

Registros

Nombre	Número	Uso
\$zero	0	Constante 0
\$at	1	Reservado para el ensamblador
\$v0 - \$v1	2 - 3	Resultados de funciones
\$a0 - \$a3	4 - 7	Argumentos
\$t0 - \$t7	8 - 15	Temporales (caller)
\$s0 - \$s7	16 - 23	Guardados temporalmente (callee)
\$t8 - \$t9	24 - 25	Temporales (caller)
\$k0 - \$k1	26 - 27	Reservado para el kernel del SO
\$gp	28	Global Pointer
\$sp	29	Stack Pointer
\$fp	30	Frame Pointer
\$ra	31	Return Address

Directivas

.data <addr>

Los siguientes items son guardados en un segmento de memoria.

.text <addr>

Lo siguientes items son guardados en un segmento de texto.

.byte b1, b2, ..., bn
.half h1, h2, ..., hn
.word w1, w2, ..., wn
.float f1, f2, ..., fn
.double f1, f2, ..., fn
.ascii str
.asciiz str (null terminated)

Guardan el tipo de dato indicado de forma sucesiva en la memoria.

.globl sym Indica que la etiqueta sym es global y puede ser referenciada por otros archivos.

Instrucciones Aritméticas

add rd, rs, rt rd=rs+rt
addi rd, rs, imm rd=rs+imm

sub rd, rs, rt rd=rs+rt

div rs, rt rs/rt
 El cociente se guarda en el registro lo y el residuo en el registro hi

mul rd, rs, rt rd=rs*rt
mult rs, rt rs*rt

Los bits de mayor orden del resultado se guardan en el registro hi y los de menor orden en el registro lo

Instrucciones Lógicas

and rd, rs, rt rd=rs **AND** rt
andi rd, rs, imm rd=rs **AND** imm

nor rd, rs, rt rd=rs **NOR** rt

or rd, rs, rt rd=rs **OR** rt
ori rd, rs, imm rd=rs **OR** imm

xor rd, rs, rt rd=rs **XOR** rt
xori rd, rs, imm rd=rs **XOR** imm

Instrucciones de Comparación

slt rd, rs, rt rd=rs < rt
slti rd, rs, imm rd=rs < imm

Instrucciones Branch

beq rs, rt, label Salto si rs==rt
bne rs, rt, label Salto si rs!=rt

bgez rs, label Salto si 0≤rs
bgtz rs, label Salto si 0<rs

blez rs, label Salto si rs≤0
bltz rs, label Salto si rs<0

Instrucciones de Salto

j target Salto a target
jal target Salto y guarda \$ra
jr rs Salto a \$rs

Instrucciones Load

lb rt, address
lh rt, address
lw rt, address
la rt, address Carga la dirección, no el contenido.

Instrucciones Store

sb rt, address
sh rt, address
sw rt, address

Instrucciones Trap

teq rs, rt Trap si rs==rt
teqi rt, imm Trap si rs==imm
tge rt, rt Trap si rs>=rt
tlr rs, rt Trap si rs<rt

Llamadas a procedimientos

Caller:

1. Pasar los argumentos. Por convención los primeros 4 argumentos se pasan en los registros \$a0-\$a3. Los demás argumentos se pasan en la pila.

2. Guardar los registros temporales. Si el caller va a usar de nuevo la información contenida en \$v0-\$v1 y \$t0-\$t9 debe guardarlo en la pila.

3. Ejecutar la instrucción jal para brincar a la subrutina.

Callee:

1. Reservar el espacio de memoria para el frame restando el tamaño del frame al stack pointer.

2. Guardar los registros guardados temporalmente en la pila. Se deben guardar los registros \$s0-\$s7 \$fp y \$ra antes de modificarlos, ya que el caller los espera intactos. Si el calle no los va a usar no es necesario.

3. Establecer el frame pointer sumando el tamaño de la pila menos cuatro al \$sp y guardando la suma en el \$fp.

Al terminar Calle:

1. Si retorna un valor ponerlo en el registro \$v0.

2. Restaurar los registros guardados por

el callee (\$s0-\$s7 \$fp y \$ra).

3. Pop el stack frame sumando el tamaño del frame a \$sp.

4. Regresar al caller brincado a la dirección guardada en el registro \$ra.

Syscalls

Para solicitar un servicio, el programa carga el código de la llamada en el registro \$v0, carga los parámetros en los registros \$a0-\$a3 y ejecuta la instrucción syscall.

Servicio	Code	Uso
print_int	1	Imprime el entero guardado en \$a0.
print_string	4	Imprime la cadena que comienza en la dirección guardada en \$a0.
read_int	5	Lee el entero y lo guarda en \$v0.
read_string	8	Lee una cadena de longitud indicada en \$a1 y la guarda en memoria comenzando en la dirección indicada en \$a0.
exit	10	Termina el programa.

Excepciones e interrupciones

Registros de Coproc0 :

Número	Nombre	Uso
\$8	vaddr	Dirección de la memoria errónea que se intentó acceder.
\$12	status	Interrupt mask y enable bits.
\$13	cause	Tipo de interrupción y bits de interrupciones pendientes.
\$14	epc	Dirección de memoria que causo la excepción.

.kdata <addr>

Datos subsecuentes que se guardan en el segmento de datos del kernel, no en el espacio de usuario.

.ktext <addr> (0x80000180)

Instrucciones guardadas en el segmento de texto de kernel.

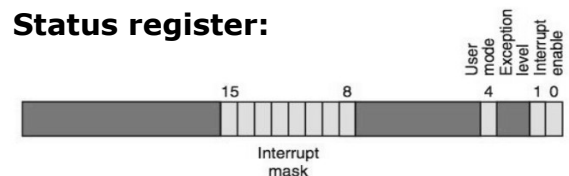
eret Cambia el registro de estado a 0 y regresa a la instrucción apuntada por el registro EPC.

mfc0 \$rd, \$rt

mtc0 \$rt, \$rd

Número	Nombre	Causa de la excepción
0	Int	Interrupción de hardware.
4	AdEL	Excepción de error de dirección (load).
5	AdES	Excepción de error de dirección (store).
8	Sys	Excepción de una llamada al sistema.
12	Ov	Excepción por overflow aritmético.
13	trap	Trap.

Status register:



Cause register:

