1η ΑΣΚΗΣΗ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

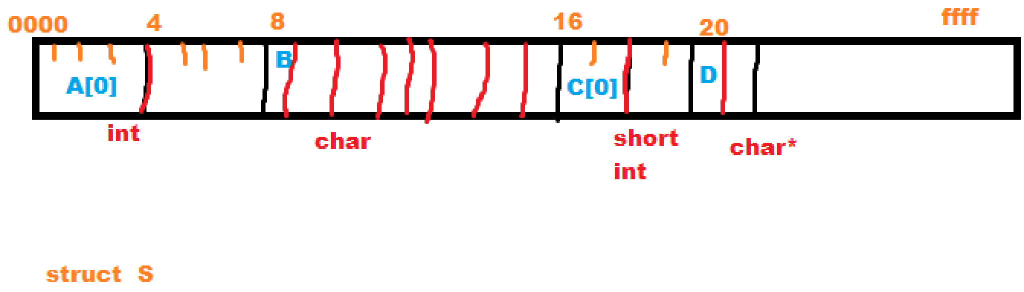
Ακ. έτος 2018-2019 ,5ο Εξάμηνο ,Σχολή: ΗΜ&ΜΥ

Ονοματεπώνυμο: Βερέμης Αλέξανδρος
ΑΜ: 03115063

ΜΕΡΟΣ Α

```
        addi   $s1, $s0, 32 #p = &array[8];
LOOP:   lw     $t0, 0($s1) #$t0=*p
        beq    $t0, $zero, END # if (*p==0) go to end
        div    $t0, $s2 # start of :if (*p < 100)
        slti   $t1, $t0, 100 # *p = *p % N;
        beq    $t1, $zero, ELSE #
        mfhi   $t0       #
        jmp    NEXT      # end of it
ELSE:   mflo   $t0      #else *p = *p / N;
NEXT:   sw     $t0, 0($s1) # array[i]=*p
        addi   $s1, $s1, 4 # p++
        jmp    LOOP    # }
END:
```

ΜΕΡΟΣ Β



Το struct S αποδίδεται παραπάνω.

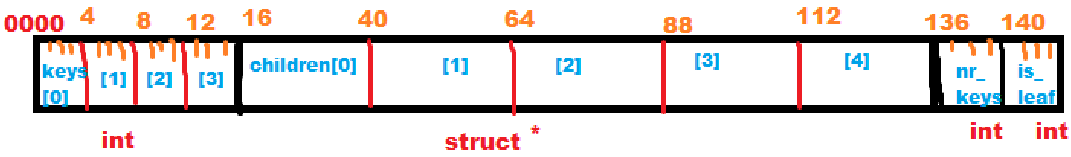
Αντιστοιχίζω s → $a0 , a→$t0, b→$t1 ,c→$t2 ,d →$t3 ,i→$t4.

```
Foo: and $t0,$t0,$zero #$t0=0x000000000

      lui $t0,0xdead
      ori $t0,$t0,0xbeef #a=deadbeef

    and $t1, $t1,$zero #b=0

    and $t2, $t2,$zero #c=0
    addi $t3,$a0,21
    lb  $t3,0($t3) #d= s→D[1]


    add $t4,$t4,$zero #i=0

LOOP1: slti $t5,$t4,8

        beq $t5,$zero,END1 #if(i>=2)go to END1
        add $t6, $a0, $t4
        lw $t6,0($t6) #$t6=s→A[i]
        add $t0, $t0, $t6 #a=a+s→A[i]
        addi $t4, $t4,4 #i++
        j LOOP1
```

```
END1 : addi $t4,$zero,8 #end of loop

#θέτω το $t4=8 να με βγαλει στο B[0].
LOOP2 : slti $t5, $t4,16


        beq $t5,$zero,END2
       add $t6, $a0, $t4
       lw $t6,0($t6)
        add $t1, $t1, $t6
        addi $t4, $t4,1
        j   LOOP2

END2 : addi $t4,$zero,16

LOOP3: slti $t5, $t4,20

      beq $t5,$zero,END3
      add $t6, $a0, $t4
      lh $t6,0($t6)
      add $t2, $t2, $t6
      addi $t4, $t4,2
      j LOOP3


END3 : addi $t4,$zero,30
       div $t0, $t4 # if (a% 30 < 10)
       mfhi $t4 #(same)
       slti $t4, $t4,10 #(same)
       beq $t4,$zero,PARAKATO #if not
       mflo $t0  # a=a/30
```

```
#όμοια πράγματα ισχύουν και για τις
#3 επαναλήψεις

PARAKATO : addi $t6, $a0,8
           sb $t1,0($t6)#s→B[0]=b
           addi $t6, $a0,16
           sh $t2,0($t6) #s→C[0]=c
           add $v0,$t0,$zero #return a
            jr $ra
```

ΜΕΡΟΣ Γ



struct bpt_node

Θεωρώ : get_children_index(a0,a1,a2)

```
get_children_index: and $t0, $t0,$zero #i=0
LOOP : slti $t1, $t0, $a1 #t1=1 if t0<a1
      add $t2, $a0, $t0 #t2= &A[i]
      lw $t2,($t2) #t2 =A[i]
      slti $t3, $a2, $t2 #t3=1 if a2<t2
      bne $t1, $t3,END
      addi $t0, $t0,1 #i++
      j LOOP

END : add $v0,$t0,$zero #return i
      jr $ra
```

θεωρώ : bpt_lookup (a0,a1)

```
bpt_lookup: bne $a0,$zero,KATO  #if n=0
            and $v0,$zero,$zero #return 0
            jr $ra

KATO: addi $sp, $sp ,-12 #saving in stack $ra,$a0,$a1
     sw $a0,0($sp)
     sw $a1,4($sp)
     sw $ra,8($sp)
     add $a0, $a0,$zero #a0=&(n->keys[0])
     add $a2, $a1,$zero #a2=a1=key
     addi $a1, $a0,136 #a1=n->nr_keys
     jal get_children_index #calling get_children_index
     lw $a0,0($sp) #loading from stack $ra,$a0,$a1
     lw $a1,4($sp)
     lw $ra,8($sp)

     addi $sp, $sp,12


     add $t6, $a0,140 #t6=&(n->is_leaf)
     lw $t6,0($t6) #t6=n->is_leaf
     addi $t5,$zero,1 #t5=1
     bne $t6, $t5,PARAKATO2 # if n->is_leaf != 1
     add $t5,$v0,$zero #t5=index
     addi $t4,$zero,4
     mult $t5, $t4
     mflo $t5
     add $t4, $a0, $t5 #t4=&(n->keys[index])
     lw $t4,0($t4) #t4= n->keys[index]
     bne $t4, $a1,PARAKATO2 #if n->keys[index]!=key
     addi $v0,$zero,1 #return ret=1
     j     END2


PARAKATO2:     addi $sp, $sp ,-12 #saving $ra,$a0,$a1
               sw $a0,0($sp)
               sw $a1,4($sp)
               sw $ra,8($sp)
               addi $t4,$zero,6
               mult $t5, $t4
               mflo $t5 #index*24
               add $t4, $a0, $t5
      # t4=&(n->children[index] - 16)
               addi $t4, $t4, 16
      # t4=&(n->children[index])
               lw $t4,0($t4) # t4=n->children[index]
               add $a0, $t4,$zero #a0= n->children[index]

               add $a1, $a1,$zero #a1=key (for safety
                               # reasons)
               jal bpt_lookup #calling bpt_lookup

               lw $a0,0($sp) #loading $ra,$a0,$a1
               lw $a1,4($sp)
               lw $ra,8($sp)

               addi $sp, $sp,12
               add $v0,$v0,$zero #ret= bpt_lookup(…)

END2:           jr $ra #return
```