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 'trace.h.0p' command

$ man trace.h.0p

trace.h(0P) POSIX Programmer's Manual trace.h(0P)

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

trace.h ? tracing

SYNOPSIS

#include <trace.h>

DESCRIPTION

The <trace.h> header shall define the posix_trace_event_info structure, which shall include at least the following members:

```
trace_event_id_t  posix_event_id
pid_t             posix_pid
void              *posix_prog_address
pthread_t         posix_thread_id
struct timespec   posix_timestamp
int               posix_truncation_status
```

The <trace.h> header shall define the posix_trace_status_info struc? ture, which shall include at least the following members:

```
int     posix_stream_full_status
int     posix_stream_overrun_status
```

int posix_stream_status

int posix_log_full_status

int posix_log_overrun_status

int posix_stream_flush_error

int posix_stream_flush_status

The <trace.h> header shall define the following symbolic constants:

POSIX_TRACE_ALL_EVENTS

POSIX_TRACE_APPEND

POSIX_TRACE_CLOSE_FOR_CHILD

POSIX_TRACE_FILTER

POSIX_TRACE_FLUSH

POSIX_TRACE_FLUSH_START

POSIX_TRACE_FLUSH_STOP

POSIX_TRACE_FLUSHING

POSIX_TRACE_FULL

POSIX_TRACE_LOOP

POSIX_TRACE_NO_OVERRUN

POSIX_TRACE_NOT_FLUSHING

POSIX_TRACE_NOT_FULL

POSIX_TRACE_INHERITED

POSIX_TRACE_NOT_TRUNCATED

POSIX_TRACE_OVERFLOW

POSIX_TRACE_OVERRUN

POSIX_TRACE_RESUME

POSIX_TRACE_RUNNING

POSIX_TRACE_START

POSIX_TRACE_STOP

POSIX_TRACE_SUSPENDED

POSIX_TRACE_SYSTEM_EVENTS

POSIX_TRACE_TRUNCATED_READ

POSIX_TRACE_TRUNCATED_RECORD

POSIX_TRACE_UNNAMED_USER_EVENT

POSIX_TRACE_UNTIL_FULL

POSIX_TRACE_WOPID_EVENTS

The <trace.h> header shall define the size_t, trace_attr_t, trace_event_id_t, trace_event_set_t, and trace_id_t types as described in <sys/types.h>.

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided.

```
int  posix_trace_attr_destroy(trace_attr_t *);
int  posix_trace_attr_getclockres(const trace_attr_t *,
        struct timespec *);
int  posix_trace_attr_getcreatetime(const trace_attr_t *,
        struct timespec *);
int  posix_trace_attr_getgenversion(const trace_attr_t *, char *);
int  posix_trace_attr_getinherited(const trace_attr_t *restrict,
        int *restrict);
int  posix_trace_attr_getlogfullpolicy(const trace_attr_t *restrict,
        int *restrict);
int  posix_trace_attr_getlogsize(const trace_attr_t *restrict,
        size_t *restrict);
int  posix_trace_attr_getmaxdatasize(const trace_attr_t *restrict,
        size_t *restrict);
int  posix_trace_attr_getmaxsystemeventsize(const trace_attr_t *restrict,
        size_t *restrict);
int  posix_trace_attr_getmaxusereventsize(const trace_attr_t *restrict,
        size_t, size_t *restrict);
int  posix_trace_attr_getname(const trace_attr_t *, char *);
int  posix_trace_attr_getstreamfullpolicy(const trace_attr_t *restrict,
        int *restrict);
int  posix_trace_attr_getstreamsize(const trace_attr_t *restrict,
        size_t *restrict);
int  posix_trace_attr_init(trace_attr_t *);
int  posix_trace_attr_setinherited(trace_attr_t *, int);
int  posix_trace_attr_setlogfullpolicy(trace_attr_t *, int);
int  posix_trace_attr_setlogsize(trace_attr_t *, size_t);
```

```
int  posix_trace_attr_setmaxdatasize(trace_attr_t *, size_t);
int  posix_trace_attr_setname(trace_attr_t *, const char *);
int  posix_trace_attr_setstreamfullpolicy(trace_attr_t *, int);
int  posix_trace_attr_setstreamsize(trace_attr_t *, size_t);
int  posix_trace_clear(trace_id_t);
int  posix_trace_close(trace_id_t);
int  posix_trace_create(pid_t, const trace_attr_t *restrict,
        trace_id_t *restrict);
int  posix_trace_create_withlog(pid_t, const trace_attr_t *restrict,
        int, trace_id_t *restrict);
void posix_trace_event(trace_event_id_t, const void *restrict, size_t);
int  posix_trace_eventid_equal(trace_id_t, trace_event_id_t,
        trace_event_id_t);
int  posix_trace_eventid_get_name(trace_id_t, trace_event_id_t, char *);
int  posix_trace_eventid_open(const char *restrict,
        trace_event_id_t *restrict);
int  posix_trace_eventset_add(trace_event_id_t, trace_event_set_t *);
int  posix_trace_eventset_del(trace_event_id_t, trace_event_set_t *);
int  posix_trace_eventset_empty(trace_event_set_t *);
int  posix_trace_eventset_fill(trace_event_set_t *, int);
int  posix_trace_eventset_ismember(trace_event_id_t,
        const trace_event_set_t *restrict, int *restrict);
int  posix_trace_eventtypelist_getnext_id(trace_id_t,
        trace_event_id_t *restrict, int *restrict);
int  posix_trace_eventtypelist_rewind(trace_id_t);
int  posix_trace_flush(trace_id_t);
int  posix_trace_get_attr(trace_id_t, trace_attr_t *);
int  posix_trace_get_filter(trace_id_t, trace_event_set_t *);
int  posix_trace_get_status(trace_id_t,
        struct posix_trace_status_info *);
int  posix_trace_getnext_event(trace_id_t,
        struct posix_trace_event_info *restrict, void *restrict,
        size_t, size_t *restrict, int *restrict);
```

```
int  posix_trace_open(int, trace_id_t *);
int  posix_trace_rewind(trace_id_t);
int  posix_trace_set_filter(trace_id_t, const trace_event_set_t *, int);
int  posix_trace_shutdown(trace_id_t);
int  posix_trace_start(trace_id_t);
int  posix_trace_stop(trace_id_t);
int  posix_trace_timedgetnext_event(trace_id_t,
        struct posix_trace_event_info *restrict, void *restrict,
        size_t, size_t *restrict, int *restrict,
        const struct timespec *restrict);
int  posix_trace_trid_eventid_open(trace_id_t, const char *restrict,
        trace_event_id_t *restrict);
int  posix_trace_trygetnext_event(trace_id_t,
        struct posix_trace_event_info *restrict, void *restrict, size_t,
        size_t *restrict, int *restrict);
```

The following sections are informative.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

The <trace.h> header may be removed in a future version.

SEE ALSO

<sys_types.h>

The System Interfaces volume of POSIX.1?2017, Section 2.11, Tracing,

posix_trace_attr_destroy(), posix_trace_attr_getclockres(),

posix_trace_attr_getinherited(), posix_trace_attr_getlogsize(),

posix_trace_clear(), posix_trace_close(), posix_trace_create(),

posix_trace_event(), posix_trace_eventid_equal(),

posix_trace_eventset_add(), posix_trace_eventtypelist_getnext_id(),

posix_trace_get_attr(), posix_trace_get_filter(), posix_trace_get?

next_event(), posix_trace_start()

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 .