NAME

sg_get_process_stats, sg_get_process_count, sg_process_compare_name, sg_process_compare_pid, sg_process_compare_uid, sg_process_compare_gid, sg_process_compare_size, sg_process_compare_res, sg_process_compare_cpu, sg_process_compare_time − get process statistics

SYNOPSIS

#include <statgrab.h>

sg_process_stats **sg_get_process_stats*(int **entries*);

sg_process_count **sg_get_process_count*(void);

int *sg_process_compare_name*(const void **va*, const void **vb*);

int *sg_process_compare_pid*(const void **va*, const void **vb*);

int *sg_process_compare_uid*(const void **va*, const void **vb*);

int *sg_process_compare_gid*(const void **va*, const void **vb*);

int *sg_process_compare_size*(const void **va*, const void **vb*);

int *sg_process_compare_res*(const void **va*, const void **vb*);

int *sg_process_compare_cpu*(const void **va*, const void **vb*);

int *sg_process_compare_time*(const void **va*, const void **vb*);

DESCRIPTION

The sg_get_process_stats takes a pointer to an int, entries, which is filled with the number of processes the snapshot contains. The return value is a pointer to the first member of an array of sg_process_stats structures; the number of entries in the array is returned in entries.

sg_get_process_count returns a pointer to a static buffer of type sg_process_count.

These two functions provide information on the process running on the system. In the first case lots of detail is provided, and in the second case a summary of the states of the current processes.

sg_process_compare_name

sg_process_compare_pid

sg_process_compare_uid

sg_process_compare_gid

sg_process_compare_size

sg_process_compare_res

sg_process_compare_cpu

sg_process_compare_time

These functions compare two sg_process_stats entries, and return an int to represent which one is greater. The main use of these functions is to be passed to qsort to sort the sg_process_stats by the given type.

For example:

qsort(sg_process_stats ps, int ps_size, sizeof *ps, sg_process_compare_pid);

RETURN VALUES

The structure returned by sg_get_process_stats is of type sg_process_stats.

```
typedef struct {
        char *process_name;
        char *proctitle;
        pid_t pid;
        pid_t parent; /* Parent pid */
        pid_t pgid;   /* process id of process group leader */

        uid_t uid;
        uid_t euid;
        gid_t gid;
        gid_t egid;

        unsigned long long proc_size; /* in bytes */
        unsigned long long proc_resident; /* in bytes */
        time_t time_spent; /* time running in seconds */
        double cpu_percent;
        int nice;
        sg_process_state state;
} sg_process_stats;



typedef enum {
        SG_PROCESS_STATE_RUNNING,
        SG_PROCESS_STATE_SLEEPING,
        SG_PROCESS_STATE_STOPPED,
        SG_PROCESS_STATE_ZOMBIE,
        SG_PROCESS_STATE_UNKNOWN
} sg_process_state;
```

process_name
The name of the command that was run.

proctitle
The command line (the "title") of the process. Take note − this can be modified by the process, so isn't guaranteed to be the original command line.

pid The process ID.

parent The parent process ID.

pgid The process ID of the process group leader.

uid The UID the process is running as.

euid The effective UID the process is running as.

gid The GID the process is running as.

egid The effective GID the process is running as.

proc_size
The size of the process in bytes.

proc_resident
The size of the process that's resident in memory.

time_spent
The number of seconds the process has been running.

cpu_percent
The current percentage of CPU the process is using.

nice The nice value of the process.

state The current state of the process. See sg_process_state for permitted values.

The structure returned by sg_get_process_count is of type sg_process_count.

```
typedef struct{
        int total;
        int running;
        int sleeping;
        int stopped;
        int zombie;
}sg_process_count;
```

total The total number of processes.

running
The number of running processes.

sleeping
The number of sleeping processes.

stopped
The number of stopped processes.

zombie The number of zombie processes.

SEE ALSO

statgrab(3)

WEBSITE

http://www.i−scream.org/libstatgrab/