Os Library Tutorial

os.clock()

Return CPU time since Lua started in seconds.

```
> = os.clock()
11056.989
```

os.date([format [, time]])

Return formatted date string, or table of time information. The format string has the same format as the C `strftime()` function.

TODO: insert info, examples and links to strftime()

If the format string is `"*t"` a table is returned containing the time information, e.g,

```
> table.foreach(os.date('*t'), print)
hour    14
min     36
wday    1
year    2003
yday    124
month   5
sec     33
day     4
isdst   true
```

If the format is preceeded by "!" the time is converted to Coordinated Universal Time, e.g.,

```
> table.foreach(os.date('!*t'), print)
hour    21
min     36
wday    1
year    2003
yday    124
```

```
month   5
sec     42
day     4
isdst   false
```

os.difftime(t2, t1)

Calculate the number of seconds between time t1 to time t2.

```
> t1 = os.time()
> -- wait a little while then type....
> = os.difftime(os.time(), t1)
31
> = os.difftime(os.time(), t1)
38
```

os.execute(command)

Execute an operating system shell command. This is like the C `system()` function. The system dependent status code is returned.

```
> = os.execute("echo hello")
hello
0
> = os.execute("mmmmm")  -- generate an error
'mmmmm' is not recognized as an internal or external command,
operable program or batch file.
1
```

os.exit([code])

Calls the C function exit, with an optional code, to terminate the host program. The default value for code is the success code.

```
> os.exit(0)   -- kill the Lua shell we are in and pass 0 back to parent shell
```

os.getenv(varname)

Returns the value of the process environment variable varname, or nil if the variable is not defined.

```
> = os.getenv("BANANA")
nil
```

```
> = os.getenv("USERNAME")
Nick
```

os.remove(filename)

Deletes the file with the given name. If this function fails, it returns nil, plus a string describing the error.

```
> os.execute("echo hello > banana.txt")
> = os.remove("banana.txt")
true
> = os.remove("banana.txt")
nil     banana.txt: No such file or directory   2
```

os.rename(oldname, newname)

Renames file named oldname to newname. If this function fails, it returns nil, plus a string describing the error.

```
> os.execute("echo hello > banana.txt")
> = os.rename("banana.txt", "apple.txt")
true
> = os.rename("banana.txt", "apple.txt")
nil     banana.txt: No such file or directory   2
```

os.setlocale(locale [, category])

TODO: Describe: Sets the current locale of the program. locale is a string specifying a locale; category 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.

os.time([table])

Given a formatted date table, as used by `os.date()` return the time in system seconds.

```
> t = os.date('*t')  -- time now
> table.foreach(os.date('*t'), print)
hour    15
min     1
wday    1
year    2003
yday    124
```

```
month   5
sec     2
day     4
isdst   true
> = os.time(t)       -- time in system seconds
1052085659
> t.year = 2001      -- 2001, a Lua odyssey
> = os.time(t)       -- time then
989013659
```

os.tmpname ()

Generate a name that can be used for a temporary file. This only generates a name, it does not open a file.

```
> = os.tmpname()  -- on windows
\s2js.
```

FindPage · RecentChanges · preferences
edit · history
Last edited May 4, 2003 3:06 pm PDT (diff)