# std::getenv

Defined in header <cstdlib>

```
char* getenv( const char* env_var );
```

Searches the *environment list* provided by the host environment (the OS), for a string that matches the C string pointed to by env_var and returns a pointer to the C string that is associated with the matched environment list member.

| | |
|---|---|
| This function is not required to be thread-safe. Another call to getenv, as well as a call to the POSIX functions setenv() (http://pubs.opengroup.org/onlinepubs/9699919799/functions/setenv.html) , unsetenv() (http://pubs.opengroup.org/onlinepubs/9699919799/functions/unsetenv.html) , and putenv() (http://pubs.opengroup.org/onlinepubs/9699919799/functions/putenv.html) may invalidate the pointer returned by a previous call or modify the string obtained from a previous call. | (until C++11) |
| This function is thread-safe (calling it from multiple threads does not introduce a data race) as long as no other function modifies the host environment. In particular, the POSIX functions setenv() (http://pubs.opengroup.org/onlinepubs/9699919799/functions/setenv.html) , unsetenv() (http://pubs.opengroup.org/onlinepubs/9699919799/functions/unsetenv.html) , and putenv() (http://pubs.opengroup.org/onlinepubs/9699919799/functions/putenv.html) would introduce a data race if called without synchronization. | (since C++11) |

Modifying the string returned by getenv invokes undefined behavior.

## Parameters

**env_var** - null-terminated character string identifying the name of the environmental variable to look for

## Return value

Character string identifying the value of the environmental variable or null pointer if such variable is not found.

## Notes

On POSIX systems, the environment variables (http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap08.html#tag_08) are also accessible through the global variable environ, declared as `extern char **environ;` in <unistd.h>, and through the optional third argument, envp, of the main function.

## Example

Run this code

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

int main()
{
    if(const char* env_p = std::getenv("PATH"))
        std::cout << "Your PATH is: " << env_p << '\n';
}
```

Possible output:

```
Your PATH is: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

## See also

**C documentation** for **getenv**