LIBMEAS

0.1

Generated by Doxygen 1.5.8

Tue Oct 20 17:29:44 2009

Contents

Chapter 1

Directory Hierarchy

1.1 Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Directory Documentation

3.1 src/ Directory Reference

Files

- file counter.c
- file init.c
- file linkedl.c
- file report.c
- file resources.c
- file time.c

Chapter 4

File Documentation

4.1 src/counter.c File Reference

```
#include <meas.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/syscall.h>
#include <errno.h>
```

Functions

- meas_counter * meas_create_counter (meas_t **mst, unsigned long ivalue, char *name)
  Create a counter.
- unsigned long meas_set_counter (meas_counter *counter, unsigned long value)
  Set the value of a counter.
- unsigned long meas_get_counter (meas_counter counter)
  Return the value of a counter.
- unsigned long meas_inc_counter (meas_counter *counter)
  Increment the value of a counter.
- unsigned long meas_dec_counter (meas_counter *counter)
  Increment the value of a counter.

4.1.1 Function Documentation

4.1.1.1 meas_counter* meas_create_counter (meas_t ** *mst*, unsigned long *ivalue*, char * *name*)

Create a counter.

Parameters:

mst The meas user structure.

ivalue Initial value.

name A name to the counter (useful for report visualization).

Returns:

NULL on error or the created counter.

4.1.1.2 unsigned long meas_dec_counter (meas_counter * *counter*)

Increment the value of a counter.

Parameters:

counter The counter

Returns:

unsigned long New value of the counter

4.1.1.3 unsigned long meas_get_counter (meas_counter *counter*)

Return the value of a counter.

Parameters:

counter The counter

Returns:

unsigned long Value of the counter

4.1.1.4 unsigned long meas_inc_counter (meas_counter * *counter*)

Increment the value of a counter.

Parameters:

counter The counter

Returns:

unsigned long New value of the counter

4.1.1.5 unsigned long meas_set_counter (meas_counter * *counter*, unsigned long *value*)

Set the value of a counter.

Use ONLY for change initial value.

Parameters:

counter The counter

value Value of the counter

Returns:

unsigned long New value of the counter

4.2 src/init.c File Reference

```
#include <meas.h>
#include <stdlib.h>
```

Functions

- __attribute__ ((constructor))
  Global constructor for libmeas internal allocation.
- __attribute__ ((destructor))
  Global destructor for libmeas internal deallocation.
- int meas_init (meas_t **mst)
  Initialize user structures for use libmeas API.
- void meas_close (meas_t **mst)
  Destroy libmeas user structures.

4.2.1 Function Documentation

4.2.1.1 __attribute__ ((destructor))

Global destructor for libmeas internal deallocation.

4.2.1.2 __attribute__ ((constructor))

Global constructor for libmeas internal allocation.

4.2.1.3 void meas_close (meas_t ** *mst*)

Destroy libmeas user structures.

Parameters:

mst The user libmeas structure

See also:

meas_init

4.2.1.4 int meas_init (meas_t ** *mst*)

Initialize user structures for use libmeas API.

Parameters:

mst The user libmeas structure

Returns:

int FALSE on error. True otherwhise

See also:

meas_close

4.3 src/linkedl.c File Reference

```
#include <linkedl.h>
```

Functions

- int llist_create (llist **list)

  Linked list functions For libmeas internal use.

- int llist_destroy (llist **list)

  Destroy a linked list (remove all items).

- int llist_add (llist **list, void *element)

  Add item to the list.

- int llist_remove_nth (llist **list, unsigned int pos)

  Remove the nth item of the list.

- int llist_remove (llist **list, void *element)

  Remove an item from the list.

- void * llist_nth (llist *list, unsigned int index)

  Return the nth element of the list.

- int llist_index (llist *list, void *element)

  Return the index of some element list.

- int llist_length (llist *list)

  Return the number of items inside the list.

4.3.1 Function Documentation

4.3.1.1 int llist_add (llist ** *list*, void * *element*)

Add item to the list.

Parameters:

list Pointer to the list

element Any element (void pointer)

Returns:

FALSE on error. TRUE otherwise

4.3.1.2 int llist_create (llist ** *list*)

Linked list functions For libmeas internal use.

Create a empty linked list

Parameters:

list Pointer to list

Returns:

FALSE on error. TRUE otherwise

4.3.1.3 int llist_destroy (llist ** *list*)

Destroy a linked list (remove all items).

Parameters:

list Pointer to list

Returns:

FALSE on error. TRUE otherwise

4.3.1.4 int llist_index (llist * *list*, void * *element*)

Return the index of some element list.

Parameters:

list List

element Item

Returns:

Element index or -1 (if not found)

4.3.1.5 int llist_length (llist * *list*)

Return the number of items inside the list.

Parameters:

list List

Returns:

The number of items

4.3.1.6 void* llist_nth (llist * *list*, unsigned int *index*)

Return the nth element of the list.

Parameters:

list List

index Element position

Returns:

void* Pointer to the element or NULL if not found

4.3.1.7 int llist_remove (llist ** *list*, void * *element*)

Remove an item from the list.

Parameters:

list Pointer to the list

element Pointer to the item to be removed

Returns:

FALSE on error. TRUE otherwise

4.3.1.8 int llist_remove_nth (llist ** *list*, unsigned int *pos*)

Remove the nth item of the list.

Parameters:

list Pointer to the list

pos Item position at the list

Returns:

FALSE on error. TRUE otherwise

4.4 src/report.c File Reference

```
#include <meas.h>
#include <stdio.h>
#include <string.h>
#include <time.h>
```

Defines

- #define TEXTBUFFER_SIZE 1024

  Size of text buffer.

Functions

- int meas_generate_report (meas_t **mst, int parameters)

  Generate report information.

- void meas_write_report (meas_t *mst, FILE *fp)

  Write report to file descriptor.

4.4.1 Define Documentation

4.4.1.1 #define TEXTBUFFER_SIZE 1024

Size of text buffer.

4.4.2 Function Documentation

4.4.2.1 int meas_generate_report (meas_t ** *mst*, int *parameters*)

Generate report information.

NOTE1: The new report will replace any previous report generated. NOTE2: This function will just generate the report, use meas_show_report to print it.

Parameters:

mst The meas user structure.

parameters Parameters of report (REPORT_TIMERS, REPORT_COUNTERS or REPORT_-SHOW_ALL).

Returns:

FALSE on error, TRUE otherwise.

4.4.2.2 void meas_write_report (meas_t * *mst*, FILE * *fp*)

Write report to file descriptor.

Parameters:

mst The meas user structure.

fp File descriptor pointer.

4.5 src/resources.c File Reference

```
#include <meas.h>
```

Functions

- struct timeval * meas_get_utime (meas_t **mst, int who)
  Return the user time used.
- struct timeval * meas_get_stime (meas_t **mst, int who)
  Return the system time used.
- long meas_get_shared_mem (meas_t **mst, int who)
  Return the integral shared memory size.
- long meas_get_datasec (meas_t **mst, int who)
  Return the integral (unshared) data section size.
- long meas_get_stack (meas_t **mst, int who)
  Return the integral (unshared) stack size.
- long meas_get_pagefault (meas_t **mst, int who)
  Return the number of page faults.
- long meas_get_nswap (meas_t **mst, int who)
  Return the number of pages swapped.
- long meas_get_nsignals (meas_t **mst, int who)
  Return the number of signals received.
- long meas_get_blkinput (meas_t **mst, int who)
  Return the number of block input operations.
- long meas_get_blkoutput (meas_t **mst, int who)
  Return the number of block output operations.
- long meas_get_cswitches (meas_t **mst, int who)
  Return the number context switches (voluntary + involuntary).

4.5.1 Function Documentation

4.5.1.1 long meas_get_blkinput (meas_t ** *mst*, int *who*)

Return the number of block input operations.

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

long -1 on error or number of block input operations.

4.5.1.2 long meas_get_blkoutput (meas_t ∗∗ *mst*, int *who*)

Return the number of block output operations.

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

long -1 on error or number of block output operations.

4.5.1.3 long meas_get_cswitches (meas_t ∗∗ *mst*, int *who*)

Return the number context switches (voluntary + involuntary).

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

long -1 on error or number of total context switches.

4.5.1.4 long meas_get_datasec (meas_t ∗∗ *mst*, int *who*)

Return the integral (unshared) data section size.

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

long -1 on error or integral (unshared) data section size.

4.5.1.5 long meas_get_nsignals (meas_t ** *mst*, int *who*)

Return the number of signals received.

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

long -1 on error or number of signals received.

4.5.1.6 long meas_get_nswap (meas_t ** *mst*, int *who*)

Return the number of pages swapped.

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

long -1 on error or number of pages swapped.

4.5.1.7 long meas_get_pagefault (meas_t ** *mst*, int *who*)

Return the number of page faults.

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

long -1 on error or number of page faults.

4.5.1.8 long meas_get_shared_mem (meas_t ** *mst*, int *who*)

Return the integral shared memory size.

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

long -1 on error or integral shared memory size.

4.5.1.9 long meas_get_stack (meas_t ** *mst*, int *who*)

Return the integral (unshared) stack size.

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

long -1 on error or integral (unshared) stack size.

4.5.1.10 struct timeval* meas_get_stime (meas_t ** *mst*, int *who*) [read]

Return the system time used.

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

struct timeval NULL on error or amount of time that this process has been scheduled in kernel mode, measured in clock ticks (divide by sysconf(_SC_CLK_TCK).

4.5.1.11 struct timeval* meas_get_utime (meas_t ** *mst*, int *who*) [read]

Return the user time used.

Parameters:

mst The meas user structure.

who Can be: RUSAGE_SELF (calling proccess), RUSAGE_CHILDREN (all children) or RUSAGE_THREAD (calling thread).

Returns:

struct timeval NULL on error or amount of time that this process has been scheduled in user mode, measured in clock ticks (divide by sysconf(_SC_CLK_TCK).

4.6 src/time.c File Reference

```
#include <meas.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/syscall.h>
#include <errno.h>
```

Defines

- #define SYS_getjiffies 337
  syscall getjiffies number

Functions

- meas_clock * meas_start_clock (meas_t **mst, meas_clock *clock, char *name)
  Start or/and create a timer.
- int meas_stop_clock (meas_clock *clock)
  Stop a timer.

4.6.1 Define Documentation

4.6.1.1 #define SYS_getjiffies 337

syscall getjiffies number

4.6.2 Function Documentation

4.6.2.1 meas_clock* meas_start_clock (meas_t ** *mst*, meas_clock * *clock*, char * *name*)

Start or/and create a timer.

Parameters:

mst The meas user structure. This argument is necessary only in the first call to create the clock (second argument will be NULL). After that, you can just pass NULL to mst and pass the clock in second argument.

clock The clock created with this function. Use NULL in the first call.

name A name to the clock (useful for report visualization).

Returns:

NULL if both mst and clock are different of NULL or the created clock.

4.6.2.2 int meas_stop_clock (meas_clock * *clock*)

Stop a timer.

Parameters:

clock The timer

Returns:

TRUE if the timer was stopped or FALSE if the timer was already stopped.

Index