libfiber Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

Datatypes

SceFiber

Datatype for a fiber

Definition

```
#include <sce_fiber.h>
typedef struct{
        /* Private */
} SceFiber;
```

Description

This structure represents a fiber. To make this structure usable, prepare this structure from the application side and initialize it by calling `sceFiberInitialize()`.

It is not necessary or appropriate for applications to handle the members of this structure directly.

This structure must be aligned in memory in accordance with the prescribed alignment.

The macro constants for this datatype are defined as the following values.

Macro	Value	Description
SCE_FIBER_ALIGNMENT	8	Alignment for this datatype
SCE_FIBER_SIZE	128	Size

See Also

`sceFiberInitialize()`

SceFiberInfo

Datatype for fiber related information

Definition

```
#include <sce_fiber.h>
typedef struct{
        SceFiberEntry entry;
        SceUInt32 argOnInitialize;
        void* addrContext;
        SceUInt32 sizeContext;
        char name[SCE_FIBER_MAX_NAME_LENGTH + 1];
        SceUInt32 sizeContextMargin;
        /* other members are undisclosed */
} SceFiberInfo;
```

Members

entry	Entry function
argOnInitialize	Argument passed during initialization
addrContext	Starting address of the memory area used for storing context
sizeContext	Size of the memory area used for storing context
name	Name
sizeContextMargin	Remaining memory used for storing context

Description

This structure represents fiber related information. Applications can obtain its contents with `sceFiberGetInfo()`.

This structure must be aligned in memory in accordance with the prescribed alignment.

The macro constants for this datatype are defined as the following values.

Macro	Value	Description
SCE_FIBER_INFO_ALIGNMENT	8	Alignment for this datatype
SCE_FIBER_INFO_SIZE	128	Size

In *sizeContextMargin*, the remaining memory for storing the context passed at the time of fiber creation will be inserted. The memory units are in bytes.

The content of *sizeContextMargin* is valid only for the fibers created during the measurement period. For other fibers, the *sizeContextMargin* value is undefined. Start the measurement period with `sceFiberStartContextSizeCheck()` and stop it with `sceFiberStopContextSizeCheck()`. In addition, the measurement period is not during the time when libfiber starts.

See Also

`sceFiberGetInfo(), SceFiber, sceFiberStartContextSizeCheck(), sceFiberStopContextSizeCheck()`

SceFiberEntry

Fiber entry function

Definition

```
#include <sce_fiber.h>
typedef void (*SceFiberEntry)(
        SceUInt32 argOnInitialize,
        SceUInt32 argOnRun
) __attribute__((noreturn));
```

Arguments

argOnInitialize	Argument passed during initialization
argOnRun	Argument passed when a fiber is started

Description

This datatype represents a fiber entry function. The entry function is used as the initial value of the fiber program counter. The value of the *argOnInitialize* argument in the entry function will be set when sceFiberInitialize() is called. The value of the *argOnRun* argument in the entry function will be set when the fiber is started. Fibers can be started with sceFiberRun() or sceFiberSwitch().

Notes

Return is not possible from the fiber entry function. sceFiberRun() of the thread that executed this fiber returns the SCE_FIBER_ERROR_STATE error and terminates for returning from the entry function.

See Also

sceFiberInitialize(),sceFiberRun(),sceFiberSwitch()

SceFiberOptParam

Fiber options

Definition

```
#include <sce_fiber.h>
typedef struct SceFiberOptParam {
        /* omitted */
} SceFiberOptParam;
```

Description

This structure represents fiber options. It is specified upon initializing a fiber with `sceFiberInitialize()`. Initialize this structure with `sceFiberOptParamInitialize()` in advance.

There are no currently selectable options.

This structure must be aligned in memory in accordance with the prescribed alignment.

The macro constants for this datatype are defined as the following values.

Macro	Value	Description
SCE_FIBER_OPT_PARAM_ALIGNMENT	8	Alignment for this datatype
SCE_FIBER_OPT_PARAM_SIZE	128	Size

See Also

`sceFiberInitialize(),sceFiberOptParamInitialize()`

Functions

sceFiberInitialize

Initialize a fiber

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberInitialize(
        SceFiber* fiber,
        const char* name,
        SceFiberEntry entry,
        SceUInt32 argOnInitialize,
        void* addrContext,
        SceUInt32 sizeContext,
        const SceFiberOptParam* option
);
```

Calling Conditions

Can be called from a thread and a fiber

Multithread safe

Arguments

fiber	Fiber to be initialized
name	Name given to this fiber
entry	Entry function
argOnInitialize	Argument for the entry function
addrContext	Starting address of the memory area used for storing the fiber's context
sizeContext	Size of the memory area used for storing the fiber's context
option	Options for this fiber

Return Values

Returns `SCE_OK(0)` for success.

Returns one of the following error codes for errors.

Macro	Value	Description
SCE_FIBER_ERROR_NULL	0x80590001	*fiber* or *name* or *entry* is a NULL pointer.
SCE_FIBER_ERROR_ALIGNMENT	0x80590002	One of the following: - *fiber* or *addrContext* is not within the correct boundaries. - *Option* is not NULL and is not within the correct boundaries.
SCE_FIBER_ERROR_RANGE	0x80590003	*sizeContext* is a non-zero value, and is smaller than SCE_FIBER_CONTEXT_MINIMUM_SIZE.
SCE_FIBER_ERROR_INVALID	0x80590004	One of the following: - *sizeContext* is not multiples of 8. - *addrContext* is NULL and *sizeContext* is a non-zero value. - *addrContext* is not NULL and *sizeContext* is 0. - *Option* is not NULL and the contents are invalid.

Description

This function initializes the fiber specified with *fiber*, and sets its state to the initialized state. Always initialize `SceFiber` structures with this function before use. The `SceFiber` structure specified with *fiber* will become usable.

name is the name given to this fiber. A name of a length up to the length specified by `SCE_FIBER_MAX_NAME_LENGTH` can be given. If *name* exceeds `SCE_FIBER_MAX_NAME_LENGTH`, the name of this fiber is truncated so that the length is `SCE_FIBER_MAX_NAME_LENGTH`.

entry is the entry point for the fiber. Execution starts at this point when the state transitions from the initialized state to the running state.

argOnInitialize is passed as an argument to *entry*.

addrContext and *sizeContext* are used to specify the memory used for this fiber's context storage area. If NULL is specified for *addrContext*, and 0 is specified for *sizeContext*, execution of this fiber will use the stack in the thread. If, in this situation, the fiber is suspended, the state will return to the initialized state, not the suspended state. When this fiber transitions again to the running state, execution will begin again from the entry point.

Specify options, or NULL, for *option*. When specifying options, first create an `SceFiberOptParam` structure, and initialize this structure with `sceFiberOptParamInitialize()` in advance. It is not necessary to hold the contents of *option* after calling this function.

See Also

`sceFiberFinalize()`,`sceFiberOptParamInitialize()`

sceFiberOptParamInitialize

Initialize the structure which specifies fiber options

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberOptParamInitialize(
        SceFiberOptParam* optParam
);
```

Calling Conditions

Can be called from a thread and a fiber

Multithread safe

Arguments

optParam	Pointer to SceFiberOptParam structure to be initialized

Return Values

Returns SCE_OK(0) for success.

Returns one of the following error codes for errors.

Macro	Value	Description
SCE_FIBER_ERROR_NULL	0x80590001	*optParam* is a NULL pointer.
SCE_FIBER_ERROR_ALIGNMENT	0x80590002	*optParam* is not within the correct boundaries.

Description

This function initializes the SceFiberOptParam structure specified with *optParam*. The SceFiberOptParam structure passed to sceFiberInitialize() must first be initialized with this function.

See Also

sceFiberInitialize()

sceFiberFinalize

Finalize the fiber

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberFinalize(
        SceFiber* fiber
);
```

Calling Conditions

Can be called from a thread and a fiber

Multithread safe

Arguments

fiber	The fiber to be finalized

Return Values

Returns `SCE_OK(0)` for success.

Returns one of the following error codes for errors.

Macro	Value	Description
`SCE_FIBER_ERROR_NULL`	0x80590001	*fiber* is a NULL pointer.
`SCE_FIBER_ERROR_ALIGNMENT`	0x80590002	*fiber* is not within the correct boundaries.
`SCE_FIBER_ERROR_INVALID`	0x80590004	Detected that the contents of *fiber* have been corrupted.
`SCE_FIBER_ERROR_STATE`	0x80590006	The fiber specified with *fiber* is in the running state, or the specified structure is invalid.

Description

This function is used to finalize the fiber specified with *fiber*. The fiber to be finalized must be in either the initialized state or suspended state. The `SceFiber` structure specified with *fiber* will become unusable.

See Also

`sceFiberInitialize()`

sceFiberRun

Execute fibers

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberRun(
        SceFiber* fiber,
        SceUInt32 argOnRunTo,
        SceUInt32* argOnReturn
);
```

Calling Conditions

Can only be called from a thread

Multithread safe

Arguments

fiber	Fiber to be executed
argOnRunTo	User-defined number passed to fiber to be executed
argOnReturn	Pointer to the variable for receiving the user-defined number passed by `sceFiberReturnToThread()`

Return Values

Returns `SCE_OK(0)` for success.

Returns one of the following error codes for errors.

Macro	Value	Description
`SCE_FIBER_ERROR_NULL`	0x80590001	*fiber* is a NULL pointer.
`SCE_FIBER_ERROR_ALIGNMENT`	0x80590002	*fiber* is not within the correct boundaries.
`SCE_FIBER_ERROR_INVALID`	0x80590004	Detected that the contents of *fiber* have been corrupted.
`SCE_FIBER_ERROR_PERMISSION`	0x80590005	Caller is a fiber.
`SCE_FIBER_ERROR_STATE`	0x80590006	One of the following: - The fiber specified with *fiber* is in the running state, or the specified structure is invalid. - The executed fiber was returned from the entry function.

Description

This function is used for the thread to start the execution of fibers. When this function is called, the fiber specified with *`fiber`* is executed in the thread which called it, and the state changes to the running state. This function is completed by the fiber running on the thread calling `sceFiberReturnToThread()`.

If the fiber is in the initialized state, *`argOnRunTo`* is passed as argument *`argOnRun`* to *`entry`*. If the fiber is in the suspended state and the pointer *`argOnRun`* passed to the function (`sceFiberSwitch()` or `sceFiberReturnToThread()`) that suspended this fiber is not NULL, *`argOnRunTo`* will be stored in `*`*`argOnRun`*.

If a value other than NULL is specified to *`argOnReturn`*, the value passed to `sceFiberReturnToThread()` parameter *`argOnReturn`* will be stored in `*`*`argOnReturn`*.

If the fiber executed in the thread that called this function returns from the entry function, this function returns the `SCE_FIBER_ERROR_STATE` error and terminates. If, at this time, a value other than NULL is specified to *`argOnReturn`*, the pointer to the `SceFiber` structure of the problematic fiber is saved to `*`*`argOnReturn`*.

See Also

`sceFiberSwitch(),sceFiberReturnToThread()`

sceFiberSwitch

Switch execution of fibers

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberSwitch(
        SceFiber* fiber,
        SceUInt32 argOnRunTo,
        SceUInt32* argOnRun
);
```

Calling Conditions

Can only be called from a fiber

Multithread safe

Arguments

fiber	Fiber to be executed next
argOnRunTo	User-defined number passed to fiber to be executed next
argOnRun	Pointer for receiving user-defined number from function which restarts execution of this fiber

Return Values

Returns `SCE_OK(0)` for success.

Returns one of the following error codes for errors.

Macro	Value	Description
`SCE_FIBER_ERROR_NULL`	0x80590001	*fiber* is a NULL pointer.
`SCE_FIBER_ERROR_ALIGNMENT`	0x80590002	*fiber* is not within the correct boundaries.
`SCE_FIBER_ERROR_INVALID`	0x80590004	Detected that the contents of *fiber* have been corrupted.
`SCE_FIBER_ERROR_PERMISSION`	0x80590005	The caller is not a fiber
`SCE_FIBER_ERROR_STATE`	0x80590006	The fiber specified with *fiber* is in the running state, or the specified structure is invalid.

Description

This function switches the fiber that is being executed by a thread with another fiber. When this function is called, the state of the fiber specified with *fiber* changes to the running state. If the immediately preceding state was the initialized state, execution starts from the entry point. If the immediately preceding state was the suspended state, the function call used to suspend execution is completed. Execution of the calling fiber is stopped. Fibers with `sceFiberInitialize()` *addrContext* values other than NULL will change to the suspended state. Fibers with NULL values will change to the initialized state.

The function call will finally be completed when `sceFiberSwitch()` or `sceFiberRun()` is used to return the caller fiber (in the suspended state) to the running state.

If the fiber specified with *fiber* is in the initialized state, *argOnRunTo* is passed as argument *argOnRun* to *entry*. If the fiber specified with *fiber* is in the suspended state and the pointer *argOnRun* passed to the function (`scefiberSwitch()` or `scefiberReturnToThread()`) that suspended the fiber specified with *fiber* is not NULL, *argOnRunTo* will be stored in `*`*argOnRun*.

If a value other than NULL is specified to *argOnRun*, *argOnRunTo* from the function (`sceFiberRun()` or `sceFiberSwitch()`) used to change this fiber's state to the running state is passed to `*`*argOnRun*.

See Also

`sceFiberInitialize(),sceFiberRun(),sceFiberReturnToThread()`

sceFiberGetSelf

Get the fiber

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberGetSelf(
        SceFiber** fiber
);
```

Calling Conditions

Can be called from a thread and a fiber

Multithread safe

Arguments

fiber	Pointer that returns pointer of fiber currently being executed

Return Values

Returns SCE_OK(0) for success.

Returns the following error code for errors.

Macro	Value	Description
SCE_FIBER_ERROR_NULL	0x80590001	*fiber* is a NULL pointer.

Description

This function is used to obtain the currently-executed fiber. When the caller is a fiber, the address to the fiber currently being executed is stored in **fiber*. When the caller is not a fiber, NULL is stored in **fiber*.

See Also

SceFiber

sceFiberGetInfo

Get fiber related information

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberGetInfo(
        SceFiber* fiber,
        SceFiberInfo* fiberInfo
);
```

Calling Conditions

Multithread safe

Arguments

Fiber	Fiber of which information to be obtained
fiberInfo	Pointer to the structure storing the obtained information

Return Values

Returns `SCE_OK(0)` for success.

Returns one of the following error codes for errors.

Macro	Value	Description
`SCE_FIBER_ERROR_NULL`	0x80590001	*fiber* or *fiberInfo* is a NULL pointer.
`SCE_FIBER_ERROR_ALIGNMENT`	0x80590002	*fiber* or *fiberInfo* is not within the correct boundaries.
`SCE_FIBER_ERROR_INVALID`	0x80590004	Detected that the contents of *fiber* have been corrupted.

Description

This function is used to obtain information on a fiber. Information on a fiber specified by *fiber* is copied to the `SceFiberInfo` structure which published its members.

See Also

`SceFiberInfo`, `sceFiberGetSelf()`

sceFiberRenameSelf

Rename fiber

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberRenameSelf(
        const char* name
);
```

Calling Conditions

Multithread safe

Arguments

name New name applied to fiber currently being executed

Return Values

Returns SCE_OK(0) for success.

Returns the following error code for errors.

Macro	Value	Description
SCE_FIBER_ERROR_NULL	0x80590001	*name* is a NULL pointer.
SCE_FIBER_ERROR_PERMISSION	0x80590005	The caller is not a fiber

Description

This function is used to rename the currently-executed fiber.

See Also

SceFiber

sceFiberStartContextSizeCheck

Start the measurement period for the amount of memory used for the fiber context

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberStartContextSizeCheck(
        SceUInt32 flags
);
```

Calling Conditions

Multithread safe

Arguments

flags	Flags

Return Values

Returns `SCE_OK(0)` for success.

Returns one of the following error codes for errors.

Macro	Value	Description
`SCE_FIBER_ERROR_INVALID`	0x80590004	A value other than 0 was specified for *flags*
`SCE_FIBER_ERROR_STATE`	0x80590006	Already a measurement period

Description

This function starts the measurement period for the amount of memory used for the fiber context. For details, refer to the datatype `SceFiberInfo`.

There are currently no flags to assign to *flags*. Specify 0.

See Also

`SceFiberInfo`, `sceFiberGetInfo()`, `sceFiberStopContextSizeCheck()`

sceFiberStopContextSizeCheck

Stop the measurement period for the amount of memory used for the fiber context

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberStopContextSizeCheck(void);
```

Calling Conditions

Multithread safe

Arguments

None

Return Values

Returns `SCE_OK(0)` for success.

Returns the following error code for errors.

Macro	Value	Description
`SCE_FIBER_ERROR_STATE`	0x80590006	Not yet a measurement period

Description

This function stops the measurement period for the amount of memory used for the fiber context. For details, refer to the datatype `SceFiberInfo`.

See Also

`SceFiberInfo`, `sceFiberGetInfo()`, `sceFiberStartContextSizeCheck()`

sceFiberPushUserMarkerWithHud

Push a marker with support for the Razor HUD

Definition

```
#include <sce_fiber.h>
int sceFiberPushUserMarkerWithHud(
        const char* label,
        SceUInt32 color,
        SceUInt32 flags
);
```

Calling Conditions

Can only be called from a fiber

Multithread safe

Arguments

label	Label to describe the marker
color	Color of marker on HUD
flags	`SCE_RAZOR_MARKER_DISABLE_HUD` to disable marker on HUD `SCE_RAZOR_MARKER_ENABLE_HUD` to enable on HUD

Return Values

Returns `SCE_OK(0)` for success.

Returns one of the following error codes for errors.

Macro	Value	Description
`SCE_FIBER_ERROR_STATE`	0x80590006	One of the following - The caller is not a fiber - libperf is not loaded
`SCE_FIBER_ERROR_BUSY`	0x80590007	The maximum number of markers has been exceeded

Description

This function pushes a marker with a specified label. Markers are visible to the Razor host tool, the Razor HUD (if *flags* is set to `SCE_RAZOR_MARKER_ENABLE_HUD`). Users can create their own custom performance HUD by creating the trace buffers using `sceRazorCpuStartUserMarkerTrace()` and accessing the trace through `sceRazorCpuGetUserMarkerTraceBuffer()`. A maximum of 64 markers per thread may be pushed onto the stack.

Use this function instead of `sceRazorCpuPushMarkerWithHud()` to push a marker from a fiber.

For details, refer to the "libperf Overview" and "libperf Reference" documents.

See Also

`sceRazorCpuPushMarkerWithHud()`, `sceFiberPopUserMarkerWithHud()`

sceFiberPopUserMarkerWithHud

Pop a marker with support for the Razor HUD

Definition

```
#include <sce_fiber.h>
int sceFiberPopUserMarkerWithHud(void);
```

Calling Conditions

Can only be called from a fiber

Multithread safe

Arguments

None

Return Values

Returns SCE_OK(0) for success.

Returns one of the following error codes for errors.

Macro	Value	Description
SCE_FIBER_ERROR_STATE	0x80590006	One of the following - The caller is not a fiber - libperf is not loaded
SCE_FIBER_ERROR_BUSY	0x80590007	No poppable marker exists

Description

This function pops a marker.

Use this function instead of sceRazorCpuPopMarker() to pop a marker from a fiber.

For details, refer to the "libperf Overview" and "libperf Reference" documents.

See Also

sceRazorCpuPopMarker(), sceFiberPushUserMarkerWithHud()

sceFiberReturnToThread

Suspend fiber execution and return to the thread

Definition

```
#include <sce_fiber.h>
SceInt32 sceFiberReturnToThread(
        SceUInt32 argOnReturn,
        SceUInt32* argOnRun
);
```

Calling Conditions

Can only be called from a fiber

Multithread safe

Arguments

argOnReturn	User-defined number passed to `sceFiberRun()`
argOnRun	Pointer for receiving user-defined number from function which restarts execution of this fiber

Return Values

Returns `SCE_OK(0)` for success.

Returns the following error code for errors.

Macro	Value	Description
SCE_FIBER_ERROR_PERMISSION	0x80590005	The caller is not a fiber

Description

This function is used to suspend execution of the calling fiber, and return to the thread. When this function is called, `sceFiberRun()` of the thread that is executing the fiber will be completed.

Execution of the calling fiber is stopped. Fibers with `sceFiberInitialize()` *addrContext* values other than NULL will change to the suspended state. Fibers with NULL values will change to the initialized state.

Any user-defined integer can be specified in *argOnReturn*. This value will be stored in **argOnReturn* provided that the *argOnReturn* parameter of `sceFiberRun()` called by the thread contains a value other than NULL.

The function call will finally be completed when `sceFiberSwitch()` or `sceFiberRun()` is used to return the caller fiber (in the suspended state) to the running state.

See Also

`sceFiberInitialize(),sceFiberRun(),sceFiberSwitch()`

Constants

Fiber Macros

List of Fiber Related Macros

Header

```
#include <sce_fiber.h>
```

Definition

Macro	Value	Description
`SCE_FIBER_SIZE`	128	`SceFiber` structure size
`SCE_FIBER_ALIGNMENT`	8	`SceFiber` structure alignment
`SCE_FIBER_INFO_SIZE`	128	`SceFiberInfo` structure size
`SCE_FIBER_INFO_ALIGNMENT`	8	`SceFiberInfo` structure alignment
`SCE_FIBER_OPT_PARAM_SIZE`	128	`SceFiberOptParam` structure size
`SCE_FIBER_OPT_PARAM_ALIGNMENT`	8	`SceFiberOptParam` structure alignment
`SCE_FIBER_CONTEXT_MINIMUM_SIZE`	512	Minimum required memory for the fiber context storage area
`SCE_FIBER_CONTEXT_ALIGNMENT`	8	Alignment of fiber context storage area
`SCE_FIBER_MAX_NAME_LENGTH`	31	Maximum length of name that can be given to fiber

Return Codes

List of return codes returned by libfiber

Header

```
#include <sce_fiber.h>
```

Definition

Macro	Value	Description
`SCE_OK`	0x00000000	Success
`SCE_FIBER_ERROR_NULL`	0x80590001	The specified pointer is NULL
`SCE_FIBER_ERROR_ALIGNMENT`	0x80590002	There is an alignment error in the specified pointer
`SCE_FIBER_ERROR_RANGE`	0x80590003	The specified value is not within the accepted range
`SCE_FIBER_ERROR_INVALID`	0x80590004	The specified parameter is invalid.
`SCE_FIBER_ERROR_PERMISSION`	0x80590005	The caller is attempting to perform an unauthorized operation
`SCE_FIBER_ERROR_STATE`	0x80590006	The fiber state does not allow that operation
`SCE_FIBER_ERROR_BUSY`	0x80590007	The resources cannot be used