



Nome:

Gabari70

Matrícula: /

Prova 1

(7.0) 1) Em engenharia, o ato de a partir de um produto (equipamento, circuito, etc) recriar seu projeto (diagrama esquemático, modelos, etc) é chamado de Engenharia Reversa. Em computação, com a Engenharia Reversa se pode obter o código fonte de um programa em uma linguagem de alto nível a partir de seu código executável.

Dados os segmentos de memória de dados e de texto abaixo:

- a) (3.0) “Desassemble” o programa em linguagem de máquina em linguagem Assembly MIPS.
- b) (3.0) “Descomplie” o programa em Assembly MIPS em um programa em C que o possa ter gerado.
- c) (1.0) Qual o significado do que é escrito na tela?

Segmento de Dados (Little Endian)	Endereço	Conteúdo (hexa)
...		
0x10010000	0xd454e4f	
0x10010004	0x52004947	
0x10010008	0x55414c00	
0x1001000C	0xd41494f	
0x10010010	0x52000000	
...		

Segmento de Texto	Endereço	Conteúdo (hexa)	Conteúdo (bin)
...			
0x00400038	0x1085000c	00010000100001010000000000000001100	
0x0040003c	0x0085402a	00000000100001010100000000101010	
0x00400040	0x11000005	0001000100000000000000000000000101	
0x00400044	0x20020004	0010000000000001000000000000000100	
0x00400048	0x3c011001	00111100000000010001000000000001	
0x0040004c	0x34240000	0011010000100100000000000000000000000000	
0x00400050	0x00000000c	000000000000000000000000000000001100	
0x00400054	0x03e00008	00000011110000000000000000000001000	
0x00400058	0x20020004	00100000000000010000000000000001000	
0x0040005c	0x3c011001	00111100000000010001000000000001	
0x00400060	0x3424000c	00110100001001000000000000000001100	
0x00400064	0x00000000c	00000000000000000000000000000001100	
0x00400068	0x03e00008	00000011110000000000000000000001000	
0x0040006c	0x20020004	00100000000000010000000000000001000	
0x00400070	0x3c011001	00111100000000010001000000000001	
0x00400074	0x34240006	00110100001001000000000000000001100	
0x00400078	0x00000000c	00000000000000000000000000000001100	
0x0040007c	0x03e00008	00000011110000000000000000000001000	
...			

(2.0) 2) A arquitetura x86 possui instruções bastante poderosas inexistente na arquitetura MIPS. A instrução MAC (Multiply and Accumulate) é muito útil para realizar filtragem digital. Escreva o código real da pseudo-instrução abaixo:

Macmm.s Imm^1 Imm^2 Imm^3 Imm^4 Imm^5 # $\text{Memory}[\$t0+100] = \text{Memory}[\$t0+100] + \$f0 * \text{Memory}[\$t1+100]$

(1.0) 3) Explique como e porque a média geométrica é usada como ferramenta para resumir medidas de desempenho na comparação de processadores.

(2.0) 4) A Arquitetura x86 deriva da arquitetura de seu primeiro processador de 32 bits, o Intel 80386.

a) (1.0) Qual o número de registradores de uso geral deste processador 386? Qual o número aproximado de instruções da sua ISA?

b) (1.0) O último lançamento da Intel, o processador Core i7, possui quantos registradores de uso geral? Qual o número aproximado de instruções da sua ISA?

BOA SORTE!

Gabarito

1)

9)

Endereço	conteúdo	Assembly	
0x00400038	0x1085000C	beq \$a0, \$a1, 0x000C	
0,1 03C	0x0085402A	SLT \$t0, \$a0, \$a1	
040	0x11000005	beq \$t0, \$ZERO, 0x0005	
044	0X20020004	addi \$v0, \$ZERO, 4	0,3
048	0X3C011001	Lui \$at, 0x1001	
04C	0X34240000	ori \$a0, \$at, 0x0000	
050	0X0000000C	SYSCALL	
054	0x03E0 0008	jr \$ra	
058	0X20020004	addi \$v0, \$ZERO, 4	
05C	0X3C011001	Lui \$at, 0x1001	
060	0X3424000C	ori \$a0, \$at, 0x000C	
064	0X0000000C	SYSCALL	
068	0x03E0 0008	jr \$ra	
06C	0X20020004	addi \$v0, \$ZERO, 4	
070	0X3C011001	Lui \$at, 0x1001	
074	0X34240006	ori \$a0, \$at, 0x0006	
078	0X0000000C	SYSCALL	
07C	0x03E0 0008	jr \$ra	

4d 45 4e 4f 52 00

0,2 0x10010000	M E N O R	0
0,2 0x1001 0006	I G U A L	0
0,2 0x1001 000C	M A I O R	0

b) É um procedimento:

```
// Int 4
void Proc (int A, int B)
{
    if (A == B)
        printf ("IGUAL");
    else
        if (!CAL(B)) // ou (A > B)
            printf ("MAIOR");
        else
            printf ("MENOR");
    // return 4;
}
```

c) DADO 2 VALORES A e B

escreve a relação entre A e B
com A maior, menor ou igual a B

2) PSEUDO INSTRUÇÃO: não pode modificar nenhum registrador ADICIONAL!

```
addi $sp,$sp,-8
swc1 $f2,4($sp)          # forma 1
swc1 $f1,0($sp)
addi $at,$t1,imm2
lwcl $f1,0($at)
mul.s $f1,$f2,$f1
addi $at,$t0,imm1
lwcl $f2,0($at)
add.s $f2,$f2,$f1
swc1 $f2,0($at)
twc1 $f1,0($sp)
lwcl $f2,4($sp)
addi $sp,$sp,8
```

FORM 2 : considerando que o montador
é Caffaz de montan LWC1 c SWC1

addi \$5\$, \$sp, -8
swc1 \$f1, 0(\$sp)
swc1 \$f2, 4(\$sp)
lwc1 \$f1, Imm2(\$t1) ←
mul.s \$f1, \$f0, \$f1
lwc1 \$f2, Imm1(\$t0) ←
add.s \$f2, \$fe, \$f1
swc1 \$f2, Imm1(\$t0) ←
lwc1 \$f2, 4(\$sp)
lwc1 \$f1, 0(\$sp)
addi \$sp, \$sp, 8

3) É usada a média geométrica dos tempos de processamento normalizado devido à propriedade

$$MG\left(\frac{Y}{X}\right) = \frac{MG(Y)}{MG(X)} \rightarrow \text{é uma constante.}$$

O resultado é independente da máquina base e do workload utilizado,
ou seja, um resultado consistente.

4) Patterson na EDUCA Pág. 169

→ 386 → 8 Regs de 32 bits uso geral
→ cerca de 120 instruções

i7 → 64 Regs de 64 bits uso geral
~ 8 Regs de 30 bits float point
16 Regs 128 bits media Regs
→ mais de 900 instruções