MIPS Disc1

Topics

- 1. Overview of mips programming
- 2. Mips exercise

Exercise

Try

C	MIPS

// $s0 -> a, $s1 -> b
// $s2 -> c, $s3 -> z
int a = 4, b = 5, c = 6, z;
z = a + b + c + 10;
	addiu $s0, $0, 4
	addiu $s1, $0, 5
	addiu $s2, $0, 6

Try

C	MIPS

// $s0 -> a, $s1 -> b
// $s2 -> c, $s3 -> z
int a = 4, b = 5, c = 6, z;
z = a + b + c + 10;
	addiu $s0, $0, 4
	addiu $s1, $0, 5
	addiu $s2, $0, 6
	
	addu $s3, $s0, $s1
	addu $s3, $s3, $s2
	addiu $s3, $s3, 10

Try

C	MIPS

// $s0 -> int * p = intArr;
// $s1 -> a;
*p = 0;
int a = 2;
p[1] = p[a] = a;
	la $s0 intArr

Try

C	MIPS

// $s0 -> int * p = intArr;
// $s1 -> a;
*p = 0;
int a = 2;
p[1] = p[a] = a;
	la $s0 intArr
	sw $0, 0($s0)
	addiu $s1, $0, 2
	sw $s1, 4($s0)
	sll $t0, $s1, 2
	add $t0, $t0, $s0
	sw $s1, 0($t0)

Try

C	MIPS

// $s0 -> a, $s1 -> b
int a = 5, b = 10;
if (a + a == b) {
  a = 0;
} else {
  b = a - 1;
}	

Try

C	MIPS

// $s0 -> a, $s1 -> b
int a = 5, b = 10;
if (a + a == b) {
  a = 0;
} else {
  b = a - 1;
}
	addiu $s0, $0, 5
	addiu $s1, $0, 10
	addu $t0, $s0, $s0
	bne $t0, $s1, else
	xor $s0, $0, $0
	j exit
	else:
	addiu $s1, $s0, -1
	exit:

Try

C	MIPS

	addiu $s0, $0, 0
	addiu $s1, $0, 1
	addiu $t0, $0, 30
	loop:
	beq $s0, $t0, exit
	addu $s1, $s1, $s1
	addiu $s0, $s0, 1
	j loop
	exit:

Try

C	MIPS

// computes s1 = 2^30
s1 = 1;
for(s0=0;s0<30;s0++) {
  s1 *= 2;
}
	addiu $s0, $0, 0
	addiu $s1, $0, 1
	addiu $t0, $0, 30
	loop:
	beq $s0, $t0, exit
	addu $s1, $s1, $s1
	addiu $s0, $s0, 1
	j loop
	exit: