Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'posix_trace_create.3p' command

$ man posix_trace_create.3p

POSIX_TRACE_CREATE(3P) POSIX Programmer's Manual POSIX_TRACE_CREATE(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

posix_trace_create, posix_trace_create_withlog, posix_trace_flush, posix_trace_shutdown ? trace stream initialization, flush, and shutdown from a process (TRACING)

SYNOPSIS

```
#include <sys/types.h>
#include <trace.h>
int posix_trace_create(pid_t pid,
    const trace_attr_t *restrict attr,
    trace_id_t *restrict trid);
int posix_trace_create_withlog(pid_t pid,
    const trace_attr_t *restrict attr, int file_desc,
    trace_id_t *restrict trid);
int posix_trace_flush(trace_id_t trid);
int posix_trace_shutdown(trace_id_t trid);
```

DESCRIPTION

The posix_trace_create() function shall create an active trace stream.

It allocates all the resources needed by the trace stream being created for tracing the process specified by pid in accordance with the attr argument. The attr argument represents the initial attributes of the trace stream and shall have been initialized by the function posix_trace_attr_init() prior to the posix_trace_create() call. If the argument attr is NULL, the default attributes shall be used. The attr attributes object shall be manipulated through a set of functions de? scribed in the posix_trace_attr family of functions. If the attributes of the object pointed to by attr are modified later, the attributes of the trace stream shall not be affected. The creation-time attribute of the newly created trace stream shall be set to the value of the system clock, if the Timers option is not supported, or to the value of the CLOCK_REALTIME clock, if the Timers option is supported.

The pid argument represents the target process to be traced. If the process executing this function does not have appropriate privileges to trace the process identified by pid, an error shall be returned. If the pid argument is zero, the calling process shall be traced.

The posix_trace_create() function shall store the trace stream identi? fier of the new trace stream in the object pointed to by the trid argu? ment. This trace stream identifier shall be used in subsequent calls to control tracing. The trid argument may only be used by the following functions:

posix_trace_clear() posix_trace_getnext_event()

posix_trace_eventid_equal() posix_trace_shutdown()

posix_trace_eventid_get_name() posix_trace_start()

posix_trace_eventtypelist_getnext_id() posix_trace_stop()

posix_trace_eventtypelist_rewind() posix_trace_timedgetnext_event()

posix_trace_get_attr() posix_trace_trid_eventid_open()

posix_trace_get_status() posix_trace_trygetnext_event()

If the Trace Event Filter option is supported, the following additional functions may use the trid argument:

posix_trace_get_filter() posix_trace_set_filter()

In particular, notice that the operations normally used by a trace ana?

lyzer process, such as posix_trace_rewind() or posix_trace_close(), cannot be invoked using the trace stream identifier returned by the posix_trace_create() function.

A trace stream shall be created in a suspended state. If the Trace Event Filter option is supported, its trace event type filter shall be empty.

The posix_trace_create() function may be called multiple times from the same or different processes, with the system-wide limit indicated by the runtime invariant value {TRACE_SYS_MAX}, which has the minimum value {_POSIX_TRACE_SYS_MAX}.

The trace stream identifier returned by the posix_trace_create() func? tion in the argument pointed to by trid is valid only in the process that made the function call. If it is used from another process, that is a child process, in functions defined in POSIX.1?2008, these func? tions shall return with the error [EINVAL].

The posix_trace_create_withlog() function shall be equivalent to posix_trace_create(), except that it associates a trace log with this stream. The file_desc argument shall be the file descriptor designating the trace log destination. The function shall fail if this file de? scriptor refers to a file with a file type that is not compatible with the log policy associated with the trace log. The list of the appropri? ate file types that are compatible with each log policy is implementa? tion-defined.

The posix_trace_create_withlog() function shall return in the parameter pointed to by trid the trace stream identifier, which uniquely identi? fies the newly created trace stream, and shall be used in subsequent calls to control tracing. The trid argument may only be used by the following functions:

posix_trace_clear() posix_trace_get_status()

posix_trace_eventid_equal() posix_trace_getnext_event()

posix_trace_eventid_get_name() posix_trace_shutdown()

posix_trace_eventtypelist_getnext_id() posix_trace_start()

posix_trace_eventtypelist_rewind() posix_trace_stop()

posix_trace_flush() posix_trace_timedgetnext_event()

posix_trace_get_attr() posix_trace_trid_eventid_open()

If the Trace Event Filter option is supported, the following additional functions may use the trid argument:

posix_trace_get_filter() posix_trace_set_filter()

In particular, notice that the operations normally used by a trace ana? lyzer process, such as posix_trace_rewind() or posix_trace_close(), cannot be invoked using the trace stream identifier returned by the posix_trace_create_withlog() function.

The posix_trace_flush() function shall initiate a flush operation which copies the contents of the trace stream identified by the argument trid into the trace log associated with the trace stream at the creation time. If no trace log has been associated with the trace stream pointed to by trid, this function shall return an error. The termination of the flush operation can be polled by the posix_trace_get_status() function. During the flush operation, it shall be possible to trace new trace events up to the point when the trace stream becomes full. After flushing is completed, the space used by the flushed trace events shall be available for tracing new trace events.

If flushing the trace stream causes the resulting trace log to become full, the trace log full policy shall be applied. If the trace log-full-policy attribute is set, the following occurs:

POSIX_TRACE_UNTIL_FULL

The trace events that have not yet been flushed shall be dis? carded.

POSIX_TRACE_LOOP

The trace events that have not yet been flushed shall be written to the beginning of the trace log, overwriting previous trace events stored there.

POSIX_TRACE_APPEND

The trace events that have not yet been flushed shall be appended to the trace log.

The posix_trace_shutdown() function shall stop the tracing of trace

events in the trace stream identified by trid, as if posix_trace_stop() had been invoked. The posix_trace_shutdown() function shall free all the resources associated with the trace stream.

The posix_trace_shutdown() function shall not return until all the re? sources associated with the trace stream have been freed. When the posix_trace_shutdown() function returns, the trid argument becomes an invalid trace stream identifier. A call to this function shall uncondi? tionally deallocate the resources regardless of whether all trace events have been retrieved by the analyzer process. Any thread blocked on one of the trace_getnext_event() functions (which specified this trid) before this call is unblocked with the error [EINVAL].

If the process exits, invokes a member of the exec family of functions, or is terminated, the trace streams that the process had created and that have not yet been shut down, shall be automatically shut down as if an explicit call were made to the posix_trace_shutdown() function.

For an active trace stream with log, when the posix_trace_shutdown() function is called, all trace events that have not yet been flushed to the trace log shall be flushed, as in the posix_trace_flush() function, and the trace log shall be closed.

When a trace log is closed, all the information that may be retrieved later from the trace log through the trace interface shall have been written to the trace log. This information includes the trace at? tributes, the list of trace event types (with the mapping between trace event names and trace event type identifiers), and the trace status. In addition, unspecified information shall be written to the trace log to allow detection of a valid trace log during the posix_trace_open() operation.

The posix_trace_shutdown() function shall not return until all trace events have been flushed.

RETURN VALUE

Upon successful completion, these functions shall return a value of zero. Otherwise, they shall return the corresponding error number.

The posix_trace_create() and posix_trace_create_withlog() functions

store the trace stream identifier value in the object pointed to by trid, if successful.

ERRORS

The posix_trace_create() and posix_trace_create_withlog() functions shall fail if:

EAGAIN No more trace streams can be started now. {TRACE_SYS_MAX} has been exceeded.

EINTR The operation was interrupted by a signal. No trace stream was created.

EINVAL One or more of the trace parameters specified by the attr param?eter is invalid.

ENOMEM The implementation does not currently have sufficient memory to create the trace stream with the specified parameters.

EPERM The caller does not have appropriate privileges to trace the process specified by pid.

ESRCH The pid argument does not refer to an existing process.

The posix_trace_create_withlog() function shall fail if:

EBADF The file_desc argument is not a valid file descriptor open for writing.

EINVAL The file_desc argument refers to a file with a file type that does not support the log policy associated with the trace log.

ENOSPC No space left on device. The device corresponding to the argu?ment file_desc does not contain the space required to create this trace log.

The posix_trace_flush() and posix_trace_shutdown() functions shall fail if:

EINVAL The value of the trid argument does not correspond to an active trace stream with log.

EFBIG The trace log file has attempted to exceed an implementation-de?fined maximum file size.

ENOSPC No space left on device.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

The posix_trace_create(), posix_trace_create_withlog(), posix_trace_flush(), and posix_trace_shutdown() functions may be re? moved in a future version.

SEE ALSO

clock_getres(), exec, posix_trace_attr_destroy(), posix_trace_clear(), posix_trace_close(), posix_trace_eventid_equal(), posix_trace_event? typelist_getnext_id(), posix_trace_get_attr(), posix_trace_get_fil? ter(), posix_trace_getnext_event(), posix_trace_start(), posix_trace_start(), time()

The Base Definitions volume of POSIX.1?2017, <sys_types.h>, <trace.h>

COPYRIGHT