Unix::Statgrab - System Monitoring

Jens Rehsack

2013

Overview

Part I

Introduction

Audience

Audience

- Developer who wants to create or improve monitoring software
- Developer who wants to evaluate system stats for content sensitive code paths
- Developer who wants to to learn the difference to earlier libstatgrab / Unix::Statgrab API
- Developers or Operators (Admins) who wants to learn about measurement of statistic values of the machine

Audience

Audience

- Developer who wants to create or improve monitoring software
- Developer who wants to evaluate system stats for content sensitive code paths
- Developer who wants to to learn the difference to earlier libstatgrab / Unix::Statgrab API
- Developers or Operators (Admins) who wants to learn about measurement of statistic values of the machine

Prerequisites of the Audience

Following knowledge is expected:

- advanced skills in at least one object oriented and procedural programming language
- more than one year practical experience in object oriented development
- Experience with Unix or compatible operating systems
- slightly above basic Perl experience

Motivation

XS / C

- use of native API to get OS stats
- performance advantage
- interoperability (most VM's have a * native interface)
- portability - widest calling convention support in C

Platforms I

Tested and confirmed running

- DragonFly BSD 3.4
- FreeBSD 7,8 (i386, amd64), FreeBSD 9 (i386, amd64, sparc64, ia64), FreeBSD 10-CURRENT (i386, amd64, sparc64, ia64)
- HP-UX 11.11 (parisc) HP-UX 11.23 (parisc, ia64), HP-UX 11.31 (ia64)
- Linux 2.6 (Ubuntu 10.04, SLES 9-11, Redhat 6, CentOS 6, µCLinux/arm7), Linux 3.X (Ubuntu 12.04)
- MacOS X 10.6, 10.8 (amd64)
- NetBSD 5.1-6.1 (amd64), NetBSD-CURRENT (amd64)
- OpenBSD 4.9, 5.3 (amd64)
- Solaris 8,9 (sparc), Solaris 10 (sparc, x86 & amd64), Solaris 11 (amd64)
- AIX 5.2, 5.3, 6.1 (ppc64)

Platforms II

in progress ...

- Windows (using Interix, maybe mSys)
- kFreeBSD
- Hurd

Platforms II

in progress ...

- Windows (using Interix, maybe mSys)
- kFreeBSD
- Hurd

Wishlist

- Digital Unix / Tru64 / OSF1
- Haiku
- VMS
- zOS

Part II

libstatgrab

Host Info

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```

Host Info

bundles some operating system information as

- name (Linux, FreeBSD, AIX),

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```

Host Info

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```

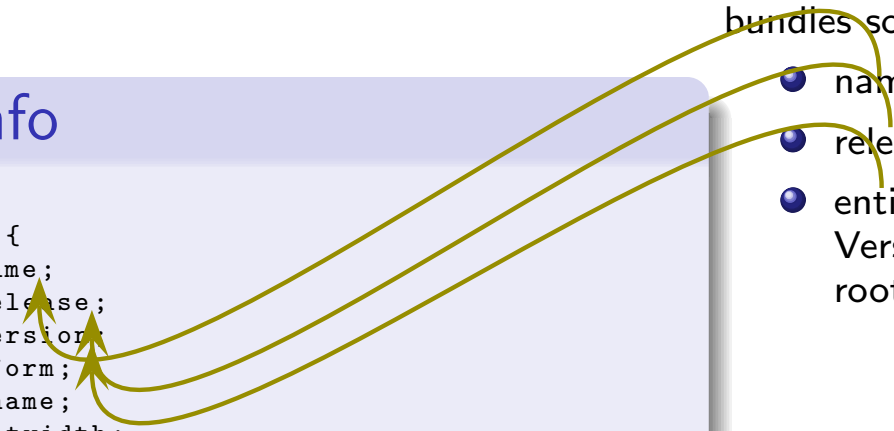
bundles some operating system information as

- name (Linux, FreeBSD, AIX),
- release (eg. kernel version),

Host Info

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```



bundles some operating system information as

- name (Linux, FreeBSD, AIX),
- release (eg. kernel version),
- entire OS version string (eg. Darwin Kernel Version 12.4.0: Wed May 1 17:57:12 PDT 2013; root:xnu-2050.24.15 1/RELEASE_X86_64),

Host Info

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```

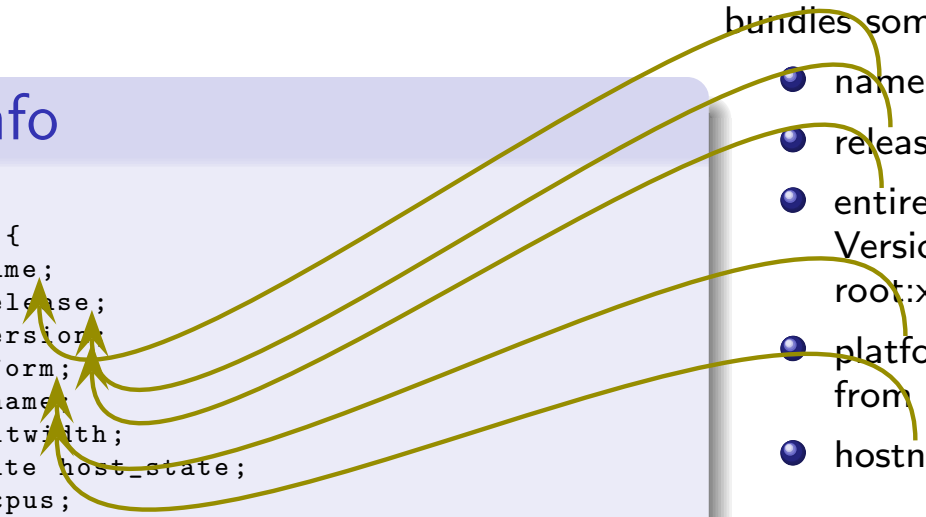
bundles some operating system information as

- name (Linux, FreeBSD, AIX),
- release (eg. kernel version),
- entire OS version string (eg. Darwin Kernel Version 12.4.0: Wed May 1 17:57:12 PDT 2013; root:xnu-2050.24.15 1/RELEASE_X86_64),
- platform, what finally means CPU information from OS perspective,

Host Info

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```



bundles some operating system information as

- name (Linux, FreeBSD, AIX),
- release (eg. kernel version),
- entire OS version string (eg. Darwin Kernel Version 12.4.0: Wed May 1 17:57:12 PDT 2013; root:xnu-2050.24.15 1/RELEASE_X86_64),
- platform, what finally means CPU information from OS perspective,
- hostname name of the host.

Host Info

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```

bundles some operating system information as

- name (Linux, FreeBSD, AIX),
- release (eg. kernel version),
- entire OS version string (eg. Darwin Kernel Version 12.4.0: Wed May 1 17:57:12 PDT 2013; root:xnu-2050.24.15 1/RELEASE_X86_64),
- platform, what finally means CPU information from OS perspective,
- hostname name of the host.
- bitwidth (usually 32 or 64),

Host Info

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```

bundles some operating system information as

- name (Linux, FreeBSD, AIX),
- release (eg. kernel version),
- entire OS version string (eg. Darwin Kernel Version 12.4.0: Wed May 1 17:57:12 PDT 2013; root:xnu-2050.24.15 1/RELEASE_X86_64),
- platform, what finally means CPU information from OS perspective,
- hostname name of the host.
- bitwidth (usually 32 or 64),
- host state - one of `sg_physical_host`, `sg_virtual_machine`, `sg_paravirtual_machine`, `sg_hardware_virtualized` or `sg_unknown_configuration`),

Host Info

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```

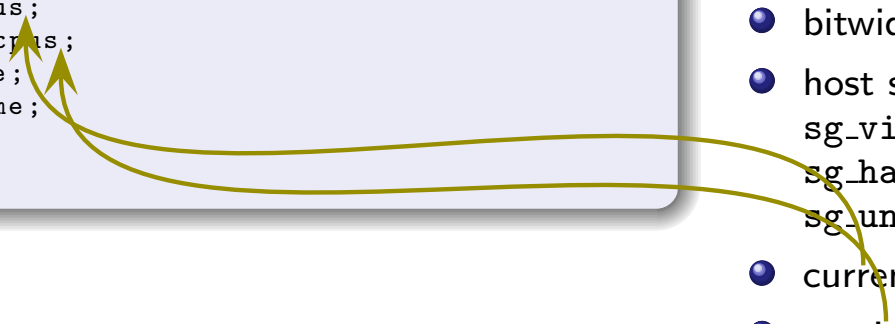
bundles some operating system information as

- name (Linux, FreeBSD, AIX),
- release (eg. kernel version),
- entire OS version string (eg. Darwin Kernel Version 12.4.0: Wed May 1 17:57:12 PDT 2013; root:xnu-2050.24.15 1/RELEASE_X86_64),
- platform, what finally means CPU information from OS perspective,
- hostname name of the host.
- bitwidth (usually 32 or 64),
- host state - one of `sg_physical_host`, `sg_virtual_machine`, `sg_paravirtual_machine`, `sg_hardware_virtualized` or `sg_unknown_configuration`),
- current number of CPU's,

Host Info

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```



bundles some operating system information as

- name (Linux, FreeBSD, AIX),
- release (eg. kernel version),
- entire OS version string (eg. Darwin Kernel Version 12.4.0: Wed May 1 17:57:12 PDT 2013; root:xnu-2050.24.15 1/RELEASE_X86_64),
- platform, what finally means CPU information from OS perspective,
- hostname name of the host.
- bitwidth (usually 32 or 64),
- host state - one of `sg_physical_host`, `sg_virtual_machine`, `sg_paravirtual_machine`, `sg_hardware_virtualized` or `sg_unknown_configuration`),
- current number of CPU's,
- maximum number of CPU's.

Host Info

sg_host_info

```
typedef struct {
    char *os_name;
    char *os_release;
    char *os_version;
    char *platform;
    char *hostname;
    unsigned bitwidth;
    sg_host_state host_state;
    unsigned ncpus;
    unsigned maxcpus;
    time_t uptime;
    time_t systime;
} sg_host_info;
```

bundles some operating system information as

- name (Linux, FreeBSD, AIX),
- release (eg. kernel version),
- entire OS version string (eg. Darwin Kernel Version 12.4.0: Wed May 1 17:57:12 PDT 2013; root:xnu-2050.24.15 1/RELEASE_X86_64),
- platform, what finally means CPU information from OS perspective,
- hostname name of the host.
- bitwidth (usually 32 or 64),
- host state - one of `sg_physical_host`, `sg_virtual_machine`, `sg_paravirtual_machine`, `sg_hardware_virtualized` or `sg_unknown_configuration`),
- current number of CPU's,
- maximum number of CPU's.
- timestamp when collected this stats.

CPU stats

sg_cpu_stats

```
typedef struct {
    unsigned long long user, kernel, idle, iowait, swap, nice, total;

    unsigned long long context_switches, voluntary_context_switches, involuntary_context_switches,
                       syscalls, interrupts, soft_interrupts,

    time_t systime;
} sg_cpu_stats;
```

CPU stats

sg_cpu_stats

```
typedef struct {
    unsigned long long user, kernel, idle, iowait, swap, nice, total;

    unsigned long long context_switches, voluntary_context_switches, involuntary_context_switches,
                       syscalls, interrupts, soft_interrupts,

    time_t systime;
} sg_cpu_stats;
```

- absolute ticks of measurable CPU states

CPU stats

sg_cpu_stats

```
typedef struct {
    unsigned long long user, kernel, idle, iowait, swap, nice, total;

    unsigned long long context_switches, voluntary_context_switches, involuntary_context_switches,
                       syscalls, interrupts, soft_interrupts,

    time_t systime;
} sg_cpu_stats;
```

- absolute ticks of measurable CPU states
- context switches over all CPU's, also separated by voluntary and involuntary

CPU stats

sg_cpu_stats

```
typedef struct {
    unsigned long long user, kernel, idle, iowait, swap, nice, total;

    unsigned long long context_switches, voluntary_context_switches, involuntary_context_switches,
                       syscalls, interrupts, soft_interrupts,

    time_t systime;
} sg_cpu_stats;
```

- absolute ticks of measurable CPU states,
- context switches over all CPU's, also separated by voluntary and involuntary,
- syscalls made, interrupts and soft-interrupts occured.

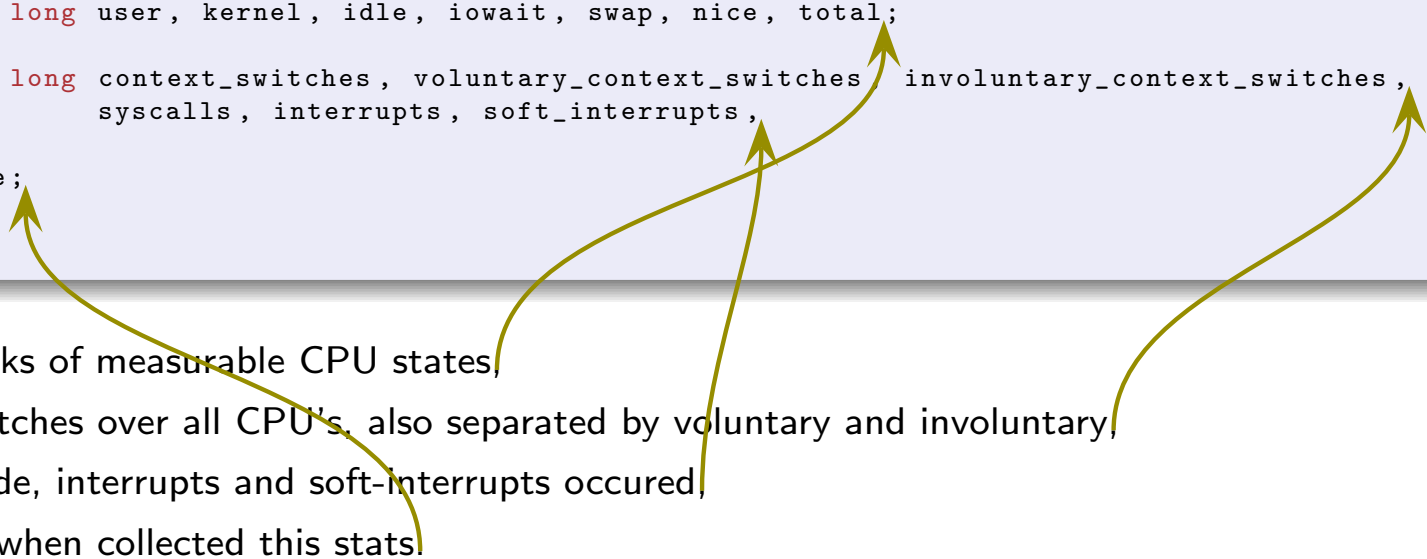
CPU stats

sg_cpu_stats

```
typedef struct {
    unsigned long long user, kernel, idle, iowait, swap, nice, total;

    unsigned long long context_switches, voluntary_context_switches, involuntary_context_switches,
                       syscalls, interrupts, soft_interrupts,

    time_t systime;
} sg_cpu_stats;
```



- absolute ticks of measurable CPU states,
- context switches over all CPU's, also separated by voluntary and involuntary,
- syscalls made, interrupts and soft-interrupts occured,
- timestamp when collected this stats.

CPU percents

sg_cpu_percents

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

CPU percents

sg_cpu_percents

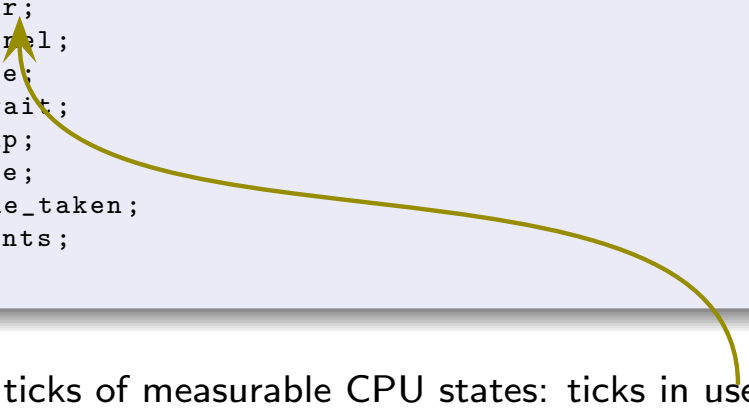
```
typedef struct {
    double user;
    double kernel;
    double idle;
    double iowait;
    double swap;
    double nice;
    time_t time_taken;
} sg_cpu_percents;
```

- relative ticks of measurable CPU states:

CPU percents

sg_cpu_percents

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



- relative ticks of measurable CPU states: ticks in user mode,

CPU percents

sg_cpu_percents

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

- relative ticks of measurable CPU states: ticks in user mode, kernel mode,

CPU percents

sg_cpu_percents

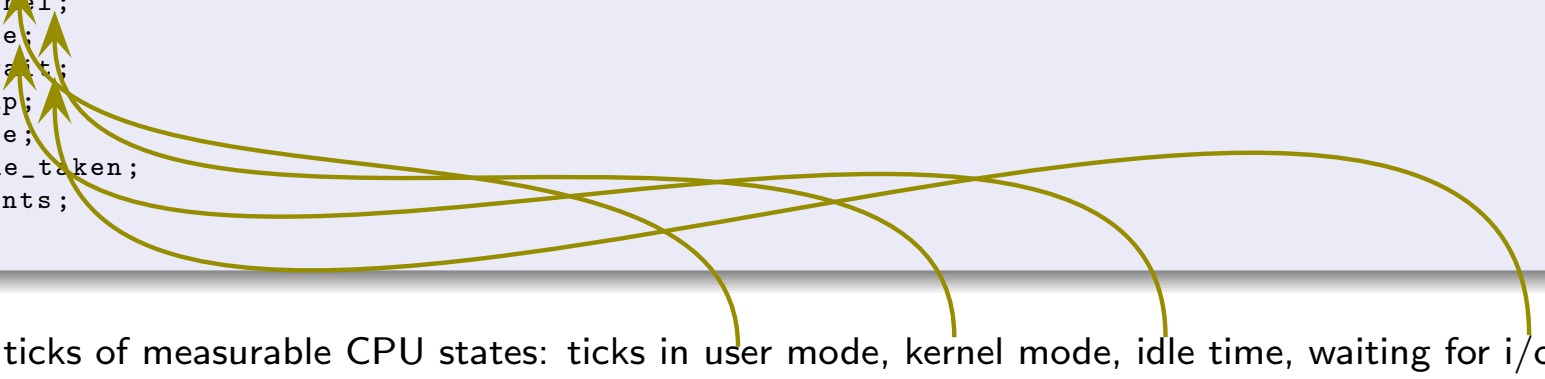
```
typedef struct {
    double user;
    double kernel;
    double idle;
    double iowait;
    double swap;
    double nice;
    time_t time_taken;
} sg_cpu_percents;
```

- relative ticks of measurable CPU states: ticks in user mode, kernel mode, idle time,

CPU percents

sg_cpu_percents

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

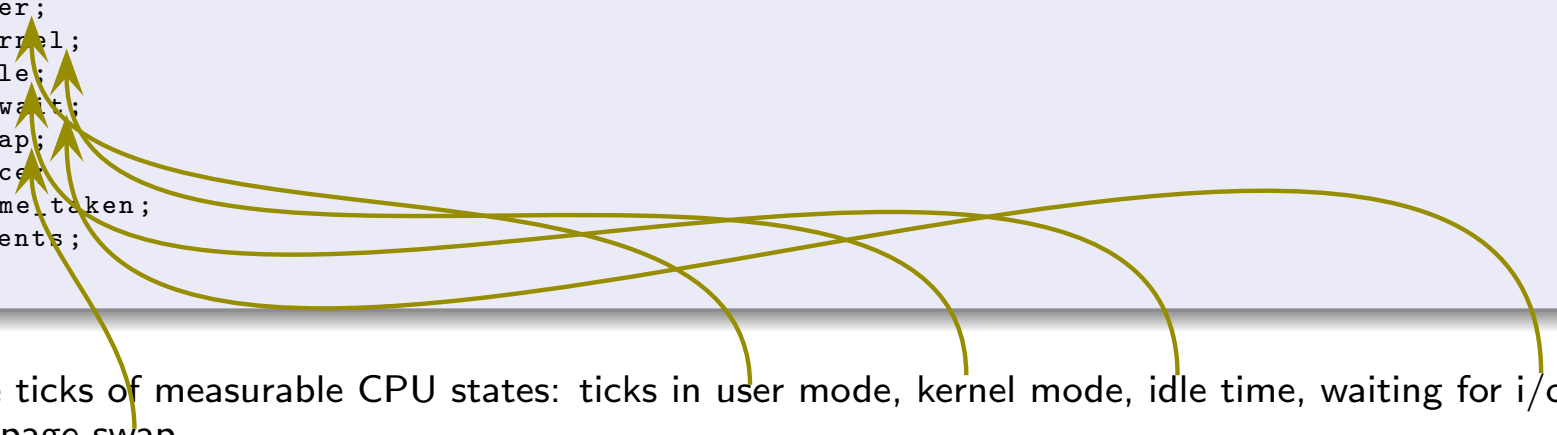


- relative ticks of measurable CPU states: ticks in user mode, kernel mode, idle time, waiting for i/o,

CPU percents

sg_cpu_percents

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

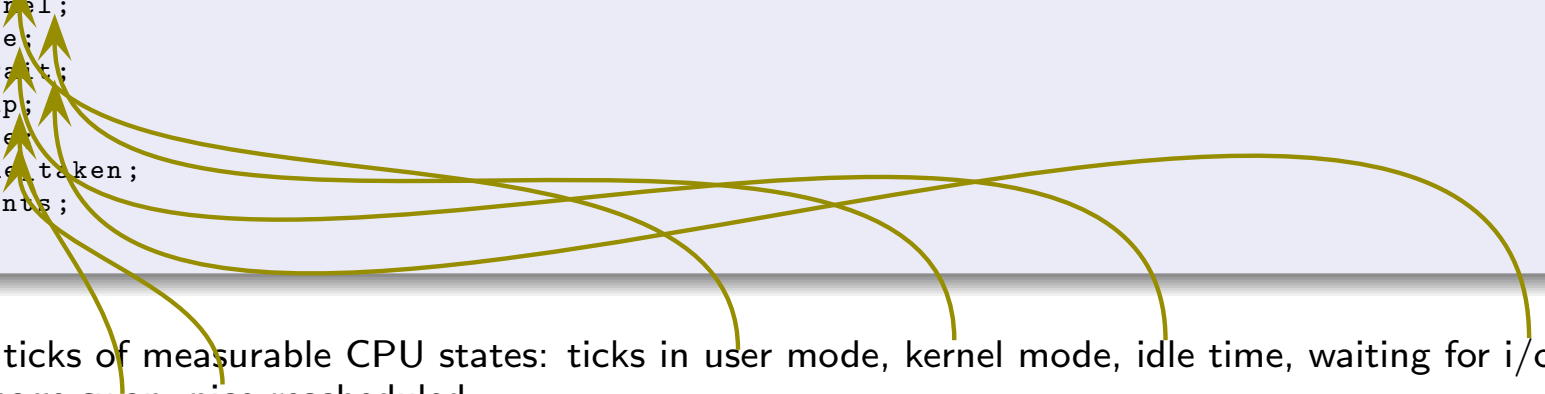


- relative ticks of measurable CPU states: ticks in user mode, kernel mode, idle time, waiting for i/o, during page swap,

CPU percents

sg_cpu_percents

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

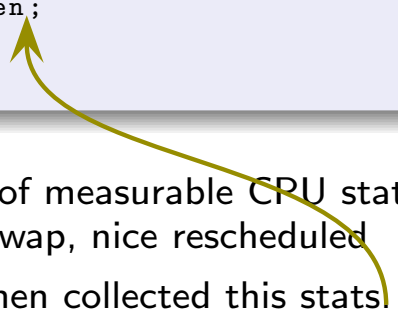


- relative ticks of measurable CPU states: ticks in user mode, kernel mode, idle time, waiting for i/o, during page swap, nice rescheduled

CPU percents

sg_cpu_percents

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



- relative ticks of measurable CPU states: ticks in user mode, kernel mode, idle time, waiting for i/o, during page swap, nice rescheduled
- timestamp when collected this stats

Load percents

sg_load_stats

```
typedef struct {
    double min1;
    double min5;
    double min15;
    time_t systime;
} sg_load_stats;
```

Load percents

sg_load_stats

```
typedef struct {
    double min1;
    double min5;
    double min15;
    time_t systime;
} sg_load_stats;
```

- percentage of cpu usage per

Load percents

sg_load_stats

```
typedef struct {
    double min1;
    double min5;
    double min15;
    time_t systime;
} sg_load_stats;
```

- percentage of cpu usage per 1 minute,

Load percents

sg_load_stats

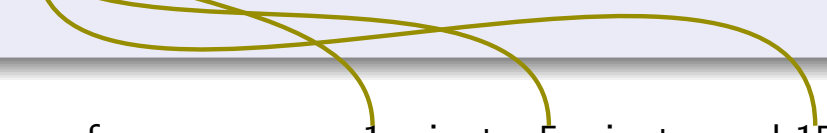
```
typedef struct {
    double min1;
    double min5;
    double min15;
    time_t systime;
} sg_load_stats;
```

- percentage of cpu usage per 1 minute, 5 minutes and

Load percents

sg_load_stats

```
typedef struct {
    double min1;
    double min5;
    double min15;
    time_t systime;
} sg_load_stats;
```

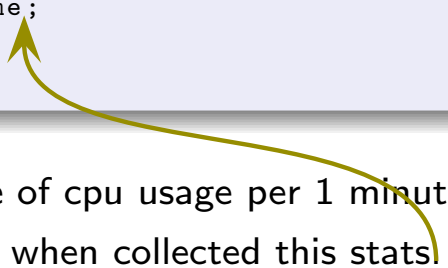


- percentage of cpu usage per 1 minute, 5 minutes and 15 minutes.

Load percents

sg_load_stats

```
typedef struct {
    double min1;
    double min5;
    double min15;
    time_t systime;
} sg_load_stats;
```



- percentage of cpu usage per 1 minute, 5 minutes and 15 minutes.
- timestamp when collected this stats.

Memory stats

sg_mem_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    unsigned long long cache;
    time_t systime;
} sg_mem_stats;
```

Memory stats

sg_mem_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    unsigned long long cache;
    time_t systime;
} sg_mem_stats;
```

- information about main memory of the system:

Memory stats

sg_mem_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    unsigned long long cache;
    time_t systime;
} sg_mem_stats;
```

- information about main memory of the system: total,

Memory stats

sg_mem_stats

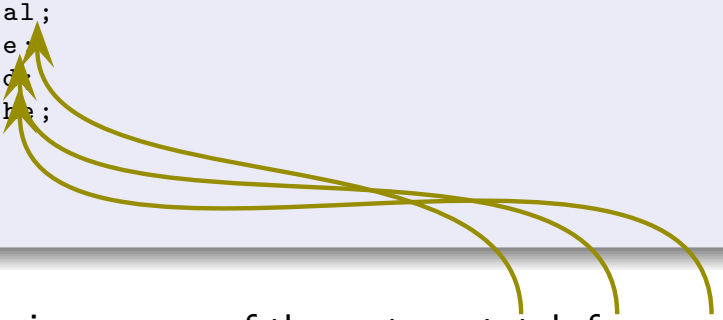
```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    unsigned long long cache;
    time_t systime;
} sg_mem_stats;
```

- information about main memory of the system: total, free,

Memory stats

sg_mem_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    unsigned long long cache;
    time_t systime;
} sg_mem_stats;
```



- information about main memory of the system: total, free, used,

Memory stats

sg_mem_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    unsigned long long cache;
    time_t systime;
} sg_mem_stats;
```

- information about main memory of the system: total, free, used, cache,

Memory stats

sg_mem_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    unsigned long long cache;
    time_t systime;
} sg_mem_stats;
```

- information about main memory of the system: total, free, used, cache,
- timestamp when collected this stats.

Swap stats

sg_swap_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    time_t systime;
} sg_swap_stats;
```

Swap stats

sg_swap_stats

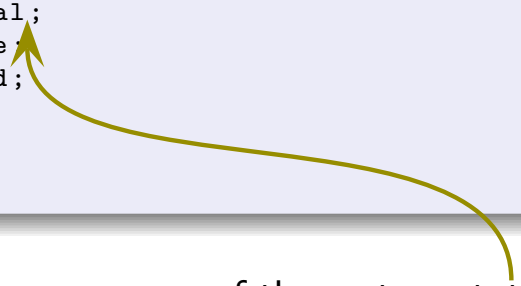
```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    time_t systime;
} sg_swap_stats;
```

- information about swap memory of the system:

Swap stats

sg_swap_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    time_t systime;
} sg_swap_stats;
```

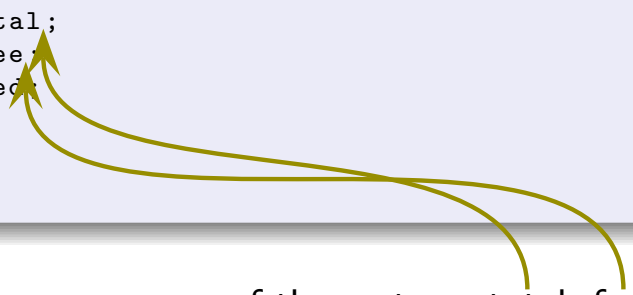


- information about swap memory of the system: total,

Swap stats

sg_swap_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    time_t systime;
} sg_swap_stats;
```



- information about swap memory of the system: total, free,

Swap stats

sg_swap_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    time_t systime;
} sg_swap_stats;
```

- information about swap memory of the system: total, free, used,

Swap stats

sg_swap_stats

```
typedef struct {
    unsigned long long total;
    unsigned long long free;
    unsigned long long used;
    time_t systime;
} sg_swap_stats;
```

- information about swap memory of the system: total, free, used,
- timestamp when collected this stats!

Disk I/O stats

sg_disk_io_stats

```
typedef struct {
    char *disk_name;
    unsigned long long read_bytes;
    unsigned long long write_bytes;
    time_t systime;
} sg_disk_io_stats;
```

Disk I/O stats

sg_disk_io_stats

```
typedef struct {
    char *disk_name;
    unsigned long long read_bytes;
    unsigned long long write_bytes;
    time_t systime;
} sg_disk_io_stats;
```

for each block device known to the system

- name of the block device,

Disk I/O stats

sg_disk_io_stats

```
typedef struct {
    char *disk_name;
    unsigned long long read_bytes;
    unsigned long long write_bytes;
    time_t systime;
} sg_disk_io_stats;
```

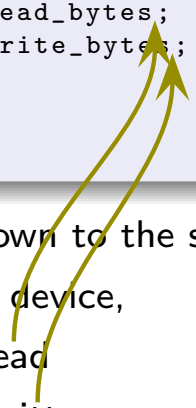
for each block device known to the system

- name of the block device,
- amount of bytes read

Disk I/O stats

sg_disk_io_stats

```
typedef struct {
    char *disk_name;
    unsigned long long read_bytes;
    unsigned long long write_bytes;
    time_t systime;
} sg_disk_io_stats;
```



for each block device known to the system

- name of the block device,
- amount of bytes read
- amount of bytes written

Disk I/O stats

sg_disk_io_stats

```
typedef struct {
    char *disk_name;
    unsigned long long read_bytes;
    unsigned long long write_bytes;
    time_t systime;
} sg_disk_io_stats;
```

for each block device known to the system

- name of the block device,
- amount of bytes read
- amount of bytes written
- timestamp when collected this stats.

Paging stats

sg_page_stats

```
typedef struct {
    unsigned long long pages_pagein;
    unsigned long long pages_pageout;
    time_t systime;
} sg_page_stats;
```

Paging stats

sg_page_stats

```
typedef struct {
    unsigned long long pages_pagein;
    unsigned long long pages_pageout;
    time_t systime;
} sg_page_stats;
```

for entire system

- amount of bytes paged in,

Paging stats

sg_page_stats

```
typedef struct {
    unsigned long long pages_pagein;
    unsigned long long pages_pageout;
    time_t systime;
} sg_page_stats;
```

for entire system

- amount of bytes paged in,
- amount of bytes paged out

Paging stats

sg_page_stats

```
typedef struct {
    unsigned long long pages_pagein;
    unsigned long long pages_pageout;
    time_t systime;
} sg_page_stats;
```

for entire system

- amount of bytes paged in,
- amount of bytes paged out
- timestamp when collected this stats!

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

Filesystem stats

for each mounted (and not filtered) file system

- name of the mounted block device,

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point
- device type: one of `sg_fs_unknown`, `sg_fs_regular`, `sg_fs_special`, `sg_fs_loopback`, `sg_fs_remote` or any combination
  Anything but unknown is covered by `sg_fs_alltypes`, any local type by `sg_fs_local`

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point
- device type: one of `sg_fs_unknown`, `sg_fs_regular`, `sg_fs_special`, `sg_fs_loopback`, `sg_fs_remote` or any combination
  Anything but unknown is covered by `sg_fs_alltypes`, any local type by `sg_fs_local`
- size of the file system in bytes

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point
- device type: one of `sg_fs_unknown`, `sg_fs_regular`, `sg_fs_special`, `sg_fs_loopback`, `sg_fs_remote` or any combination
  Anything but unknown is covered by `sg_fs_alltypes`, any local type by `sg_fs_local`
- size of the file system in bytes
- also separated into used, free and avail

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point
- device type: one of `sg_fs_unknown`, `sg_fs_regular`, `sg_fs_special`, `sg_fs_loopback`, `sg_fs_remote` or any combination
  Anything but unknown is covered by `sg_fs_alltypes`, any local type by `sg_fs_local`
- size of the file system in bytes
- also separated into used, free and avail
- inodes of the file system

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point
- device type: one of `sg_fs_unknown`, `sg_fs_regular`, `sg_fs_special`, `sg_fs_loopback`, `sg_fs_remote` or any combination
  Anything but unknown is covered by `sg_fs_alltypes`, any local type by `sg_fs_local`
- size of the file system in bytes
- also separated into used, free and avail
- inodes of the file system
- also separated into used, free and avail

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point
- device type: one of `sg_fs_unknown`, `sg_fs_regular`, `sg_fs_special`, `sg_fs_loopback`, `sg_fs_remote` or any combination
  Anything but unknown is covered by `sg_fs_alltypes`, any local type by `sg_fs_local`
- size of the file system in bytes
- also separated into used, free and avail
- inodes of the file system
- also separated into used, free and avail
- optimal size of the I/O blocks when accessing the file system in bytes

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point
- device type: one of `sg_fs_unknown`, `sg_fs_regular`, `sg_fs_special`, `sg_fs_loopback`, `sg_fs_remote` or any combination
  Anything but unknown is covered by `sg_fs_alltypes`, any local type by `sg_fs_local`
- size of the file system in bytes
- also separated into used, free and avail
- inodes of the file system
- also separated into used, free and avail
- optimal size of the I/O blocks when accessing the file system in bytes
- block size (minimum allocation size) of the file system in bytes

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point
- device type: one of `sg_fs_unknown`, `sg_fs_regular`, `sg_fs_special`, `sg_fs_loopback`, `sg_fs_remote` or any combination
  Anything but unknown is covered by `sg_fs_alltypes`, any local type by `sg_fs_local`
- size of the file system in bytes
- also separated into used, free and avail
- inodes of the file system
- also separated into used, free and avail
- optimal size of the I/O blocks when accessing the file system in bytes
- block size (minimum allocation size) of the file system in bytes
- amount of blocks of the file system

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point
- device type: one of `sg_fs_unknown`, `sg_fs_regular`, `sg_fs_special`, `sg_fs_loopback`, `sg_fs_remote` or any combination
  Anything but unknown is covered by `sg_fs_alltypes`, any local type by `sg_fs_local`
- size of the file system in bytes
- also separated into used, free and avail
- inodes of the file system
- also separated into used, free and avail
- optimal size of the I/O blocks when accessing the file system in bytes
- block size (minimum allocation size) of the file system in bytes
- amount of blocks of the file system
- also separated into used, free and avail

Filesystem stats

sg_fs_stats

```
typedef struct {
    char *device_name;
    char *fs_type;
    char *mnt_point;
    sg_fs_device_type device_type;
    unsigned long long size;
    unsigned long long used;
    unsigned long long free;
    unsigned long long avail;
    unsigned long long total_inodes;
    unsigned long long used_inodes;
    unsigned long long free_inodes;
    unsigned long long avail_inodes;
    unsigned long long io_size;
    unsigned long long block_size;
    unsigned long long total_blocks;
    unsigned long long free_blocks;
    unsigned long long used_blocks;
    unsigned long long avail_blocks;
    time_t systime;
} sg_fs_stats;
```

for each mounted (and not filtered) file system

- name of the mounted block device,
- name of the file system type (eg. ext3, ffs, zfs)
- full qualified path name of the mount point
- device type: one of `sg_fs_unknown`, `sg_fs_regular`, `sg_fs_special`, `sg_fs_loopback`, `sg_fs_remote` or any combination
  Anything but unknown is covered by `sg_fs_alltypes`, any local type by `sg_fs_local`
- size of the file system in bytes
- also separated into used, free and avail
- inodes of the file system
- also separated into used, free and avail
- optimal size of the I/O blocks when accessing the file system in bytes
- block size (minimum allocation size) of the file system in bytes
- amount of blocks of the file system
- also separated into used, free and avail
- timestamp when collected this stats

User stats

sg_user_stats

```
typedef struct {
    char *login_name;
    char *record_id;
    size_t record_id_size;
    char *device;
    char *hostname;
    pid_t pid;
    time_t login_time;
    time_t systime;
} sg_user_stats;
```

User stats

sg_user_stats

```
typedef struct {
    char *login_name;
    char *record_id;
    size_t record_id_size;
    char *device;
    char *hostname;
    pid_t pid;
    time_t login_time;
    time_t systime;
} sg_user_stats;
```

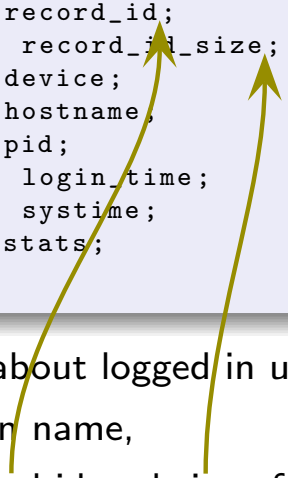
statistics about logged in users, as

- login name,

User stats

sg_user_stats

```
typedef struct {
    char *login_name;
    char *record_id;
    size_t record_id_size;
    char *device;
    char *hostname;
    pid_t pid;
    time_t login_time;
    time_t systime;
} sg_user_stats;
```



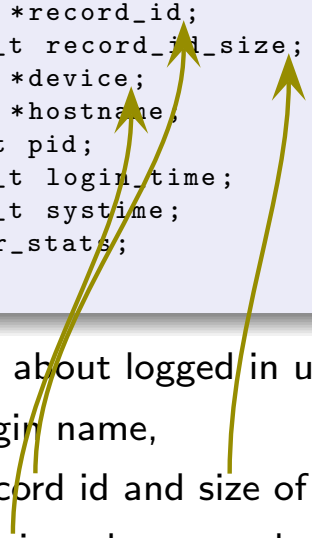
statistics about logged in users, as

- login name,
- record id and size of that field (not '\0' terminated),

User stats

sg_user_stats

```
typedef struct {
    char *login_name;
    char *record_id;
    size_t record_id_size;
    char *device;
    char *hostname;
    pid_t pid;
    time_t login_time;
    time_t systime;
} sg_user_stats;
```



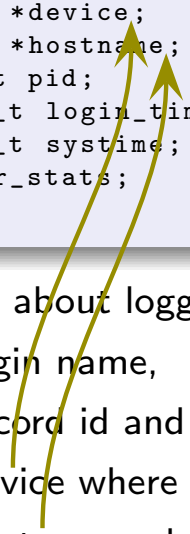
statistics about logged in users, as

- login name,
- record id and size of that field (not '\0' terminated),
- device where user logged in,

User stats

sg_user_stats

```
typedef struct {
    char *login_name;
    char *record_id;
    size_t record_id_size;
    char *device;
    char *hostname;
    pid_t pid;
    time_t login_time;
    time_t systime;
} sg_user_stats;
```



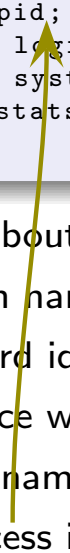
statistics about logged in users, as

- login name,
- record id and size of that field (not '\0' terminated),
- device where user logged in,
- hostname when remote login

User stats

sg_user_stats

```
typedef struct {
    char *login_name;
    char *record_id;
    size_t record_id_size;
    char *device;
    char *hostname;
    pid_t pid;
    time_t login_time;
    time_t systime;
} sg_user_stats;
```



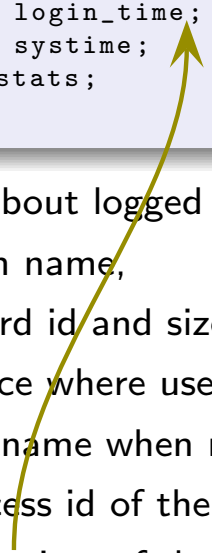
statistics about logged in users, as

- login name,
- record id and size of that field (not '\0' terminated),
- device where user logged in,
- hostname when remote login
- process id of the session's "root" process

User stats

sg_user_stats

```
typedef struct {
    char *login_name;
    char *record_id;
    size_t record_id_size;
    char *device;
    char *hostname;
    pid_t pid;
    time_t login_time;
    time_t systime;
} sg_user_stats;
```



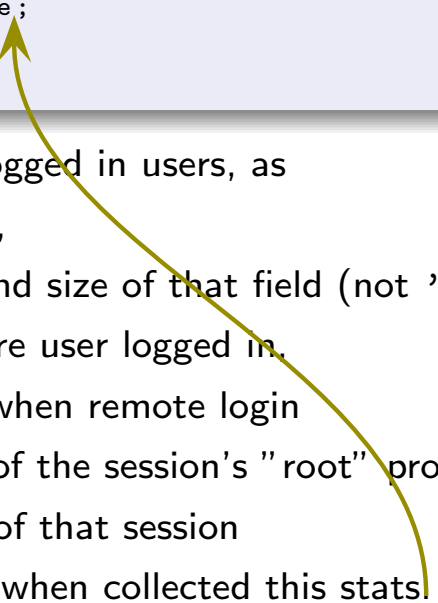
statistics about logged in users, as

- login name,
- record id and size of that field (not '\0' terminated),
- device where user logged in,
- hostname when remote login
- process id of the session's "root" process
- login time of that session

User stats

sg_user_stats

```
typedef struct {
    char *login_name;
    char *record_id;
    size_t record_id_size;
    char *device;
    char *hostname;
    pid_t pid;
    time_t login_time;
    time_t systime;
} sg_user_stats;
```



statistics about logged in users, as

- login name,
- record id and size of that field (not '\0' terminated),
- device where user logged in,
- hostname when remote login
- process id of the session's "root" process
- login time of that session
- timestamp when collected this stats.

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

Process stats

for each existing process

- name of the process image,

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process,

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process,

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to
- process' user id,

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to
- process' user id, group id,

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to
- process' user id, group id, effective user id and

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to
- process' user id, group id, effective user id and effective group id

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to
- process' user id, group id, effective user id and effective group id
- context switches done by the process, also separated by voluntary and involuntary

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to
- process' user id, group id, effective user id and effective group id
- context switches done by the process, also separated by voluntary and involuntary
- virtual memory size of the process, thereof resident

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to
- process' user id, group id, effective user id and effective group id
- context switches done by the process, also separated by voluntary and involuntary
- virtual memory size of the process, thereof resident
- start time of the process, time spent on CPU during lifetime, relative to system usage

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to
- process' user id, group id, effective user id and effective group id
- context switches done by the process, also separated by voluntary and involuntary
- virtual memory size of the process, thereof resident
- start time of the process, time spent on CPU during lifetime, relative to system usage
- nice value of the process (process scheduling increment)

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to
- process' user id, group id, effective user id and effective group id
- context switches done by the process, also separated by voluntary and involuntary
- virtual memory size of the process, thereof resident
- start time of the process, time spent on CPU during lifetime, relative to system usage
- nice value of the process (process scheduling increment)
- device type: one of `SG_PROCESS_STATE_RUNNING`, `SG_PROCESS_STATE_SLEEPING`, `SG_PROCESS_STATE_STOPPED`, `SG_PROCESS_STATE_ZOMBIE` or `SG_PROCESS_STATE_UNKNOWN`

Process stats

sg_process_stats

```
typedef struct {
    char *process_name;
    char *proctitle;
    pid_t pid;
    pid_t parent;
    pid_t pgid;
    pid_t sessid;
    uid_t uid;
    uid_t euid;
    gid_t gid;
    gid_t egid;
    unsigned long long context_switches;
    unsigned long long voluntary_context_switches;
    unsigned long long involuntary_context_switches;
    unsigned long long proc_size;
    unsigned long long proc_resident;
    time_t start_time;
    time_t time_spent;
    double cpu_percent;
    int nice;
    sg_process_state state;
    time_t systime;
} sg_process_stats;
```

for each existing process

- name of the process image,
- title of the process (usually FQPN + args)
- process id of the process, the parent process, the process group leader and the session id of the session the process belongs to
- process' user id, group id, effective user id and effective group id
- context switches done by the process, also separated by voluntary and involuntary
- virtual memory size of the process, thereof resident
- start time of the process, time spent on CPU during lifetime, relative to system usage
- nice value of the process (process scheduling increment)
- device type: one of `SG_PROCESS_STATE_RUNNING`, `SG_PROCESS_STATE_SLEEPING`, `SG_PROCESS_STATE_STOPPED`, `SG_PROCESS_STATE_ZOMBIE` or `SG_PROCESS_STATE_UNKNOWN`
- timestamp when collected this stats

Network I/O stats

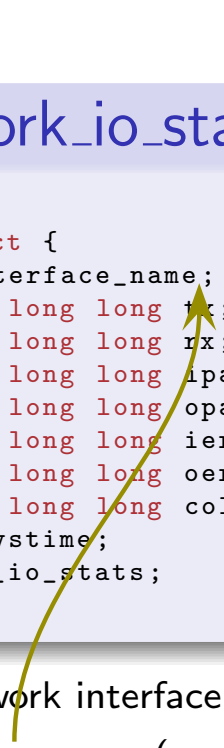
sg_network_io_stats

```
typedef struct {
    char *interface_name;
    unsigned long long tx;
    unsigned long long rx;
    unsigned long long ipackets;
    unsigned long long opackets;
    unsigned long long ierrors;
    unsigned long long oerrors;
    unsigned long long collisions;
    time_t systime;
} sg_network_io_stats;
```

Network I/O stats

sg_network_io_stats

```
typedef struct {
    char *interface_name;
    unsigned long long tx;
    unsigned long long rx;
    unsigned long long ipackets;
    unsigned long long opackets;
    unsigned long long ierrors;
    unsigned long long oerrors;
    unsigned long long collisions;
    time_t systime;
} sg_network_io_stats;
```



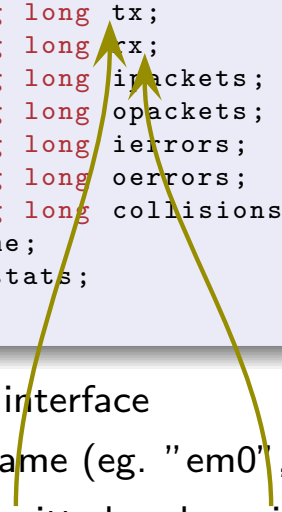
for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),

Network I/O stats

sg_network_io_stats

```
typedef struct {
    char *interface_name;
    unsigned long long tx;
    unsigned long long rx;
    unsigned long long ipackets;
    unsigned long long opackets;
    unsigned long long ierrors;
    unsigned long long oerrors;
    unsigned long long collisions;
    time_t systime;
} sg_network_io_stats;
```



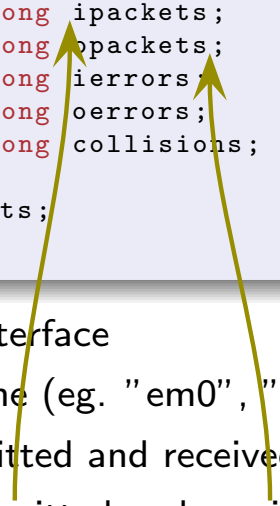
for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),
- bytes transmitted and received

Network I/O stats

sg_network_io_stats

```
typedef struct {
    char *interface_name;
    unsigned long long tx;
    unsigned long long rx;
    unsigned long long ipackets;
    unsigned long long opackets;
    unsigned long long ierrors;
    unsigned long long oerrors;
    unsigned long long collisions;
    time_t systime;
} sg_network_io_stats;
```



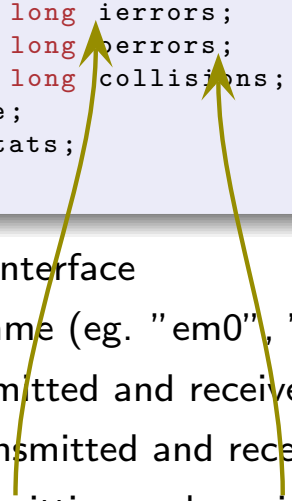
for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),
- bytes transmitted and received
- packets transmitted and received

Network I/O stats

sg_network_io_stats

```
typedef struct {
    char *interface_name;
    unsigned long long tx;
    unsigned long long rx;
    unsigned long long ipackets;
    unsigned long long opackets;
    unsigned long long ierrors;
    unsigned long long oerrors;
    unsigned long long collisions;
    time_t systime;
} sg_network_io_stats;
```



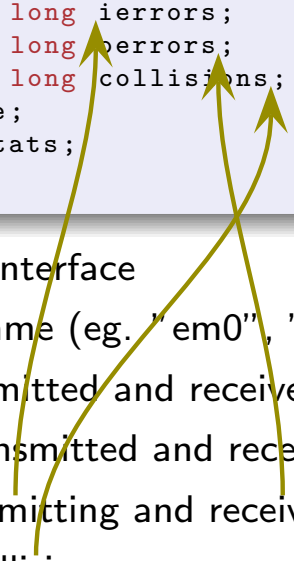
for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),
- bytes transmitted and received
- packets transmitted and received
- errors transmitting and receiving packets

Network I/O stats

sg_network_io_stats

```
typedef struct {
    char *interface_name;
    unsigned long long tx;
    unsigned long long rx;
    unsigned long long ipackets;
    unsigned long long opackets;
    unsigned long long ierrors;
    unsigned long long oerrors;
    unsigned long long collisions;
    time_t systime;
} sg_network_io_stats;
```



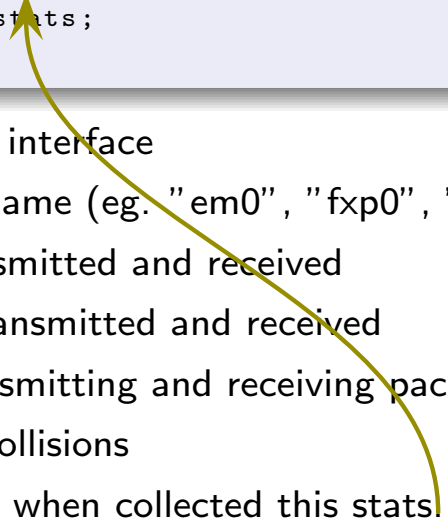
for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),
- bytes transmitted and received
- packets transmitted and received
- errors transmitting and receiving packets
- detected collisions

Network I/O stats

sg_network_io_stats

```
typedef struct {
    char *interface_name;
    unsigned long long tx;
    unsigned long long rx;
    unsigned long long ipackets;
    unsigned long long opackets;
    unsigned long long ierrors;
    unsigned long long oerrors;
    unsigned long long collisions;
    time_t systime;
} sg_network_io_stats;
```



for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),
- bytes transmitted and received
- packets transmitted and received
- errors transmitting and receiving packets
- detected collisions
- timestamp when collected this stats

Network Interface stats

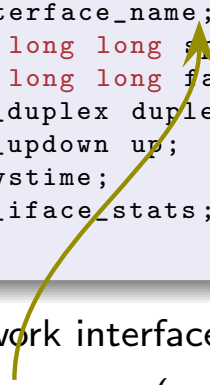
sg_network_iface_stats

```
typedef struct {
    char *interface_name;
    unsigned long long speed;
    unsigned long long factor;
    sg_iface_duplex duplex;
    sg_iface_updown up;
    time_t systime;
} sg_network_iface_stats;
```

Network Interface stats

sg_network_iface_stats

```
typedef struct {
    char *interface_name;
    unsigned long long speed;
    unsigned long long factor;
    sg_iface_duplex duplex;
    sg_iface_updown up;
    time_t systime;
} sg_network_iface_stats;
```



for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),

Network Interface stats

sg_network_iface_stats

```
typedef struct {
    char *interface_name;
    unsigned long long speed;
    unsigned long long factor;
    sg_iface_duplex duplex;
    sg_iface_updown up;
    time_t systime;
} sg_network_iface_stats;
```

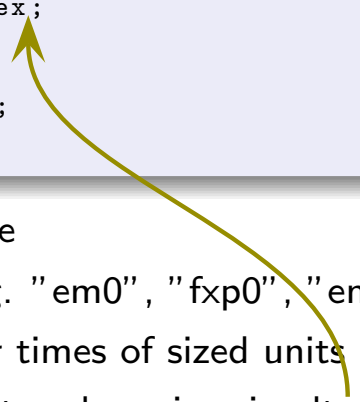
for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),
- capable to transfer times of sized units per second

Network Interface stats

sg_network_iface_stats

```
typedef struct {
    char *interface_name;
    unsigned long long speed;
    unsigned long long factor;
    sg_iface_duplex duplex;
    sg_iface_updown up;
    time_t systime;
} sg_network_iface_stats;
```



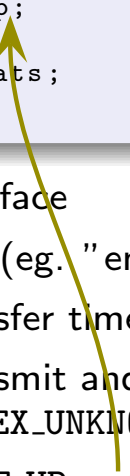
for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),
- capable to transfer times of sized units per second
- capable to transmit and receive simultanously (`SG_IFACE_DUPLEX_FULL`, `SG_IFACE_DUPLEX_HALF` or `SG_IFACE_DUPLEX_UNKNOWN`)

Network Interface stats

sg_network_iface_stats

```
typedef struct {
    char *interface_name;
    unsigned long long speed;
    unsigned long long factor;
    sg_iface_duplex duplex;
    sg_iface_updown up;
    time_t systime;
} sg_network_iface_stats;
```



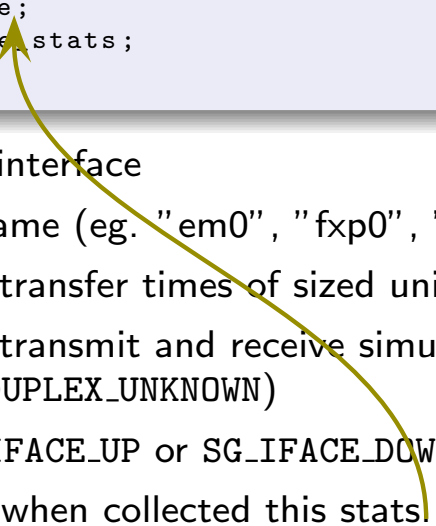
for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),
- capable to transfer times of sized units per second
- capable to transmit and receive simultanously (`SG_IFACE_DUPLEX_FULL`, `SG_IFACE_DUPLEX_HALF` or `SG_IFACE_DUPLEX_UNKNOWN`)
- NIC is `SG_IFACE_UP` or `SG_IFACE_DOWN`

Network Interface stats

sg_network_iface_stats

```
typedef struct {
    char *interface_name;
    unsigned long long speed;
    unsigned long long factor;
    sg_iface_duplex duplex;
    sg_iface_updown up;
    time_t systime;
} sg_network_iface_stats;
```



for each network interface

- interface name (eg. "em0", "fxp0", "en0", "eth0" ...),
- capable to transfer times of sized units per second
- capable to transmit and receive simultanously (`SG_IFACE_DUPLEX_FULL`, `SG_IFACE_DUPLEX_HALF` or `SG_IFACE_DUPLEX_UNKNOWN`)
- NIC is `SG_IFACE_UP` or `SG_IFACE_DOWN`
- timestamp when collected this stats

Error information

sg_error_details

```
typedef struct sg_error_details {
sg_error error;
int errno_value;
const char *error_arg;
} sg_error_details;
```

Error information

sg_error_details

```
typedef struct sg_error_details {
sg_error error;
int errno_value;
const char *error_arg;
} sg_error_details;
```

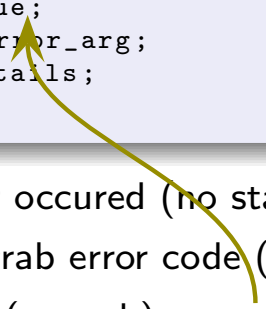
when an error occured (no stats are resulted upon querying):

- libstatgrab error code (eg. `SG_ERROR_INVALID_ARGUMENT`)

Error information

sg_error_details

```
typedef struct sg_error_details {
sg_error error;
int errno_value;
const char *error_arg;
} sg_error_details;
```



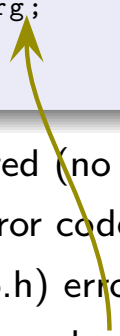
when an error occured (no stats are resulted upon querying):

- libstatgrab error code (eg. `SG_ERROR_INVALID_ARGUMENT`)
- system (errno.h) error code (eg. `EBUSY`

Error information

sg_error_details

```
typedef struct sg_error_details {
sg_error error;
int errno_value;
const char *error_arg;
} sg_error_details;
```



when an error occured (no stats are resulted upon querying):

- libstatgrab error code (eg. `SG_ERROR_INVALID_ARGUMENT`)
- system (errno.h) error code (eg. `EBUSY`
- optional error explanation message (eg. file name, process id, . . .)

Part III

Unix::Statgrab

Entry Functions

```
get_error(); # return details about last error
get_host_info(); # returns sg_host_info
get_cpu_stats(); # returns sg_cpu_stats
get_disk_io_stats(); # returns sg_disk_io_stats
get_fs_stats(); # returns sg_fs_stats
get_load_stats(); # returns sg_load_stats
get_mem_stats(); # returns sg_mem_stats
get_swap_stats(); # returns sg_swap_stats
get_network_io_stats(); # returns sg_network_io_stats
get_network_iface_stats(); # returns sg_network_iface_stats
get_page_stats(); # returns sg_page_stats
get_user_stats(); # returns sg_user_stats
get_process_stats(); # returns sg_process_stats
```

Common . . .

Common . . .

```
use Unix::Statgrab;

my $host_info = get_host_info() or croak( get_error()->strperror() );
printf( "%d\n", $host_info->entries() );
my $cpu_stats = get_cpu_stats() or croak( get_error()->strperror() );
printf( "%d\n", $cpu_stats->entries() );
my $disk_io_stats = get_disk_io_stats() or croak( get_error()->strperror() );
printf( "%d\n", $disk_io_stats->entries() );
my $fs_stats = get_fs_stats() or croak( get_error()->strperror() );
printf( "%d\n", $fs_stats->entries() );
my $load_stats = get_load_stats() or croak( get_error()->strperror() );
printf( "%d\n", $load_stats->entries() );
my $mem_stats = get_mem_stats() or croak( get_error()->strperror() );
printf( "%d\n", $mem_stats->entries() );
my $swap_stats = get_swap_stats() or croak( get_error()->strperror() );
printf( "%d\n", $swap_stats->entries() );
my $net_io_stats = get_network_io_stats() or croak( get_error()->strperror() );
printf( "%d\n", $net_io_stats->entries() );
my $net_iface_stats = get_network_iface_stats() or croak( get_error()->strperror() );
printf( "%d\n", $net_iface_stats->entries() );
my $paging_stats = get_page_stats() or croak( get_error()->strperror() );
printf( "%d\n", $paging->entries() );
my $user_stats = get_user_stats() or croak( get_error()->strperror() );
printf( "%d\n", $user_stats->entries() );
my $proc_stats = get_process_stats() or croak( get_error()->strperror() );
printf( "%d\n", $proc_stats->entries() );
```

SYNOPSIS

SYNOPSIS

```
use Unix::Statgrab;

my $host_stats = get_host_info();
print $host_stats->hostname . " is a " . $host_stats->bitwidth . " " . $host_stats->os_name . "\n";

my $filesystems = get_fs_stats();
my @mount_points = map { $filesystems->mnt_point($_) } (0 .. $filesystems->entries() - 1);
print $host_stats->hostname . " has " . join( ", ", @mount_points ) . " mounted\n";

my $proc_list = get_process_stats();
my @proc_by_type;
foreach my $proc_entry (0 .. $proc_list->entries() - 1) {
    $proc_by_type[$proc_list->state($proc_entry)]++;
}
my $total_procs = 0;
$total_procs += $_ for grep { defined $_ } @proc_by_type;
foreach my $state (qw(SG_PROCESS_STATE_RUNNING SG_PROCESS_STATE_SLEEPING
                      SG_PROCESS_STATE_STOPPED SG_PROCESS_STATE_ZOMBIE
             SG_PROCESS_STATE_UNKNOWN)) {
    defined $proc_by_type[Unix::Statgrab->$state] or next;
    print $proc_by_type[Unix::Statgrab->$state] . " of " . $total_procs . " procs in $state\n";
}
```

SYNOPSIS II

SYNOPSIS diff / percent

```
use Unix::Statgrab;

my $last_cpu_stats = get_cpu_stats() or croak( get_error()->strperror() );
do_sth_way_longer();
my $cpu_diff = get_cpu_stats()->get_cpu_stats_diff($last_cpu_stats);

my $last_cpu_percent = $last_cpu_percent->get_cpu_percents();
my $diff_cpu_percent = $cpu_diff->get_cpu_percents();
my $now_cpu_percent = get_cpu_stats()->get_cpu_percents();

my $last_disk_io_stats = get_disk_io_stats() or croak( get_error()->strperror() );
do_sth_way_longer();
my $disk_io_diff = get_disk_io_stats()->get_disk_io_stats_diff($last_disk_io_stats);

my $last_fs_stats = get_fs_stats() or croak( get_error()->strperror() );
do_sth_way_longer();
my $fs_diff = get_fs_stats()->get_fs_stats_diff($last_fs_stats);

my $last_net_io_stats = get_network_io_stats() or croak( get_error()->strperror() );
do_sth_way_longer();
my $net_io_diff = get_network_io_stats()->get_network_io_stats_diff($last_net_io_stats);

my $last_paging_stats = get_page_stats() or croak( get_error()->strperror() );
do_sth_way_longer();
my $paging_diff = get_page_stats()->get_page_stats_diff($last_paging_stats);
```

Resources

Software

http://www.i-scream.org/libstatgrab/
http://search.cpan.org/dist/Unix-Statgrab/
https://metacpan.org/module/Unix::Statgrab

Resources

Software

http://www.i-scream.org/libstatgrab/
http://search.cpan.org/dist/Unix-Statgrab/
https://metacpan.org/module/Unix::Statgrab

Mailing List

https://lists.i-scream.org/pipermail/users/
users@i-scream.org

Resources

Software

http://www.i-scream.org/libstatgrab/
http://search.cpan.org/dist/Unix-Statgrab/
https://metacpan.org/module/Unix::Statgrab

Mailing List

https://lists.i-scream.org/pipermail/users/
users@i-scream.org

IRC

irc://irc.freenode.net/#libstatgrab

Thank You

Thank you

Thank You

Thank you

- Tim Bishop for caring for high quality release

Thank You

Thank you

- Tim Bishop for caring for high quality release
- H. Merijn Brand for doing additional tests on more exotic platforms

Thank You

Thank you

- Tim Bishop for caring for high quality release
- H. Merijn Brand for doing additional tests on more exotic platforms
- Reini Urban for proving on commodity hardware for being sane

Thank You

Thank you

- Tim Bishop for caring for high quality release
- H. Merijn Brand for doing additional tests on more exotic platforms
- Reini Urban for proving on commodity hardware for being sane

Questions?

Jens Rehsack <rehsack@cpan.org>