Arquitetura de Computadores I

Ano Letivo 2014/15 - 1º Semestre

Mini-Teste Prático 1 – Terça-Feira, 28/10/2014, 9:00-11:00

Nº Mec.: _____ Nome: _____PROPOSTA DE RESOLUÇÃO_____ Turma: _____

NOTE BEM: Leia atentamente todas as questões, comente o código usando a linguagem C. Na tradução para o *Assembly* do MIPS respeite rigorosamente os aspectos estruturais e a sequência de instruções indicadas no código original fornecido.

1) Codifique em *assembly* do MIPS a seguinte função `main`:

```
#define SIZE 4
void main()
{
   char op[SIZE];
   int num1,num2,res,erro=0 ;
   //
   print_str("Introduza dois numeros:\n");
   num1 = read_int();
   num2 = read_int();
   //
   print_string("Operador [^,<<]: ");
   read_string(op,SIZE);
   //
   if(op[0] == '^' )
      res = num1 ^ num2;
   else if((op[0]=='<') && (op[0]==op[1]) )
      res = num1 << num2;
   else erro = 1;
   //
   if(erro) print_string("Erro :-(!\n");
   else {
      print_string("Resultado = ");
      print_int10(res);
   }
}
```

Variável	Registo(s)
num1	$t0
num2	$t1
res	$t2
erro	$t3

Label	Instrução em *assembly*		Comentário em C
	.eqv	SIZE, 4	
	.data		
op:	.byte	SIZE	
s_nbrs:	.asciiz	"Introduza dois numeros:\n"	
s_oper:	.asciiz	"Operador [^,<<]: "	
s_erro:	.asciiz	"Erro :-(!\n	
s_res:	.asciiz	"Resultado = "	
	.text		
	.globl main		
main:	la	$a0, s_nbrs	print_string(s_nbrs)
	li	$v0, 4	
	syscall		
	li	$v0, 5	num1 = read_int()
	syscall		
	move	$t0, $v0	
	li	$v0, 5	num2 = read_int()
	syscall		
	move	$t1, $v0	

Label	Instrução em *assembly*		Comentário em C
	la	$a0, s_oper	print_string(s_oper)
	li	$v0, 4	
	syscall		
	la	$a0, op	read_string(op,SIZE)
	li	$a1, SIZE	
	li	$v0, 8	
	syscall		
	la	$t6, op	
	lb	$t4, 0($t6)	op[0]
	lb	$t5, 1($t6)	op[1]
if1:	bne	$t4, '^', if2	if(op[0] =='^')
	xor	$t2, $t1, $t0	res = num1 ^ num2
	b	print	
if2:	bne	$t4, '<', if3	else if((op[0]=='<')...
	bne	$t4, $t5, if3	
	sllv	$t2, $t0, $t1	res = num1 << num2
	b	print	
if3:	li	$t3, 1	erro = 1
print:	bne	$t3, 1, no_err	
	la	$a0, s_erro	print_string(s_erro)
	li	$v0, 4	
	syscall		
	b	done	
no_err:	la	$a0, s_res	print_string(s_res)
	li	$v0, 4	
	syscall		
	move	$a0, $t2	print_int10(res)
	li	$v0, 1	
	syscall		
done:	jr	$ra	

Mini-Teste Prático 1 – Terça-Feira, 28/10/2014, 9:00-11:00

Nº Mec.: _____ Nome: _____PROPOSTA DE RESOLUÇÃO_____ Turma: _____

2) Codifique em *assembly* do MIPS a seguinte função `main`:

```
#define SIZE 20
void main()
{
   static int primos[SIZE];
   int nbr,c;
   int * p = primos;
   int * pstop = &primos[SIZE];
   //
   *p = 2;
   p++;
   nbr = 3;
   while(p < pstop )
   {
      c=2;
      //sugestão: use a instrução 'rem'
      while( nbr%c ) c++;
      if( c == nbr ) {
           *p = nbr;
           p++;
      }
      nbr++;
    }
}
```

Variável	Registo(s)
nbr	$t1
c	$t0
p	$a0
pstop	$a1

Label	Instrução em *assembly*		Comentário em C
	.eqv	SIZE, 20	
	.eqv	SIZE4,80	
	.data		
primos:	.word	0:SIZE	
	.text		
	.globl	main	
main:	la	$a0, primos	p=primos
	addiu	$a1,$a0,SIZE4	pstop=&primos[SIZE]
	li	$t0, 2	c=2
	sw	$t0, ($a0)	*p = 2
	addiu	$a0, $a0, 4	p++
	#		
	li	$t1, 3	nbr = 3
wh1:	bgeu	$a0,$a1,wh1_end	while (p < pstop) {
	li	$t0, 2	c = 2
wh2:	rem	$t3, $t1, $t0	while (nbr%c) {
	beqz	$t3, wh2_end	
	addi	$t0, $t0, 1	c++
	b	wh2	
wh2_end:	#		}

Label	Instrução em *assembly*		Comentário em C
if:	bne	$t0, $t1, if_end	if(c == nbr){
	sw	$t1, ($a0)	*p = nbr
	addiu	$a0, $a0, 4	p++
if_end:	#		}
	addi	$t1, $t1, 1	nbr++
	b	wh1	
wh1_end:	#		}
	jr	$ra	

Arquitetura de Computadores I

Ano Letivo 2014/15 - 1º Semestre

Mini-Teste Prático 1 – Quinta-Feira, 30/10/2014, 9:00-11:00

Nº Mec.: _____ Nome: _____PROPOSTA DE RESOLUÇÃO_____ Turma: _____

NOTE BEM: Leia atentamente todas as questões, comente o código usando a linguagem C. Na tradução para o *Assembly* do MIPS respeite rigorosamente os aspectos estruturais e a sequência de instruções indicadas no código original fornecido.

1) Codifique em *assembly* do MIPS a seguinte função `main`:

```
#define CHAR_BIN_LEN 33
#define MS_BIT 31
void main()
{
   int n,c,k;
   char bin_nbr[CHAR_BIN_LEN];
   char* p = bin_nbr;
   //
   print_string("Inteiro em decimal: " );
   n=read_int();
   //
   print_string("Em binario e:  " );
   for(c = MS_BIT; c >= 0; c--, p++)
   {
       k = n >> c;
       if(k & 1)   *p = '1';
       else        *p = '0';
   }
   *p='\0';
   print_string(bin_nbr);
}
```

Variável	Registo(s)
n	$t1
c	$t0
k	$t2
p	$a0

Label	Instrução em *assembly*		Comentário em C
	.eqv CHAR_BIN_LEN,33		
	.eqv MS_BIT,31		
	.data		
bin_nbr:	.space CHAR_BIN_LEN		
s_dec:	.asciiz "Introduza um inteiro em decimal: "		
s_bin:	.asciiz "Em binario e: "		
	#		
	.text		
	.globl main		
main:	la	$a0, s_dec	print_string(s_dec)
	li	$v0, 4	
	syscall		
	#		
	li	$v0, 5	n = read_int()
	syscall		
	move	$t1, $v0	
	#		
	la	$a0, s_bin	print_string(s_bin)
	li	$v0, 4	
	syscall		
	#		

Label	Instrução em *assembly*		Comentário em C
	# for(c = MS_BIT; c >= 0; c--, p++) {		
	la	$a0, bin_nbr	p = bin_nbr
	li	$t0, MS_BIT	c = MS_BIT
for:	bltz	$t0, for_end	
	srlv	$t2, $t1, $t0	k = n >> c
	andi	$t2, $t2, 1	k&1
	beqz	$t2, is_zero	
	li	$t2, '1'	
	sb	$t2, ($a0)	*p = '1'
	b	f_next	
is_zero:	li	$t2, '0'	
	sb	$t2, ($a0)	*p = '0'
f_next:	addiu	$a0, $a0, 1	p++
	addi	$t0, $t0, -1	c--
	b	for	
for_end:	#		}
	sb	$0, ($a0)	*p='\0'
	#		
	la	$a0, bin_nbr	print_string(bin_nbr)
	li	$v0, 4	
	syscall		
	#		
	jr	$ra	

Mini-Teste Prático 1 – Quinta-Feira, 30/10/2014, 9:00-11:00

Nº Mec.: _____ Nome: _____PROPOSTA DE RESOLUÇÃO_____ Turma: _____

2) Codifique em *assembly* do MIPS a seguinte função `main`:

```
#define MAX  0x80000000
#define MIN  0x7FFFFFFF
#define SIZE 10
void main()
{
static int val[SIZE] =
       {8,4,-3,5,124,15,-8987,9,27,16};
   int maximo=MAX;
   int minimo=MIN;
   int i;
   int *p=val;
   //
   for(i=0; i< SIZE; i++, p++) {
      if(*p > maximo) maximo = *p;
      if(*p < minimo) minimo = *p;
   }
   print_string("Maximo: ");
   print_int10(maximo);
   //
   print_string("\nMinimo: ");
   print_int10(minimo);
}
```

Variável	Registo(s)
p	$a0
i	$t0
maximo	$t2
minimo	$t3

Label	Instrução em *assembly*	Comentário em C
	.eqv MAX,0x80000000	
	.eqv MIN,0x7FFFFFFF	
	.eqv SIZE,10	
	.data	
val:	.word 8,4,-3,5,124,15,-8987,9,27,16	
s_max:	.asciiz "Maximo: "	
s_min:	.asciiz "\nMinimo: "	
	.text	
	.globl main	
main:	la $a0, val	p=val
	li $t2, MAX	maximo =MAX
	li $t3, MIN	minimo = MIN
	li $t0, 0	i = 0
	# for(i=0; i< SIZE; i++, p++) {	
for:	bge $t0, SIZE, for_end	
	lw $t1, ($a0)	$t1 = *p
	ble $t1, $t2, chk_min	if (*p > maximo)
	move $t2, $t1	maximo=*p
chk_min:	bge $t1,$t3, for_next	if (*p < minimo)

Label	Instrução em *assembly*	Comentário em C
	move $t3, $t1	minimo=*p
for_next:	addiu $a0, $a0, 4	p++
	addi $t0, $t0, 1	i++
	b for	
for_end:	#	}
	la $a0, s_max	**print_string** (s_max)
	li $v0, 4	
	syscall	
	move $a0, $t2	**print_int10** (maximo)
	li $v0, 1	
	syscall	
	la $a0, s_min	**print_string** (s_min)
	li $v0, 4	
	syscall	
	move $a0, $t3	**print_int10** (minimo)
	li $v0, 1	
	syscall	
	#	
	jr $ra	