

SYSCALL functions available in MARS

Introduction

A number of system services, mainly for input and output, are available for use by your MIPS program. They are described in the table below.

MIPS register contents are not affected by a system call, except for result registers as specified in the table below.

How to use SYSCALL system services

Step 1. Load the service number in register \$v0.

Step 2. Load argument values, if any, in \$a0, \$a1, \$a2, or \$f12 as specified.

Step 3. Issue the SYSCALL instruction.

Step 4. Retrieve return values, if any, from result registers as specified.

Example: display the value stored in \$t0 on the console

```
li $v0, 1           # service 1 is print integer
add $a0, $t0, $zero # load desired value into
argument register $a0, using pseudo-op
syscall
```

| Service | Code in \$v0 | Arguments | Result |
|---------------|--------------|---|-----------------------------|
| print integer | 1 | \$a0 = integer to print | |
| print float | 2 | \$f12 = float to print | |
| print double | 3 | \$f12 = double to print | |
| print string | 4 | \$a0 = address of null-terminated string to print | |
| read integer | 5 | | \$v0 contains integer read |
| read float | 6 | | \$f0 contains float read |
| read double | 7 | | \$f0 contains double read |
| read string | 8 | \$a0 = address of input buffer \$a1 = maximum number of characters to read | <i>See note below table</i> |