SPIM System Service Calls

The SPIM simulator uses the concept of system call ("syscall") to request a service.
A service represents specific SPIM tasks, such as:

1. Print variable of type integer
2. Read an integer value
3. Read a float value
4. Read a character value
5. Read a string value
6. Print a variable of type float
7. Print a variable of type string
8. Print a variable of type character

SPIM expects a unique code for each value type.

Rules to Request a Service

1. Your program must first load the appropriate system call code into register **$v0 (**or **$f0** for floats and double**)** using the **li** command [See table below or Page A-43 to A-45]
2. Next, the program should load the arguments into registers $a0 - $a3 (or $f12 for floating-point values) [Use **li** for integers and float; and **la** for strings]
3. Finally call "**syscall**" for each service request

Example MIPS Code to print: "Answer = 5"

```
        .data
str:
        .asciiz   "the answer ="
        .text
        li        $v0,  4             #       System call code for print string
        la        $a0,  str           #       Address of string  print
        syscall                       #       Print the string
        li        $v0,  1             #       System call code for print integer
        la        $a0,  5             #       Integer to  print
        syscall                       #       Print the integer
```

SPIM System Services

Service	System Call Code	Arguments	Result	
Print Integer	1	$a0		
Print float	2	$f12		
Print double	3	$f12		
Print String	4	$a0		
Read Integer	5		integer (in $v0)	
Read float	6		float (in $f0)	
Read double	7		double (in $f0)	
Read string	8	$a0 = buffer, $a1 = length		
Print char	11	$a0	char (in $v0)	
Read Char	12			