NAME

sg_get_cpu_stats, sg_get_cpu_stats_diff, sg_get_cpu_percents − get cpu usage

SYNOPSIS

#include <statgrab.h>

sg_cpu_percents *sg_get_cpu_percents(void);

sg_cpu_stats *sg_get_cpu_stats(void);

sg_cpu_stats *sg_get_cpu_stats_diff(void);

DESCRIPTION

sg_get_cpu_stats() and sg_get_cpu_stats_diff() both return static pointers of type sg_cpu_stats. sg_get_cpu_stats() returns the total amount of "ticks" the operating system has spent in each of the different states. sg_get_cpu_stats_diff() returns the difference in "ticks" for each of the states since last time sg_get_cpu_stats_diff() or sg_get_cpu_stats() was called. If it has never been called, it will return the result of sg_get_cpu_stats().

The value stored (the "ticks") will vary between operating systems. For example Solaris has a total of 100 per second, while Linux has substantially more. Also, different operating systems store different information − you won't find nice cpu on Solaris for example.

Because of this, you will ideally always want to work on a scale against the total, or in percentages.

sg_get_cpu_percents() returns a pointer to a static sg_cpu_percents. The function calls sg_get_cpu_stats_diff() and changes the values into percentages. If it has never been called before (and nor has sg_get_cpu_stats() or sg_get_cpu_stats_diff()), the returned percentages will be the systems total ever since its uptime. (Unless the counters have cycled)

RETURN VALUES

There are two structures returned by the CPU statistics functions.

```
typedef struct{
    long long user;
    long long kernel;
    long long idle;
    long long iowait;
    long long swap;
    long long nice;
    long long total;
    time_t systime;
}sg_cpu_stats;
```

```
typedef struct{
    float user;
    float kernel;
    float idle;
    float iowait;
    float swap;
    float nice;
    time_t time_taken;
}sg_cpu_percents;
```

user kernel idle iowait swap nice
: The different CPU states.

systime time_taken
The time taken in seconds since the last call of the function, or the system time.

SEE ALSO

statgrab(3)

WEBSITE

http://www.i−scream.org/libstatgrab/