Name	Use
$zero	Holds constant value of 0
$a0-$a3	Argument values
$v0-$v1	Function results
$t0-$t9	Temporary values
$s0-$s7	Saved values
$gp	Global pointer
$sp	Stack pointer
$fp	Frame pointer
$ra	Return address

Table 1: MIPS Registers

Instruction	Description
add rd, rs, rt	store rs + rt in rd
addi rd, rs, imm	store rs+imm into rd
sub rd, rs, rt	store rs - rt in rd
mul rd, rs, rt	store rs * rt in rd
div rd, rs, rt	store rs / rt in rd
neg rd, rs	store -rs in rd
seq rd, rs, rt	store 1 in rd if rs equals rt, 0 otherwise
sne rd, rs, rt	store 1 in rd if rs not equal rt, 0 otherwise
slt rd, rs, rt	store 1 in rd if rs less than rt, 0 otherwise
sgt rd, rs, rt	store 1 in rd if rs greather than rt, 0 otherwise
sle rd, rs, rt	store 1 in rd if rs less than or equal rt, 0 otherwise
sge rd, rs, rt	store 1 in rd if rs greater than or equal rt, 0 otherwise
sll rd, rs, amt	store value from rs shifted left by amt into rd
beqz rd, label	conditional branch to label if rd equal 0
bgtz rd, label	conditional branch to label if rd greater than 0
b label	unconditional branch to label
li rd, constant	load the constant into rd
la rd, address	load computed address (not contents) into rd
lw rd, address	load word from address into rd
lb rd, address	load byte from address into rd
sw rd, address	store word from rd into address
sb rd, address	store byte from rd into address
jal label	jump and link to label
jr rd	jump to address in rd
move rd, rs	move value in rs to rd

Table 2: MIPS Instructions

Service	Code
print integer	1
print string	4
read integer	5

Table 3: MIPS System Calls. Put the argument in register $a0 and the code into register $v0 prior to doing `syscall`.