*/\* SPDX-License-Identifier: GPL-2.0 \*/*

#ifndef [**\_ALPHA\_THREAD\_INFO\_H**](https://elixir.bootlin.com/linux/latest/C/ident/_ALPHA_THREAD_INFO_H)

#define [**\_ALPHA\_THREAD\_INFO\_H**](https://elixir.bootlin.com/linux/latest/C/ident/_ALPHA_THREAD_INFO_H)

#ifdef [**\_\_KERNEL\_\_**](https://elixir.bootlin.com/linux/latest/C/ident/__KERNEL__)

#ifndef [**\_\_ASSEMBLY\_\_**](https://elixir.bootlin.com/linux/latest/C/ident/__ASSEMBLY__)

#include <asm/processor.h>

#include <asm/types.h>

#include <asm/hwrpb.h>

#include <asm/sysinfo.h>

#endif

#ifndef [**\_\_ASSEMBLY\_\_**](https://elixir.bootlin.com/linux/latest/C/ident/__ASSEMBLY__)

struct [**thread\_info**](https://elixir.bootlin.com/linux/latest/C/ident/thread_info) {

struct [**pcb\_struct**](https://elixir.bootlin.com/linux/latest/C/ident/pcb_struct) [**pcb**](https://elixir.bootlin.com/linux/latest/C/ident/pcb); */\* palcode state \*/*

struct [**task\_struct**](https://elixir.bootlin.com/linux/latest/C/ident/task_struct) \*[**task**](https://elixir.bootlin.com/linux/latest/C/ident/task); */\* main task structure \*/*

unsigned int flags; */\* low level flags \*/*

unsigned int [**ieee\_state**](https://elixir.bootlin.com/linux/latest/C/ident/ieee_state); */\* see fpu.h \*/*

[**mm\_segment\_t**](https://elixir.bootlin.com/linux/latest/C/ident/mm_segment_t) [**addr\_limit**](https://elixir.bootlin.com/linux/latest/C/ident/addr_limit); */\* thread address space \*/*

unsigned cpu; */\* current CPU \*/*

int [**preempt\_count**](https://elixir.bootlin.com/linux/latest/C/ident/preempt_count); */\* 0 => preemptable, <0 => BUG \*/*

unsigned int status; */\* thread-synchronous flags \*/*

int [**bpt\_nsaved**](https://elixir.bootlin.com/linux/latest/C/ident/bpt_nsaved);

unsigned long [**bpt\_addr**](https://elixir.bootlin.com/linux/latest/C/ident/bpt_addr)[2]; */\* breakpoint handling \*/*

unsigned int [**bpt\_insn**](https://elixir.bootlin.com/linux/latest/C/ident/bpt_insn)[2];

};

*/\**

*\* Macros/functions for gaining access to the thread information structure.*

*\*/*

#define [**INIT\_THREAD\_INFO**](https://elixir.bootlin.com/linux/latest/C/ident/INIT_THREAD_INFO)([**tsk**](https://elixir.bootlin.com/linux/latest/C/ident/tsk)) \

{ \

.[**task**](https://elixir.bootlin.com/linux/latest/C/ident/task) = &[**tsk**](https://elixir.bootlin.com/linux/latest/C/ident/tsk), \

.**[addr\_limit](https://elixir.bootlin.com/linux/latest/C/ident/addr_limit)** = [**KERNEL\_DS**](https://elixir.bootlin.com/linux/latest/C/ident/KERNEL_DS), \

.**[preempt\_count](https://elixir.bootlin.com/linux/latest/C/ident/preempt_count)** = [**INIT\_PREEMPT\_COUNT**](https://elixir.bootlin.com/linux/latest/C/ident/INIT_PREEMPT_COUNT), \

}

*/\* How to get the thread information struct from C. \*/*

[**register**](https://elixir.bootlin.com/linux/latest/C/ident/register) struct [**thread\_info**](https://elixir.bootlin.com/linux/latest/C/ident/thread_info) \*[**\_\_current\_thread\_info**](https://elixir.bootlin.com/linux/latest/C/ident/__current_thread_info) [**\_\_asm\_\_**](https://elixir.bootlin.com/linux/latest/C/ident/__asm__)("$8");

#define [**current\_thread\_info**](https://elixir.bootlin.com/linux/latest/C/ident/current_thread_info)() [**\_\_current\_thread\_info**](https://elixir.bootlin.com/linux/latest/C/ident/__current_thread_info)

#endif */\* \_\_ASSEMBLY\_\_ \*/*

*/\* Thread information allocation. \*/*

#define [**THREAD\_SIZE\_ORDER**](https://elixir.bootlin.com/linux/latest/C/ident/THREAD_SIZE_ORDER) 1

#define [**THREAD\_SIZE**](https://elixir.bootlin.com/linux/latest/C/ident/THREAD_SIZE) (2\*[**PAGE\_SIZE**](https://elixir.bootlin.com/linux/latest/C/ident/PAGE_SIZE))

*/\**

*\* Thread information flags:*

*\* - these are process state flags and used from assembly*

*\* - pending work-to-be-done flags come first and must be assigned to be*

*\* within bits 0 to 7 to fit in and immediate operand.*

*\**

*\* TIF\_SYSCALL\_TRACE is known to be 0 via blbs.*

*\*/*

#define [**TIF\_SYSCALL\_TRACE**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_SYSCALL_TRACE) 0 */\* syscall trace active \*/*

#define [**TIF\_NOTIFY\_RESUME**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_NOTIFY_RESUME) 1 */\* callback before returning to user \*/*

#define [**TIF\_SIGPENDING**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_SIGPENDING) 2 */\* signal pending \*/*

#define [**TIF\_NEED\_RESCHED**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_NEED_RESCHED) 3 */\* rescheduling necessary \*/*

#define [**TIF\_SYSCALL\_AUDIT**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_SYSCALL_AUDIT) 4 */\* syscall audit active \*/*

#define [**TIF\_NOTIFY\_SIGNAL**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_NOTIFY_SIGNAL) 5 */\* signal notifications exist \*/*

#define [**TIF\_DIE\_IF\_KERNEL**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_DIE_IF_KERNEL) 9 */\* dik recursion lock \*/*

#define [**TIF\_MEMDIE**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_MEMDIE) 13 */\* is terminating due to OOM killer \*/*

#define [**TIF\_POLLING\_NRFLAG**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_POLLING_NRFLAG) 14 */\* idle is polling for TIF\_NEED\_RESCHED \*/*

#define [**\_TIF\_SYSCALL\_TRACE**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_SYSCALL_TRACE) (1<<[**TIF\_SYSCALL\_TRACE**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_SYSCALL_TRACE))

#define [**\_TIF\_SIGPENDING**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_SIGPENDING) (1<<[**TIF\_SIGPENDING**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_SIGPENDING))

#define [**\_TIF\_NEED\_RESCHED**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_NEED_RESCHED) (1<<[**TIF\_NEED\_RESCHED**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_NEED_RESCHED))

#define [**\_TIF\_NOTIFY\_RESUME**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_NOTIFY_RESUME) (1<<[**TIF\_NOTIFY\_RESUME**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_NOTIFY_RESUME))

#define [**\_TIF\_SYSCALL\_AUDIT**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_SYSCALL_AUDIT) (1<<[**TIF\_SYSCALL\_AUDIT**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_SYSCALL_AUDIT))

#define [**\_TIF\_NOTIFY\_SIGNAL**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_NOTIFY_SIGNAL) (1<<[**TIF\_NOTIFY\_SIGNAL**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_NOTIFY_SIGNAL))

#define [**\_TIF\_POLLING\_NRFLAG**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_POLLING_NRFLAG) (1<<[**TIF\_POLLING\_NRFLAG**](https://elixir.bootlin.com/linux/latest/C/ident/TIF_POLLING_NRFLAG))

*/\* Work to do on interrupt/exception return. \*/*

#define [**\_TIF\_WORK\_MASK**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_WORK_MASK) ([**\_TIF\_SIGPENDING**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_SIGPENDING) | [**\_TIF\_NEED\_RESCHED**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_NEED_RESCHED) | \

[**\_TIF\_NOTIFY\_RESUME**](https://elixir.bootlin.com/linux/latest/C/ident/_TIF_NOTIFY_RESUME))