

Instruções da Máquina Nativa				Instruções da Máquina Virtual				DETI-UA - ACI																																											
Transferência Memória-Registro (<i>Load</i>)		Cálculo c/ Inteiros: Operações Aritméticas		Transferência Memória-Registro (<i>Load</i>)		Salto Relativo (<i>Branch</i>)																																													
lb	Rdst, addr	add	Rdst, Rsrc1, Rsrc2	l.d	FPdst, addr	b	Label																																												
lbu	Rdst, addr	addi	Rdst, Rsrc, Imm	l.s	FPdst, addr	beqz	Rsrc, Label																																												
lw	Rdst, addr	addiu	Rdst, Rsrc, Imm	Transferência Registro-Memória (<i>Store</i>)		bnez	Rsrc, Label																																												
lwcx	CReg, addr	addu	Rdst, Rsrc1, Rsrc2	s.d	FPsrc, addr	bge	Rsrc, Src, Label																																												
Transferência Registro-Memória (<i>Store</i>)		div	Rsrc1, Rsrc2		FPsrc, addr	bgeu	Rsrc, Src, Label																																												
sb	Rsrc, addr	divu	Rsrc1, Rsrc2	Transferência Registro-Registro (<i>Move</i>)		bgt	Rsrc, Src, Label																																												
sw	Rsrc, addr	mult	Rsrc1, Rsrc2	move	Rdst, Rsrc	bgtu	Rsrc, Src, Label																																												
swcx	Creg, addr	multu	Rsrc1, Rsrc2		ble	Rsrc, Src, Label																																													
Transferência Registro-Registro (<i>Move</i>)		sub	Rdst, Rsrc1, Rsrc2	Manipulação de Const. (<i>Load Imm/sym</i>)		bleu	Rsrc, Src, Label																																												
mfhi	Rdst	subu	Rdst, Rsrc1, Rsrc2	la	Rdst, sym	blt	Rsrc, Src, Label																																												
mflo	Rdst	and	Rdst, Rsrc1, Rsrc2		li	Rdst, IMM	bltu			Rsrc, Src, Label																																									
mthi	Rsrc	andi	Rdst, Rsrc, Imm		l.d	FPdst, sym	beq	Rsrc, Src, Label																																											
mtlo	Rsrc	nor	Rdst, Rsrc1, Rsrc2		l.s	FPdst, sym	bne	Rsrc, Src, Label																																											
mfcz	Rdst, Creg	or	Rdst, Rsrc1, Rsrc2	Cálculo c/ Inteiros: Op. Aritméticas		<table><tr><th colspan="3">Tabela I: Registos do MIPS e convenção de uso</th></tr><tr><th>Nome Lóg.</th><th>Nome Real</th><th>Uso Convencionado</th></tr><tr><td>\$zero</td><td>\$0</td><td>Constante 0</td></tr><tr><td>\$at</td><td>\$1</td><td>Reservado pelo assembler</td></tr><tr><td>\$v0..\$v1</td><td>\$2..\$3</td><td>Cálculo de expressões e valor de retorno das</td></tr><tr><td>\$a0..\$a3</td><td>\$4..\$7</td><td>Primeiros 4 parâmetros das funções</td></tr><tr><td>\$t0..\$t7</td><td>\$8..\$15</td><td>Geral (não são preservados pelas funções)</td></tr><tr><td>\$s0..\$s7</td><td>\$16..\$23</td><td>Geral (não podem ser alterados pelas funções)</td></tr><tr><td>\$t8..\$t9</td><td>\$24..\$25</td><td>Geral (não são preservados pelas funções)</td></tr><tr><td>\$k0..\$k1</td><td>\$26..\$27</td><td>Reservado pelo <i>kernel</i> do S.O.</td></tr><tr><td>\$gp</td><td>\$28</td><td>Ponteiro para área global (<i>Global Pointer</i>)</td></tr><tr><td>\$sp</td><td>\$29</td><td><i>Stack Pointer</i></td></tr><tr><td>\$fp</td><td>\$30</td><td><i>Frame Pointer</i></td></tr><tr><td>\$ra</td><td>\$31</td><td>Endereço de retorno das funções (<i>Return Address</i>)</td></tr></table>				Tabela I: Registos do MIPS e convenção de uso			Nome Lóg.	Nome Real	Uso Convencionado	\$zero	\$0	Constante 0	\$at	\$1	Reservado pelo assembler	\$v0..\$v1	\$2..\$3	Cálculo de expressões e valor de retorno das	\$a0..\$a3	\$4..\$7	Primeiros 4 parâmetros das funções	\$t0..\$t7	\$8..\$15	Geral (não são preservados pelas funções)	\$s0..\$s7	\$16..\$23	Geral (não podem ser alterados pelas funções)	\$t8..\$t9	\$24..\$25	Geral (não são preservados pelas funções)	\$k0..\$k1	\$26..\$27	Reservado pelo <i>kernel</i> do S.O.	\$gp	\$28	Ponteiro para área global (<i>Global Pointer</i>)	\$sp	\$29	<i>Stack Pointer</i>	\$fp	\$30	<i>Frame Pointer</i>	\$ra	\$31	Endereço de retorno das funções (<i>Return Address</i>)
Tabela I: Registos do MIPS e convenção de uso																																																			
Nome Lóg.	Nome Real	Uso Convencionado																																																	
\$zero	\$0	Constante 0																																																	
\$at	\$1	Reservado pelo assembler																																																	
\$v0..\$v1	\$2..\$3	Cálculo de expressões e valor de retorno das																																																	
\$a0..\$a3	\$4..\$7	Primeiros 4 parâmetros das funções																																																	
\$t0..\$t7	\$8..\$15	Geral (não são preservados pelas funções)																																																	
\$s0..\$s7	\$16..\$23	Geral (não podem ser alterados pelas funções)																																																	
\$t8..\$t9	\$24..\$25	Geral (não são preservados pelas funções)																																																	
\$k0..\$k1	\$26..\$27	Reservado pelo <i>kernel</i> do S.O.																																																	
\$gp	\$28	Ponteiro para área global (<i>Global Pointer</i>)																																																	
\$sp	\$29	<i>Stack Pointer</i>																																																	
\$fp	\$30	<i>Frame Pointer</i>																																																	
\$ra	\$31	Endereço de retorno das funções (<i>Return Address</i>)																																																	
mtcz	Rsrc, Creg	ori	Rdst, Rsrc, Imm	Cálculo c/ Inteiros: Op. Lógicas Bitwise																																															
mov.d	FPdst, FPsrc	xor	Rdst, Rsrc1, Rsrc2	not																																															
mov.s	FPdst, FPsrc	xori	Rdst, Rsrc, Imm	rol																																															
Manipulação de Const. (<i>Load Immediate</i>)		Cálculo c/ Inteiros: Operações de Shift		ror																																															
lui	Rdst, Imm	sll	Rdst, Rsrc1, Imm5	seq																																															
Instruções de Comparação		sllv	Rdst, Rsrc1, Rsrc2	sge																																															
slt	Rdst, Rsrc1, Rsrc2	sra	Rdst, Rsrc1, Imm5	sgeu																																															
sltu	Rdst, Rsrc1, Rsrc2	srav	Rdst, Rsrc1, Rsrc2	sgt																																															
slti	Rdst, Rsrc, Imm	srl	Rdst, Rsrc1, Imm5	sgtu																																															
sltiu	Rdst, Rsrc, Imm	srlv	Rdst, Rsrc1, Rsrc2	sle																																															
Salto Relativo (<i>Branch</i>) e Absoluto (<i>Jump</i>)		Cálculo em Vírgula Flutuante		sleu																																															
bczf	Label	abs.p	FPdst, FPsrc	sne																																															
bczt	Label	add.p	FPdst, FPsrc1, FPsrc2	<table><tr><th colspan="2">Tabela II: Registos da FPU do MIPS e convenção de uso</th></tr><tr><th>Nome Lógico</th><th>Uso Convencionado</th></tr><tr><td>\$f0(\$f1) ... \$f2(\$f3)</td><td>Cálculo de expressões e valor de retorno das funções</td></tr><tr><td>\$f4(\$f5) ... \$f10(\$f11)</td><td>Geral (não são preservados pelas funções)</td></tr><tr><td>\$f12(\$f13) ... \$f14(\$f15)</td><td>Passagem de parâmetros para funções.</td></tr><tr><td>\$f16(\$f17) ... \$f18(\$f19)</td><td>Geral (não são preservados pelas funções)</td></tr><tr><td>\$f20(\$f21) ... \$f30(\$f31)</td><td>Geral (não podem ser alterados pelas funções)</td></tr></table>						Tabela II: Registos da FPU do MIPS e convenção de uso		Nome Lógico	Uso Convencionado	\$f0(\$f1) ... \$f2(\$f3)	Cálculo de expressões e valor de retorno das funções	\$f4(\$f5) ... \$f10(\$f11)	Geral (não são preservados pelas funções)	\$f12(\$f13) ... \$f14(\$f15)	Passagem de parâmetros para funções.	\$f16(\$f17) ... \$f18(\$f19)	Geral (não são preservados pelas funções)	\$f20(\$f21) ... \$f30(\$f31)	Geral (não podem ser alterados pelas funções)																												
Tabela II: Registos da FPU do MIPS e convenção de uso																																																			
Nome Lógico	Uso Convencionado																																																		
\$f0(\$f1) ... \$f2(\$f3)	Cálculo de expressões e valor de retorno das funções																																																		
\$f4(\$f5) ... \$f10(\$f11)	Geral (não são preservados pelas funções)																																																		
\$f12(\$f13) ... \$f14(\$f15)	Passagem de parâmetros para funções.																																																		
\$f16(\$f17) ... \$f18(\$f19)	Geral (não são preservados pelas funções)																																																		
\$f20(\$f21) ... \$f30(\$f31)	Geral (não podem ser alterados pelas funções)																																																		
beq	Rsrc1, Rsrc2, Label	c.eq.p	FPsrc1, FPsrc2																																																
bgez	Rsrc, Label	c.le.p	FPsrc1, FPsrc2																																																
bgezal	Rsrc, Label	c.lt.p	FPsrc1, FPsrc2																																																
bgtz	Rsrc, Label	cvt.d.s	FPdst, FPsrc																																																
blez	Rsrc, Label	cvt.d.w	FPdst, FPsrc																																																
bltz	Rsrc, Label	cvt.s.d	FPdst, FPsrc																																																
bltzal	Rsrc, Label	cvt.s.w	FPdst, FPsrc																																																
bne	Rsrc1, Rsrc2, Label	cvt.w.d	FPdst, FPsrc																																																
j	Label	cvt.w.s	FPdst, FPsrc																																																
jal	Label	div.p	FPdst, FPsrc1, FPsrc2																																																
jalr	Rsrc	mul.p	FPdst, FPsrc1, FPsrc2																																																
jr	Rsrc	neg.p	FPdst, FPsrc																																																
		sub.p	FPdst, FPsrc1, FPsrc2																																																
		Manipulação de Exceções e Traps																																																	
		break	n																																																
		nop																																																	
		eret																																																	
		syscall																																																	

Tabela III: Notação			
Imm	Valor imediato (constante) de 16 bits	addr	Endereço na forma Imm(Rsrc) = (Rsrc) + Imm
IMM	Valor imediato de 32 bits	B_k(Rsrc)	Byte índice k de Rsrc
Rsrc (1, 2)	Registo fonte (1 ou 2)	FPdst	Registo destino do coprocessador aritmético
(Rsrc)	Conteúdo de Rsrc	FPsrc (1, 2)	Registo fonte do coprocessador aritmético (1 ou 2)
Rdst	Registo destino	C_z	Coprocessador nº z
CReg	Registo do Coprocessador C_z	Src	Rsrc ou IMM
sym	Endereço do símbolo (label) sym	Imm5	Valor imediato (constante) de 5 bits

Tabela IV: <i>System Calls</i> do MARS			
Protótipo equivalent em C	\$v0	Parâmetros de entrada	Retorno
void print_int10(int value)	1	\$a0 = value (int)	
void print_float(float value)	2	\$f12 = value (float)	
void print_double(double value)	3	\$f12 = value (double)	
void print_string(char *str)	4	\$a0 = str	
int read_int(void)	5		\$v0
float read_float(void)	6		\$f0
double read_double(void)	7		\$f0
void read_string(char *buf, int length)	8	\$a0 = buf, \$a1 = length	
void *sbrk(int amount)	9	\$a0 = amount	\$v0
void exit(void)	10		
void _print_char(char value)	11	\$a0 = value (char)	
char read_char(void)	12		\$v0
void print_int16(unsigned int value)	34	\$a0 = value (unsigned int)	
void print_int2(unsigned int value)	35	\$a0 = value (unsigned int)	
void print_intu10(unsigned int value)	36	\$a0 = value (unsigned int)	

Tabela V - Directivas do Assembler	
Directivas	Descrição
Para controlo dos Segmentos	
.data [address]	Coloca os próximos itens no segmento de dados do utilizador (opcionalmente a partir de <i>address</i>).
.text [address]	Coloca os próximos itens no segmento de código do utilizador (opcionalmente a partir de <i>address</i>).
.kdata [address]	Coloca os próximos itens no segmento de dados do <i>kernel</i> (opcionalmente a partir de <i>address</i>).
.ktext [address]	Coloca os próximos itens no segmento de código do <i>kernel</i> (opcionalmente a partir de <i>address</i>).
Para criação de constantes e variáveis em memória:	
.ascii str	Armazena uma <i>string</i> em memória sem lhe acrescentar o terminador '\0'.
.eqv label, valor	Substitui todas as ocorrências de <i>label</i> no programa por <i>valor</i> .
.asciiz str	Armazena uma <i>string</i> em memória acrescentando-lhe o terminador '\0'.
.byte b ₁ , ..., b _n	Armazena as grandezas de 8 bits b ₁ , ..., b _n em sucessivos bytes de memória.
.half h ₁ , ..., h _n	Armazena as grandezas de 16 bits h ₁ , ..., h _n em sucessivas meias palavras de memória.
.word w ₁ , ..., w _n	Armazena as grandezas de 32 bits w ₁ , ..., w _n em sucessivas palavras de memória.
.float f ₁ , ..., f _n	Armazena f ₁ , ..., f _n em vírgula flutuante, precisão simples (32 bits) no seg. de dados.
.double d ₁ , ..., d _n	Armazena d ₁ , ..., d _n em vírgula flutuante, precisão dupla (64 bits) no seg. de dados.
.space n	Reserva <i>n</i> bytes no segmento de dados, sem inicializar
Para controlo do alinhamento:	
.align n	Alinha o próximo item num endereço múltiplo de 2 ⁿ .
Para referências externas:	
.globl sym	Declara que o símbolo <i>sym</i> é global e pode ser referenciado em outros ficheiros.
.extern sym size	Declara que o item associado a <i>sym</i> ocupa <i>size</i> bytes e é um símbolo global.