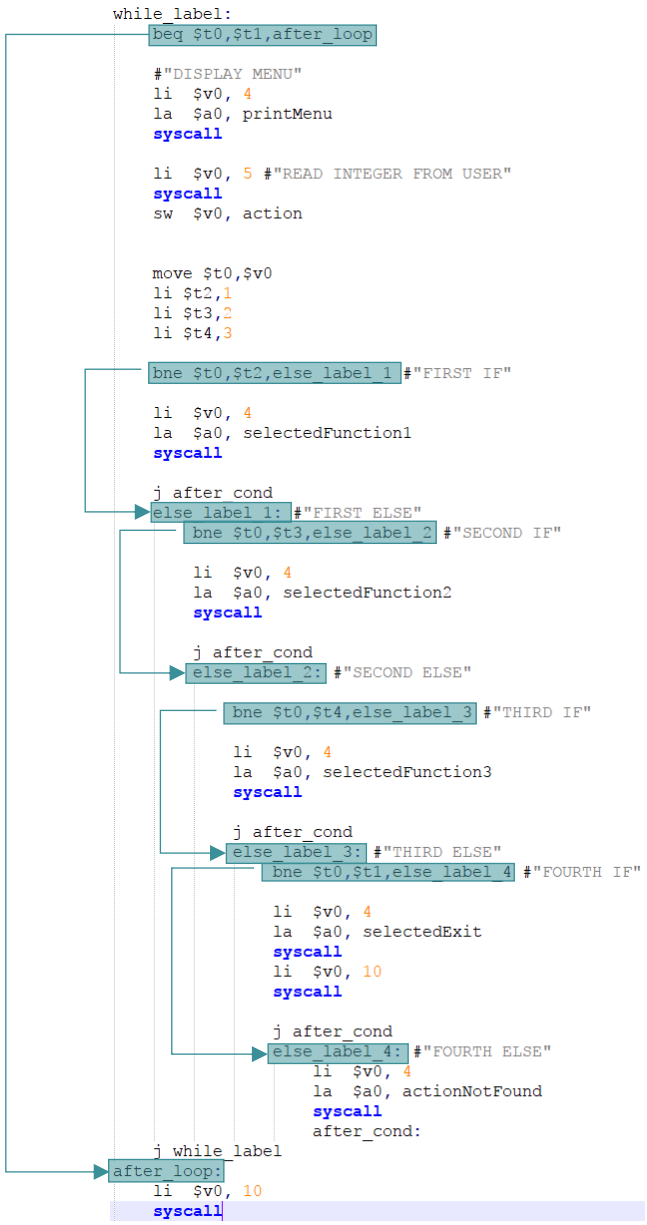
Στην Assembly, μας ζητήθηκε να υλοποιήσουμε μόνο το menu. Αντί να εκτελεστούν οι ενέργειες της κάθε συνάρτησης, εκτυπώνεται απλά ένα μήνυμα λέγοντας ποια συνάρτηση έχουμε επιλέξει.

Η εκτύπωση και ανάγνωση στοιχείων γίνεται ανάλογα με τον κωδικό αριθμό του $v0.

```
#"DISPLAY MENU"
li  $v0, 4
la  $a0, printMenu
syscall

li  $v0, 5 #"READ INTEGER FROM USER"
syscall
sw  $v0, action
```

Η αλλαγή της ροής του προγράμματος γινόταν με target branches ως εξής:



Συμπέρασμα

Στο 3ο εργαστήριο εμβαθύναμε περαιτέρω στη Clang και κάναμε μια αρχή στη γλώσσα Assembly.