

Guía rápida del ensamblador del MIPS 32

Registros MIPS y usos

| Nombre del Registro | Número | Uso |
|---------------------|---------|--|
| zero | 0 | Constante 0 |
| at | 1 | Reservada para ensamblador |
| v0 | 2 | Evaluación de expresiones y resultado de funciones |
| v1 | 3 | Evaluación de expresiones y resultado de funciones |
| a0 | 4 | Argumento 1 |
| a1 | 5 | Argumento 2 |
| a2 | 6 | Argumento 3 |
| a3 | 7 | Argumento 4 |
| t0...t7 | 8...15 | Temporal (No se guarda valor entre llamadas) |
| s0...s7 | 16...23 | Temporal (Se guarda valor entre llamadas) |
| t8, t9 | 24, 25 | Temporal (No se guarda valor entre llamadas) |
| k0, k1 | 26, 27 | Reservado para el kernel del Sistema operativo |
| gp | 28 | Puntero al área global |
| sp | 29 | Puntero de pila |
| fp | 30 | Puntero de marco de pila |
| ra | 31 | Dirección de retorno, usada por llamadas a función |

Llamadas al sistema

| Servicio | Código de llamada | Argumentos | Resultado |
|--------------|-------------------|--------------------------------|-------------------------|
| print_int | 1 | \$a0 = entero | |
| print_float | 2 | \$f12 = real (32 bits) | |
| print_double | 3 | \$FP12 = real (64 bits) | |
| print_string | 4 | \$a0 = cadena | |
| read_int | 5 | | Entero (en \$v0) |
| read_float | 6 | | Real 32 bits (en \$f0) |
| read_double | 7 | | Real 64 bits (en \$FP0) |
| read_string | 8 | \$a0 = buffer, \$a1 = longitud | |
| sbrk | 9 | \$a0 = cantidad | Dirección (en \$v0) |
| exit | 10 | | |
| print_char | 11 | \$a0 = byte | |
| read_char | 12 | | \$v0 (cód ASCII) |

Directivas del ensamblador

| | |
|----------------------------|---|
| .ascii cadena | Almacena la cadena en memoria, pero no termina con NULL ('\0') |
| .asciiz cadena | Almacena la cadena en memoria y coloca un NULL ('\0') al final de esta |
| .byte b1, ..., bn | Almacena N valores en bytes sucesivos de memoria |
| .data | Las siguientes definiciones de datos que aparezcan se almacenan en el segmento de datos. Puede llevar un argumento que indica la dirección a partir de donde se empezarán a almacenar los datos. |
| .double d1, ..., dn | Almacena N valores reales de doble precisión en direcciones consecutivas de memoria. |
| .extern etiqueta n | Declara que los datos almacenados a partir de <i>etiqueta</i> ocupan N bytes y que <i>etiqueta</i> es un símbolo global. Esta directiva permite al ensamblador almacenar datos en una zona del segmento de datos que puede ser accedida a través del registro \$gp. |
| .float f1, ..., fn | Almacena N valores reales de precisión simple en posiciones consecutivas de memoria. |
| .globl simbolo | Declara un símbolo global que se puede referenciar desde otros programas. |
| .half h1, ..., hn | Almacena N números de 16 bits en medias palabras consecutivas. |
| .text | Las instrucciones que siguen a esta directiva se ponen en el segmento de código. Puede llevar un parámetro que indica dónde empieza la zona de código. |
| .word w1, ..., wn | Almacena N cantidades de 32 bits (1 palabra) en posiciones consecutivas de memoria |

Instrucciones aritméticas y lógicas

En todas las instrucciones siguientes, Src2 puede ser tanto un registro como un valor inmediato (un entero de 16 bits) y en aquellas donde pone inm solo acepta un valor inmediato

| | |
|--------------------------|--|
| add Rdest, Rsrc1, Src2 | Suma con desbordamiento |
| addi Rdest, Rsrc1, inm | Suma un número inmediato con desbordamiento |
| addu Rdest, Rsrc1, Src2 | Suma sin desbordamiento |
| addiu Rdest, Rsrc1, inm | Suma un número inmediato sin desbordamiento |
| and Rdest, Rsrc1, Src2 | Operación lógica AND |
| andi Rdest, Rsrc1, inm | Operación lógica AND con un número inmediato |
| div Rsrc1, Rsrc2 | Divide con desbordamiento. Deja el cociente en el registro <i>lo</i> y el resto en el registro <i>hi</i> |
| divu Rsrc1, Rsrc2 | Divide sin desbordamiento. Deja el cociente en el registro <i>lo</i> y el resto en el registro <i>hi</i> |
| div Rdest, Rsrc1, Rrc2 | Divide con desbordamiento |
| divu Rdest, Rsrc1, Rrc2 | Divide sin desbordamiento |
| mul Rdest, Rsrc1, Src2 | Multiplica sin desbordamiento |
| mult Rsrc1, Rsrc2 | Multiplica, la parte baja del resultado se deja en el registro <i>lo</i> y la parte alta en el registro <i>hi</i> |
| multu Rsrc1, Rsrc2 | Multiplica sin desbordamiento, la parte baja del resultado se deja en el registro <i>lo</i> y la parte alta en el registro <i>hi</i> |
| mod Rdest, Rsrc1, Rsrc2 | Módulo de la división con desbordamiento |
| modu Rdest, Rsrc1, Rsrc2 | Módulo de la división sin desbordamiento |
| nop | No realiza ninguna operación |
| nor Rdest, Rsrc1, Src2 | Operación Lógica NOR |
| or Rdest, Rsrc1, Src2 | Operación Lógica OR |
| ori Rdest, Rsrc1, inm | Operación Lógica OR con un inmediato |
| rem Rdest, Rsrc1, Rsrc2 | Módulo de la división con desbordamiento |
| rotr rdest, rsrc1, inm | Rotación hacia la derecha de inm bits |
| sll Rdest, Rsrc1, inm | Desplazamiento lógico de bits a la izquierda |

| | |
|-------------------------|--|
| srl Rdest, Rsrc1, inm | Desplazamiento lógico de bits a la derecha |
| sra Rdest, Rsrc1, inm | Desplazamiento aritmético de bits a la derecha |
| sub Rdest, Rsrc1, Src2 | Resta (con desbordamiento) |
| subu Rdest, Rsrc1, Src2 | Resta (sin desbordamiento) |
| xor Rdest, Rsrc1, Src2 | Operación Lógica XOR |
| xori Rdest, Rsrc1, inm | Operación Lógica XOR con número inmediato |

Instrucciones de manipulación de constantes

| | |
|----------------------|---|
| li Rdest, inmediato | Cargar valor inmediato |
| lui Rdest, inmediato | Cargar los 16 bits de la parte baja del valor inmediato en la parte alta del registro. Los bits de la parte baja se pone a 0. |

Instrucciones de comparación

En todas las instrucciones siguientes, Src2 puede ser un registro o un valor inmediato (de 16 bits).

| | |
|--------------------------|---|
| slt Rdest, Rsrc1, Rsrc2 | Pone Rdest a 1 si Rsrc1 es menor a Rsrc2, en otro caso pone 0 (para números con signo). |
| sltu Rdest, Rsrc1, Rsrc2 | Pone Rdest a 1 si Rsrc1 es menor a Rsrc2, en otro caso pone 0 (para números sin signo). |
| slti Rdest, Rsrc1, inm | Pone Rdest a 1 si Rsrc1 es menor a inm, en otro caso pone 0 (para números con signo). |
| sltiu Rdest, Rsrc1, inm | Pone Rdest a 1 si Rsrc1 es menor a inm, en otro caso pone 0 (para números sin signo). |

Instrucciones de almacenamiento

| | |
|--------------------|--|
| sb Rsrc, dirección | Almacena el byte más bajo de Rsrc en la dirección indicada. |
| sh Rsrc, dirección | Almacena la media palabra (16 bits) baja de un registro en la dirección de memoria indicada. |
| sw Rsrc, dirección | Almacena la Rsrc en la dirección indicada. |

Instrucciones de bifurcación y salto

En todas las instrucciones siguientes, Src2 puede ser un registro o un valor inmediato. Las instrucciones de bifurcación (branch) usan un desplazamiento de 16 bits con signo; por lo que se puede saltar 2^{15-1} instrucciones hacia delante o 2^{15} instrucciones hacia atrás. Las instrucciones de salto (jump) contienen un campo de dirección de 26 bits.

| | |
|----------------------------|--|
| b etiqueta | Bif. incondicional a la instrucción que está en etiqueta. |
| beq Rsrc1, Src2, etiqueta | Bif. condicional si Rsrc1 es igual a Src2. |
| beqz Rsrc, etiqueta | Bif. condicional si el registro Rsrc es igual a 0. |
| bge Rsrc1, Src2, etiqueta | Bif. condicional si el registro Rsrc1 es mayor o igual a Src2 (con signo). |
| bgeu Rsrc1, Src2, etiq | Bif. condicional si el registro Rsrc1 es mayor o igual a Src2 (sin signo). |
| bgez Rsrc, etiqueta | Bif. condicional si el registro Rsrc es mayor o igual a 0. |
| bgezal Rsrc, etiqueta | Bif. condicional si el registro Rsrc es mayor o igual a 0. Guarda la dirección actual en el registro \$ra (\$31) |
| bgt Rsrc1, Src2, etiqueta | Bif. condicional si el registro Rsrc1 es mayor que Src2 (con signo). |
| bgtu Rsrc1, Src2, etiqueta | Bif. condicional si el registro Rsrc1 es mayor que Src2 (sin signo). |
| bgtz Rsrc, etiqueta | Bif. condicional si Rsrc es mayor que 0. |
| ble Rsrc1, Src2, etiqueta | Bif. condicional si Rsrc1 es menor o igual a Src2 (con signo). |
| bleu Rsrc1, Src2, etiqueta | Bif. condicional si Rsrc1 es menor o igual a Src2 (sin signo). |
| blez Rsrc, etiqueta | Bif. condicional si Rsrc es menor o igual a 0. |
| blt Rsrc1, Src2, etiqueta | Bif. condicional si Rsrc1 es menor que Src2 (con signo). |
| bltu Rsrc1, Src2, etiqueta | Bif. condicional si Rsrc1 es menor que Src2 (sin signo). |
| bltz Rsrc, etiqueta | Bif. condicional si Rsrc es menor que 0. |
| bne Rsrc1, Src2, etiqueta | Bif. condicional si Rsrc1 no es igual a Src2. |
| bnez Rsrc, etiqueta | Bif. condicional si Rsrc no es igual a 0. |
| j etiqueta | Salto incondicional. |
| jal etiqueta | Salto incondicional, almacena la dirección actual en \$ra (\$31). |
| jalr Rsrc | Salto incondicional, almacena la dirección actual en \$ra (\$31). |
| jalr Rsrc1, Rsrc2 | Salto incondicional, almacena la dirección actual en Rsrc1. |
| jr Rsrc | Salto incondicional. |

Instrucciones de carga

| | |
|----------------------|---|
| la Rdest, dirección | Carga dirección en Rdest (el valor de dirección, no el contenido) |
| lb Rdest, dirección | Carga el byte de la dirección especificada y extiende el signo |
| lbu Rdest, dirección | Carga el byte de la dirección especificada, no extiende el signo |
| lh Rdest, dirección | Carga 16 bits de la dirección especificada, se extiende el signo |
| lhu Rdest, dirección | Carga 16 bits de la dirección especificada, no se extiende signo |
| lw Rdest, dirección | Carga una palabra de la dirección especificada. |

Instrucciones de transferencia de datos

| | |
|------------------|---|
| move Rdest, Rsrc | Mueve el contenido del registro Rsrc al registro Rdest. |
| mfhi Rdest | Mueve el contenido del registro HI al registro Rdest. |
| mflo Rdest | Mueve el contenido del registro LO al registro Rdest. |
| mthi Rsrc | Mueve el contenido del registro Rsrc al registro HI. |
| mtlo Rsrc | Mueve el contenido del registro Rsrc al registro LO. |

Instrucciones aritméticas del coprocesador de coma flotante

| | |
|------------------|--|
| abs.s fd, fs, ft | Valor absoluto de un número real de 32 bits. |
| abs.d fd, fs, ft | Valor absoluto de un número real de 64 bits. |
| add.s fd, fs, ft | Suma los registros fs y ft y almacena el resultado en fd (float) |
| add.d fd, fs, ft | Suma los registros fs y ft y almacena el resultado en fd (double) |
| div.s fd, fs, ft | Divide fs entre ft y deja el resultado en fd (float) |
| div.d fd, fs, ft | Divide fs entre ft y deja el resultado en fd (double) |
| mul.s fd, fs, ft | Multiplica los registros fs y ft y deja su resultado en fd. (float) |
| mul.d fd, fs, ft | Multiplica los registros fs y ft y deja su resultado en fd. (double) |
| rsqrt.s fd, fs | 1.0 entre el resultado de la raíz cuadrada de fs (float) |
| rsqrt.d fd, fs | 1.0 entre el resultado de la raíz cuadrada de fs (double) |
| sqr.s fd, fs | Raíz cuadrada de fs (float) |
| sqr.d fd, fs | Raíz cuadrada de fs (double) |
| sub.s fd, fs, ft | Resta (float) |
| sub.d fd, fs, ft | Resta (double) |

Instrucciones de carga y almacenamiento del coprocesador de coma flotante

| | |
|-------------------|--|
| l.s fs, dirección | Carga en fs el valor del float (32 bits) que se encuentra a partir de la dirección especificada. |
| l.d fd, dirección | Carga fd con el valor del double (64 bits) que se encuentra a partir de la dirección especificada. |
| s.s fs, dirección | Almacena el registro fs a partir de la dirección indicada. (float) |
| s.d fd, dirección | Almacena un double (64 bits) en la dirección indicada, el valor de 64 bits proviene de fd. |
| li.s fs, valor | Carga en el registro fs del coprocesador matemático el valor (float) |
| li.d fd, valor | Carga en el registro fd del coprocesador matemático el valor (double) |

El campo dirección se puede representar utilizando direccionamiento absoluto, indirecto de registro o relativo a registro

Instrucciones de transferencia de datos entre registros

| | |
|-------------------|--|
| mfc1 Rdest, CPsrc | Mueve el contenido del registro CPsrc del coprocesador en coma flotante al registro de la CPU Rdest. |
| mtc1 Rsrc, CPdest | Mueve el contenido del registro Rsrc de la CPU al registro CPdest del coprocesador en coma flotante. |
| mov.s fd, fs | Mueve el contenido del registro fs al registro fd. (float) |
| mov.d fd, fs | Mueve el contenido del registro fs al registro fd. (double) |

Instrucciones de conversión

| | |
|-------------------|---|
| cvt.d.s fd, fs | Convierte un float en un double, el resultado se guarda en fd |
| cvt.d.w fd, Rsrc | Convierte un entero en un double, el resultado se guarda en fd |
| cvt.s.d fd, fs | Convierte un double en un float, el resultado se guarda en fd |
| cvt.s.w fd, Rsrc | Convierte un entero en un float, el resultado se guarda en fd |
| cvt.w.s Rdest, fs | Convierte un float en un entero, el resultado se guarda en Rdest |
| cvt.w.d Rdest, fs | Convierte un double en un entero, el resultado se guarda en Rdest |