



## **Red Hat Enterprise Linux Release 9.2 Manual Pages on 'pthread\_barrier\_wait.3p' command**

**\$ man pthread\_barrier\_wait.3p**

PTHREAD\_BARRIER\_WAIT(3P) POSIX Programmer's Manual PTHREAD\_BARRIER\_WAIT(3P)

### PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

### NAME

pthread\_barrier\_wait ? synchronize at a barrier

### SYNOPSIS

```
#include <pthread.h>

int pthread_barrier_wait(pthread_barrier_t *barrier);
```

### DESCRIPTION

The pthread\_barrier\_wait() function shall synchronize participating threads at the barrier referenced by barrier. The calling thread shall block until the required number of threads have called pthread\_barrier\_wait() specifying the barrier.

When the required number of threads have called pthread\_barrier\_wait() specifying the barrier, the constant PTHREAD\_BARRIER\_SERIAL\_THREAD shall be returned to one unspecified thread and zero shall be returned to each of the remaining threads. At this point, the barrier shall be reset to the state it had as a result of the most recent pthread\_barrier\_init() function that referenced it.

The constant PTHREAD\_BARRIER\_SERIAL\_THREAD is defined in <pthread.h>

and its value shall be distinct from any other value returned by `pthread_barrier_wait()`.

The results are undefined if this function is called with an uninitialized barrier.

If a signal is delivered to a thread blocked on a barrier, upon return from the signal handler the thread shall resume waiting at the barrier if the barrier wait has not completed (that is, if the required number of threads have not arrived at the barrier during the execution of the signal handler); otherwise, the thread shall continue as normal from the completed barrier wait. Until the thread in the signal handler returns from it, it is unspecified whether other threads may proceed past the barrier once they have all reached it.

A thread that has blocked on a barrier shall not prevent any unblocked thread that is eligible to use the same processing resources from eventually making forward progress in its execution. Eligibility for processing resources shall be determined by the scheduling policy.

## RETURN VALUE

Upon successful completion, the `pthread_barrier_wait()` function shall return `PTHREAD_BARRIER_SERIAL_THREAD` for a single (arbitrary) thread synchronized at the barrier and zero for each of the other threads.

Otherwise, an error number shall be returned to indicate the error.

## ERRORS

This function shall not return an error code of `[EINTR]`.

The following sections are informative.

## EXAMPLES

None.

## APPLICATION USAGE

Applications using this function may be subject to priority inversion, as discussed in the Base Definitions volume of POSIX.1-2017, Section 3.291, Priority Inversion.

## RATIONALE

If an implementation detects that the value specified by the barrier argument to `pthread_barrier_wait()` does not refer to an initialized

barrier object, it is recommended that the function should fail and re-  
port an [EINVAL] error.

#### FUTURE DIRECTIONS

None.

#### SEE ALSO

`pthread_barrier_destroy()`

The Base Definitions volume of POSIX.1-2017, Section 3.291, Priority  
Inversion, Section 4.12, Memory Synchronization, `<pthread.h>`

#### COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form  
from IEEE Std 1003.1-2017, Standard for Information Technology -- Por-  
table Operating System Interface (POSIX), The Open Group Base Specifi-  
cations Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of  
Electrical and Electronics Engineers, Inc and The Open Group. In the  
event of any discrepancy between this version and the original IEEE and  
The Open Group Standard, the original IEEE and The Open Group Standard  
is the referee document. The original Standard can be obtained online  
at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are  
most likely to have been introduced during the conversion of the source  
files to man page format. To report such errors, see [https://www.ker-  
nel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .

IEEE/The Open Group            2017            PTHREAD\_BARRIER\_WAIT(3P)