std::system

Defined in header <cstdlib>

```
int system( const char* command );
```

Calls the host environment's command processor (e.g. /bin/sh, cmd.exe, command.com) with the parameter command. Returns an implementation-defined value (usually the value that the invoked program returns).

If command is a null pointer, checks if the host environment has a command processor and returns a nonzero value if and only if the command processor exists.

Parameters

command – character string identifying the command to be run in the command processor. If a null pointer is given, command processor is checked for existence

Return value

Implementation-defined value. If command is a null pointer, returns a nonzero value if and only if the command processor exists.

Notes

On POSIX systems, the return value can be decomposed using WEXITSTATUS and WSTOPSIG (http://pubs.opengroup.org/onlinepubs/9699919799/functions/wait.html)

The related POSIX function popen (http://pubs.opengroup.org/onlinepubs/9699919799/functions/popen.html) makes the output generated by command available to the caller.

An explicit flush of std::cout is also necessary before a call to **std::system**, if the spawned process performs any screen I/O.

Example

Run this code

```
#include <cstdlib>
#include <fstream>
#include <iostream>

int main()
{
    std::system("ls -l >test.txt"); // execute the UNIX command "ls -l >test.txt"
    std::cout << std::ifstream("test.txt").rdbuf();
}
```

Possible output:

```
total 16
-rwxr-xr-x 1 2001 2000 8859 Sep 30 20:52 a.out
-rw-rw-rw- 1 2001 2000  161 Sep 30 20:52 main.cpp
-rw-r--r-- 1 2001 2000    0 Sep 30 20:52 test.txt
```

See also

C documentation for **system**

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=cpp/utility/program/system&oldid=121621"