

# LUA - OPERATING SYSTEM FACILITIES

[http://www.tutorialspoint.com/lua/lua\\_standard\\_libraries\\_operating\\_system\\_facilities.htm](http://www.tutorialspoint.com/lua/lua_standard_libraries_operating_system_facilities.htm)

Copyright © tutorialspoint.com

[Previous Page](#)

[Next Page](#)

In any application, it is often required for to access Operating System level functions and it is made available with Operating System library. The list of functions available are listed in the following table.

S.N.	Library / Method & Purpose
1.	<b>os.clock</b> Returns an approximation of the amount in seconds of CPU time used by the program.
2.	<b>os.date</b> [ <i>format</i> [, <i>time</i> ]] Returns a string or a table containing date and time, formatted according to the given string format.
3.	<b>os.difftime</b> <i>t2</i> , <i>t1</i> Returns the number of seconds from time <i>t1</i> to time <i>t2</i> . In POSIX, Windows, and some other systems, this value is exactly <i>t2-t1</i> .
4.	<b>os.execute</b> [ <i>command</i> ] This function is equivalent to the ANSI C function <code>system</code> . It passes <i>command</i> to be executed by an operating system shell. Its first result is true if the command terminated successfully, or nil otherwise.
5.	<b>os.exit</b> [ <i>code</i> [, <i>close</i> ]] Calls the ANSI C function <code>exit</code> to terminate the host program. If <i>code</i> is true, the returned status is <code>EXIT_SUCCESS</code> ; if <i>code</i> is false, the returned status is <code>EXIT_FAILURE</code> ; if <i>code</i> is a number, the returned status is this number.
6.	<b>os.getenv</b> <i>varname</i> Returns the value of the process environment variable <i>varname</i> , or nil if the variable is not defined.
7.	<b>os.remove</b> <i>filename</i> Deletes the file <i>filename</i> , on POSIX systems with the given name. If this function fails, it returns nil, plus a string describing the error and the error code.
8.	<b>os.rename</b> <i>oldname</i> , <i>newname</i> Renames file or directory named <i>oldname</i> to <i>newname</i> . If this function fails, it returns nil, plus a string describing the error and the error code.

9.	<b>os.setlocale</b> <i>locale</i> [, <i>category</i> ]  Sets the current locale of the program. <i>locale</i> is a system-dependent string specifying a locale; <i>category</i> is an optional string describing which category to change: "all", "collate", "ctype", "monetary", "numeric", or "time"; the default category is "all". The function returns the name of the new locale, or nil if the request cannot be honored.
10.	<b>os.time</b> [ <i>table</i> ]  Returns the current time when called without arguments, or a time representing the date and time specified by the given table. This table must have fields year, month, and day, and may have fields hour <i>defaultis</i> 12, min <i>defaultis</i> 0, sec <i>defaultis</i> 0, and isdst <i>defaultis</i> nil. For a description of these fields, see the os.date function.
11.	<b>os.tmpname</b>  Returns a string with a file name that can be used for a temporary file. The file must be explicitly opened before its use and explicitly removed when no longer needed.

## Common OS functions

A simple example using common math functions is shown below.

```
-- Date with format
io.write("The date is ", os.date("%m/%d/%Y"), "\n")

-- Date and time
io.write("The date and time is ", os.date(), "\n")

-- Time
io.write("The OS time is ", os.time(), "\n")

-- Wait for some time
for i=1,1000000 do
end

-- Time since Lua started
io.write("Lua started before ", os.clock(), "\n")
```

When we run the above program, we will get similar output to the following.

```
The date is 01/25/2014
The date and time is 01/25/14 07:38:40
The OS time is 1390615720
Lua started before 0.013
```

The above examples are just a few of the common examples, we can use OS library based on our need, so try using all the functions to be more familiar. There are functions like `remove` which helps in removing file, `execute` that helps us executing OS commands as explained above.