Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

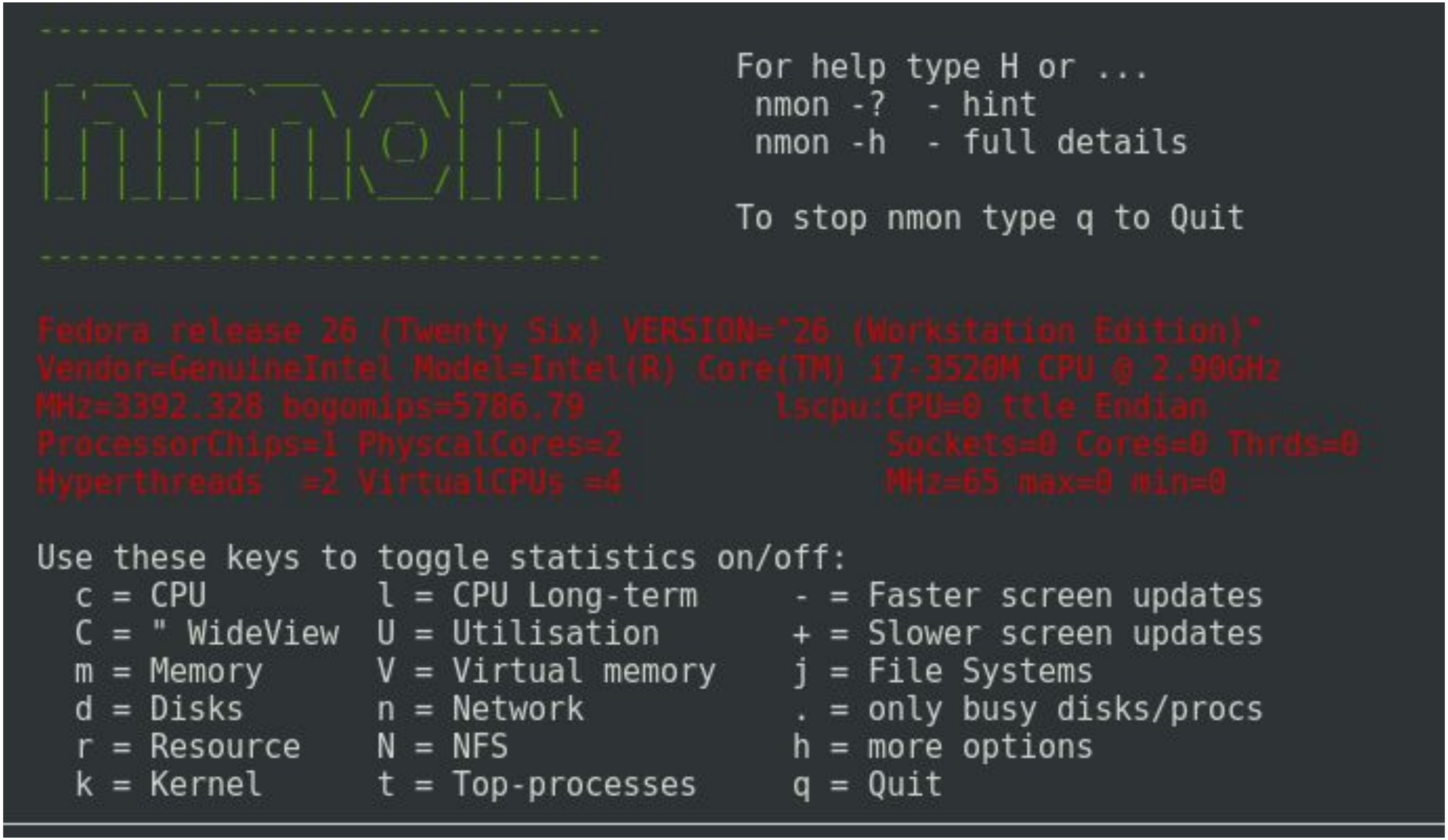
Utility nmon

Студент гр. 13541/3: Медведев М.А.

Преподаватель: Душутина Е.В.

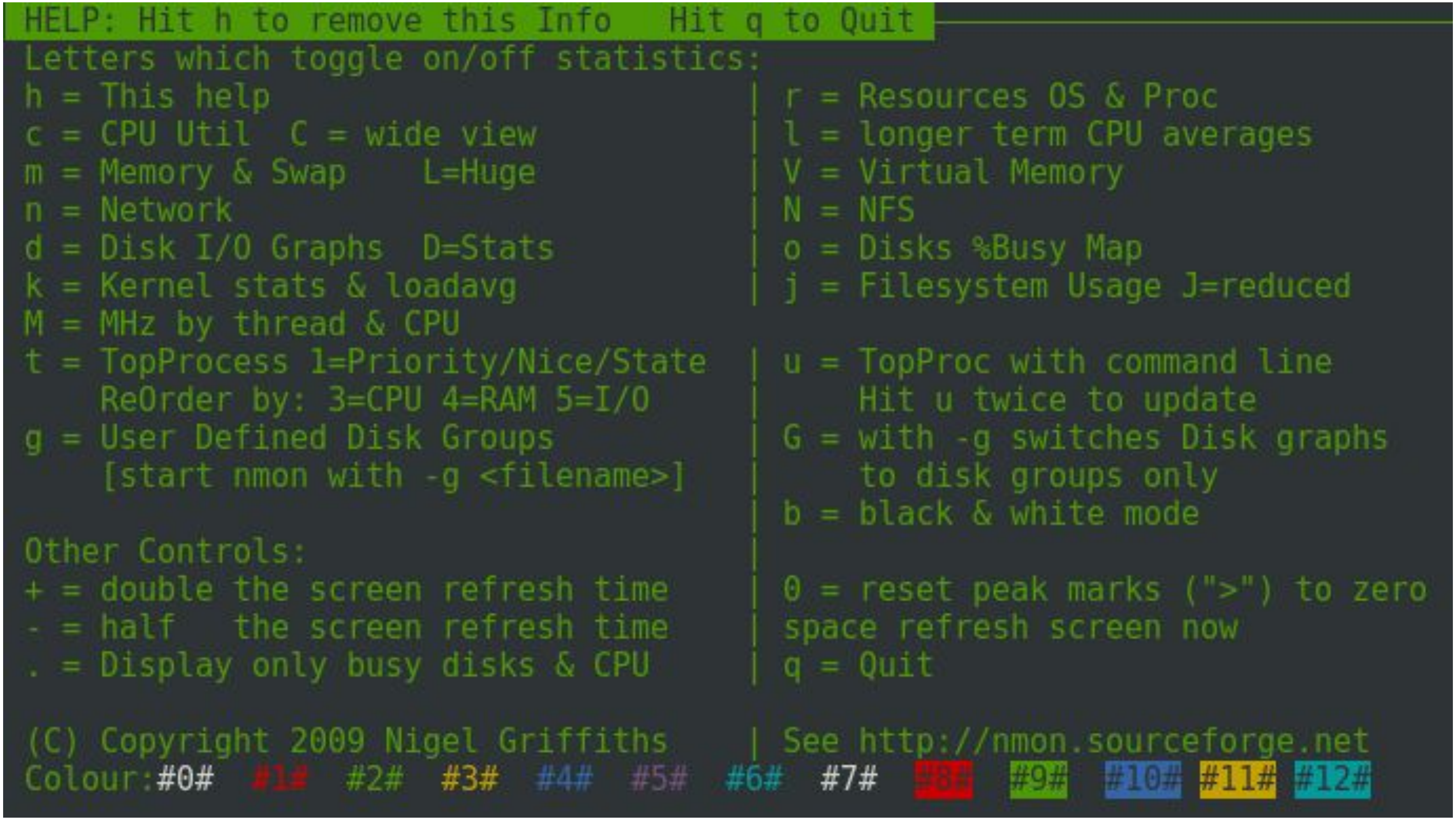
Санкт-Петербург
2018

nmon - системный администратор, тюнер, инструмент сравнения



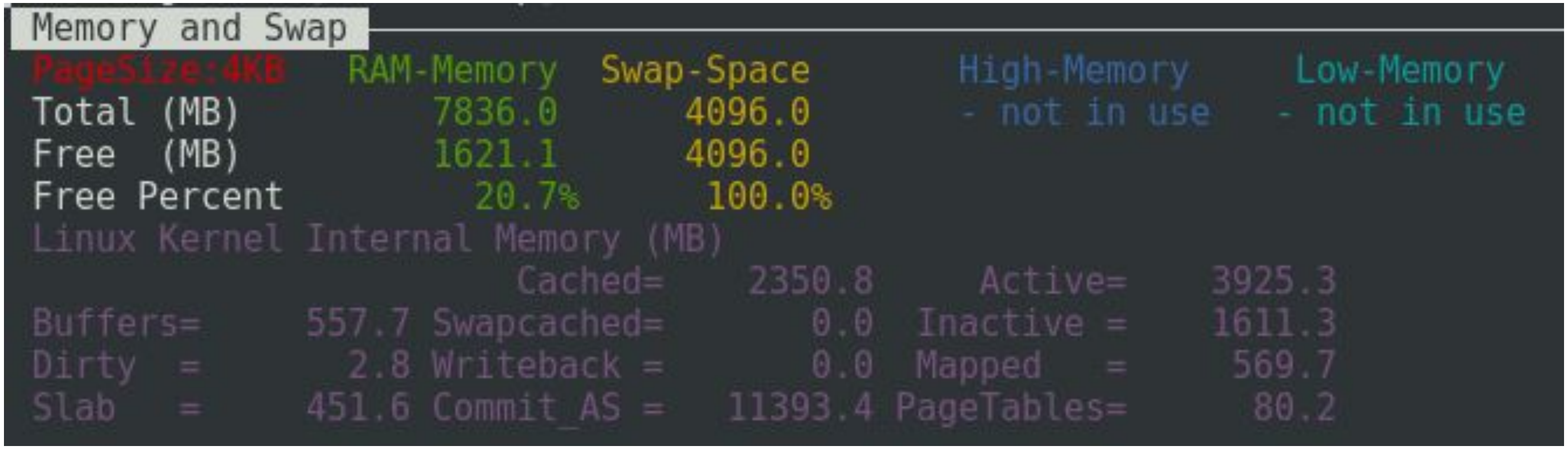
```
------------------------------
                                       For help type H or ...
 _ __  _ __ ___   ___  _ __             nmon -?  - hint
| '_ \| '_ ` _ \ / _ \| '_ \            nmon -h  - full details
| | | | | | | | | (_) | | | |
|_| |_|_| |_| |_|\___/|_| |_|          To stop nmon type q to Quit

------------------------------

Fedora release 26 (Twenty Six) VERSION="26 (Workstation Edition)"
Vendor=GenuineIntel Model=Intel(R) Core(TM) i7-3520M CPU @ 2.90GHz
MHz=3392.328 bogomips=5786.79           lscpu:CPU=0 ttle Endian
ProcessorChips=1 PhyscalCores=2               Sockets=0 Cores=0 Thrds=0
Hyperthreads  =2 VirtualCPUs =4               MHz=65 max=0 min=0

Use these keys to toggle statistics on/off:
  c = CPU           l = CPU Long-term     - = Faster screen updates
  C = " WideView    U = Utilisation       + = Slower screen updates
  m = Memory        V = Virtual memory    j = File Systems
  d = Disks         n = Network           . = only busy disks/procs
  r = Resource      N = NFS               h = more options
  k = Kernel        t = Top-processes     q = Quit
```

Ключи

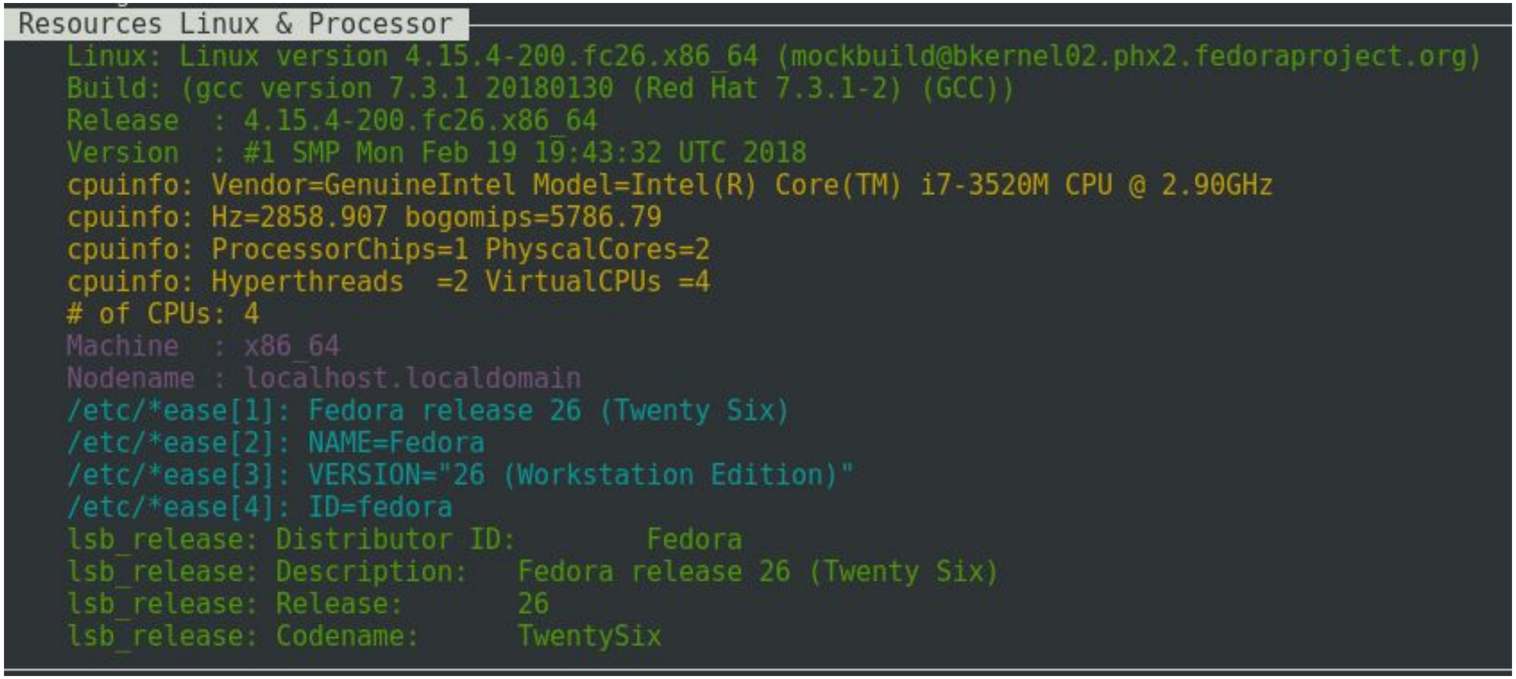


```
HELP: Hit h to remove this Info   Hit q to Quit
Letters which toggle on/off statistics:
h = This help                           | r = Resources OS & Proc
c = CPU Util  C = wide view             | l = longer term CPU averages
m = Memory & Swap    L=Huge             | V = Virtual Memory
n = Network                             | N = NFS
d = Disk I/O Graphs  D=Stats            | o = Disks %Busy Map
k = Kernel stats & loadavg              | j = Filesystem Usage J=reduced
M = MHz by thread & CPU
t = TopProcess 1=Priority/Nice/State    | u = TopProc with command line
    ReOrder by: 3=CPU 4=RAM 5=I/O       |     Hit u twice to update
g = User Defined Disk Groups            | G = with -g switches Disk graphs
    [start nmon with -g <filename>]     |     to disk groups only
                                        | b = black & white mode
Other Controls:                         |
+ = double the screen refresh time      | 0 = reset peak marks (">") to zero
- = half   the screen refresh time      | space refresh screen now
. = Display only busy disks & CPU       | q = Quit

(C) Copyright 2009 Nigel Griffiths      | See http://nmon.sourceforge.net
Colour:#0# #1# #2# #3# #4# #5# #6# #7# #8# #9# #10# #11# #12#
```

Отображение информации с ключами -m & -r

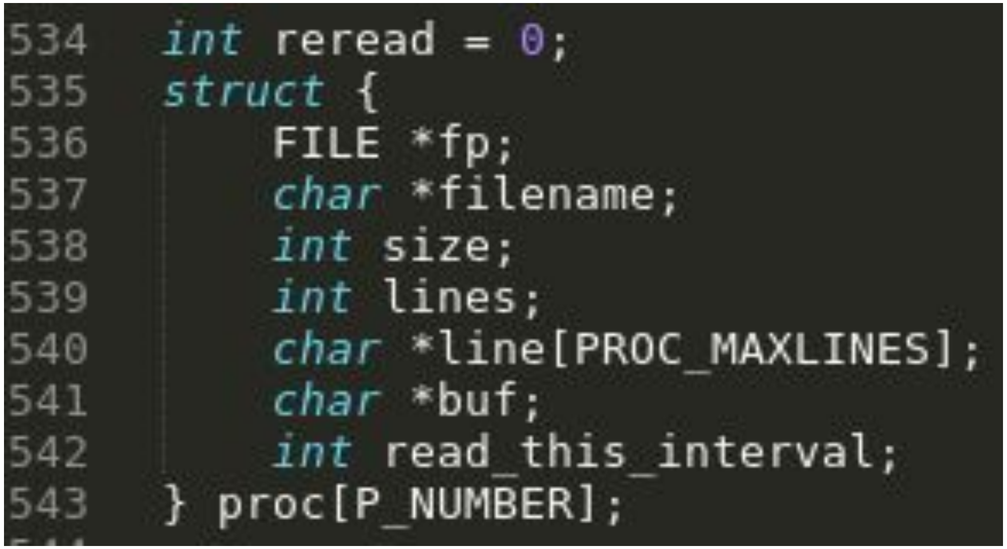


```
 Memory and Swap 
 PageSize:4KB   RAM-Memory  Swap-Space       High-Memory     Low-Memory
 Total (MB)         7836.0     4096.0        - not in use   - not in use
 Free  (MB)         1621.1     4096.0
 Free Percent         20.7%     100.0%
 Linux Kernel Internal Memory (MB)
                        Cached=    2350.8     Active=    3925.3
 Buffers=     557.7 Swapcached=       0.0  Inactive =    1611.3
 Dirty  =       2.8 Writeback =       0.0  Mapped   =     569.7
 Slab   =     451.6 Commit_AS =   11393.4 PageTables=      80.2
```



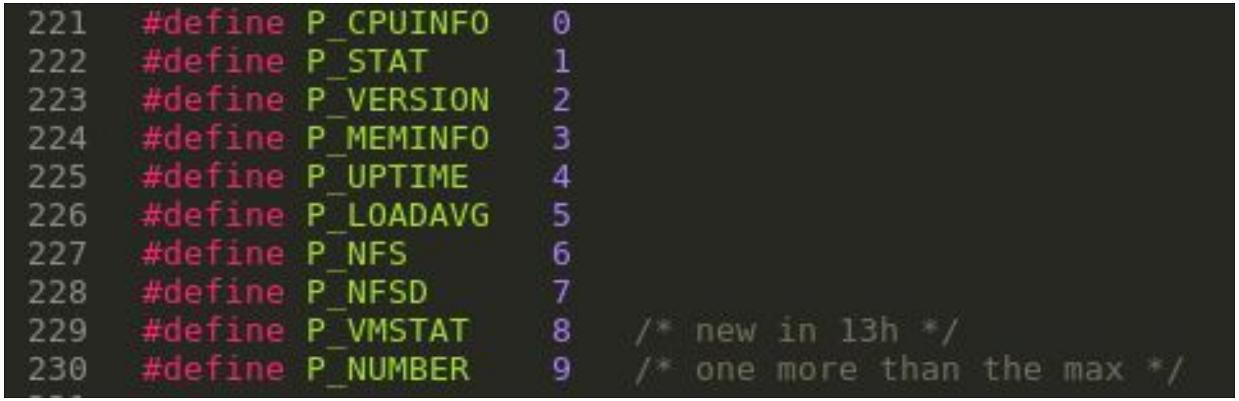
```
 Resources Linux & Processor 
    Linux: Linux version 4.15.4-200.fc26.x86_64 (mockbuild@bkernel02.phx2.fedoraproject.org)
    Build: (gcc version 7.3.1 20180130 (Red Hat 7.3.1-2) (GCC))
    Release  : 4.15.4-200.fc26.x86_64
    Version  : #1 SMP Mon Feb 19 19:43:32 UTC 2018
    cpuinfo: Vendor=GenuineIntel Model=Intel(R) Core(TM) i7-3520M CPU @ 2.90GHz
    cpuinfo: Hz=2858.907 bogomips=5786.79
    cpuinfo: ProcessorChips=1 PhyscalCores=2
    cpuinfo: Hyperthreads  =2 VirtualCPUs =4
    # of CPUs: 4
    Machine  : x86_64
    Nodename : localhost.localdomain
    /etc/*ease[1]: Fedora release 26 (Twenty Six)
    /etc/*ease[2]: NAME=Fedora
    /etc/*ease[3]: VERSION="26 (Workstation Edition)"
    /etc/*ease[4]: ID=fedora
    lsb_release: Distributor ID:        Fedora
    lsb_release: Description:   Fedora release 26 (Twenty Six)
    lsb_release: Release:       26
    lsb_release: Codename:      TwentySix
```

Структура proc

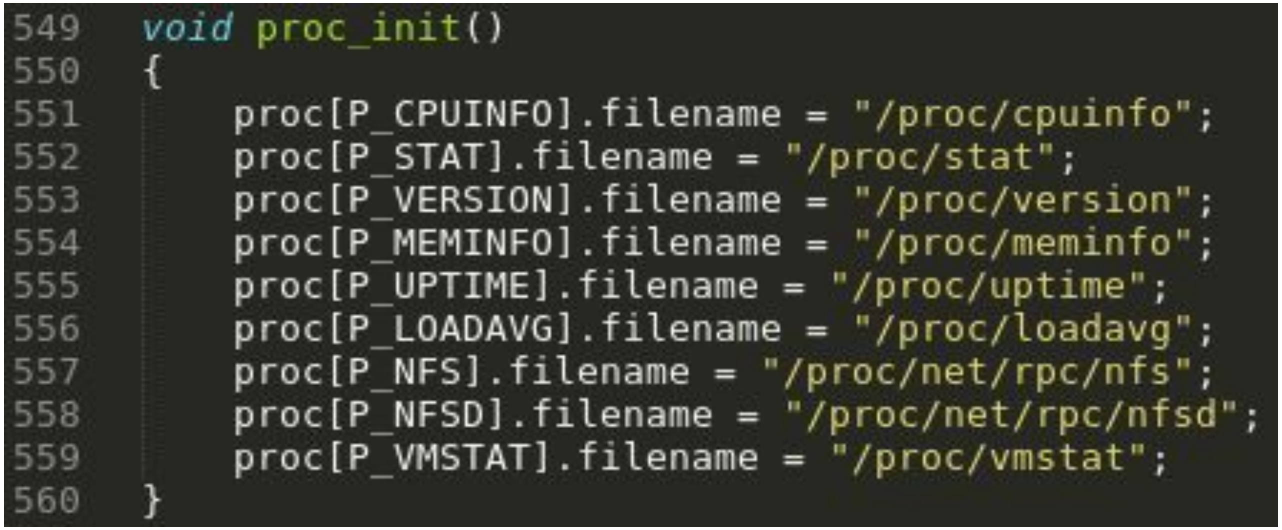


```
int reread = 0;
struct {
    FILE *fp;
    char *filename;
    int size;
    int lines;
    char *line[PROC_MAXLINES];
    char *buf;
    int read_this_interval;
} proc[P_NUMBER];
```

Откуда получаем данные ?

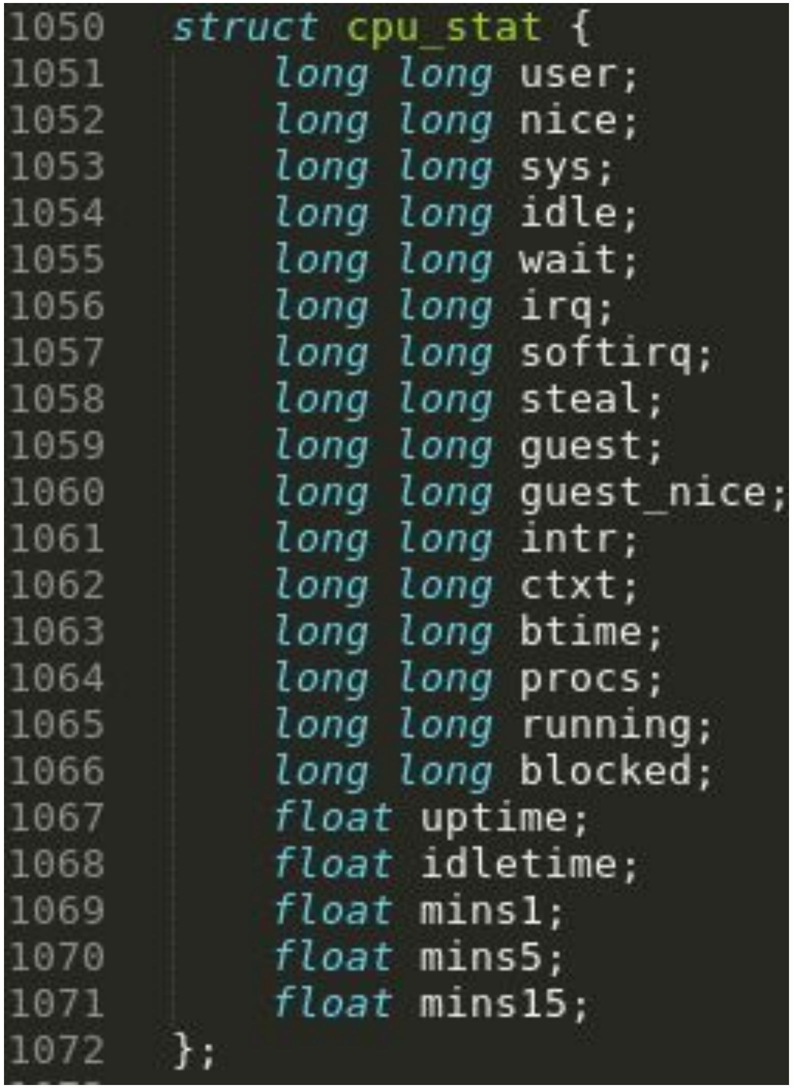


```
#define P_CPUINFO   0
#define P_STAT      1
#define P_VERSION   2
#define P_MEMINFO   3
#define P_UPTIME    4
#define P_LOADAVG   5
#define P_NFS       6
#define P_NFSD      7
#define P_VMSTAT    8   /* new in 13h */
#define P_NUMBER    9   /* one more than the max */
```

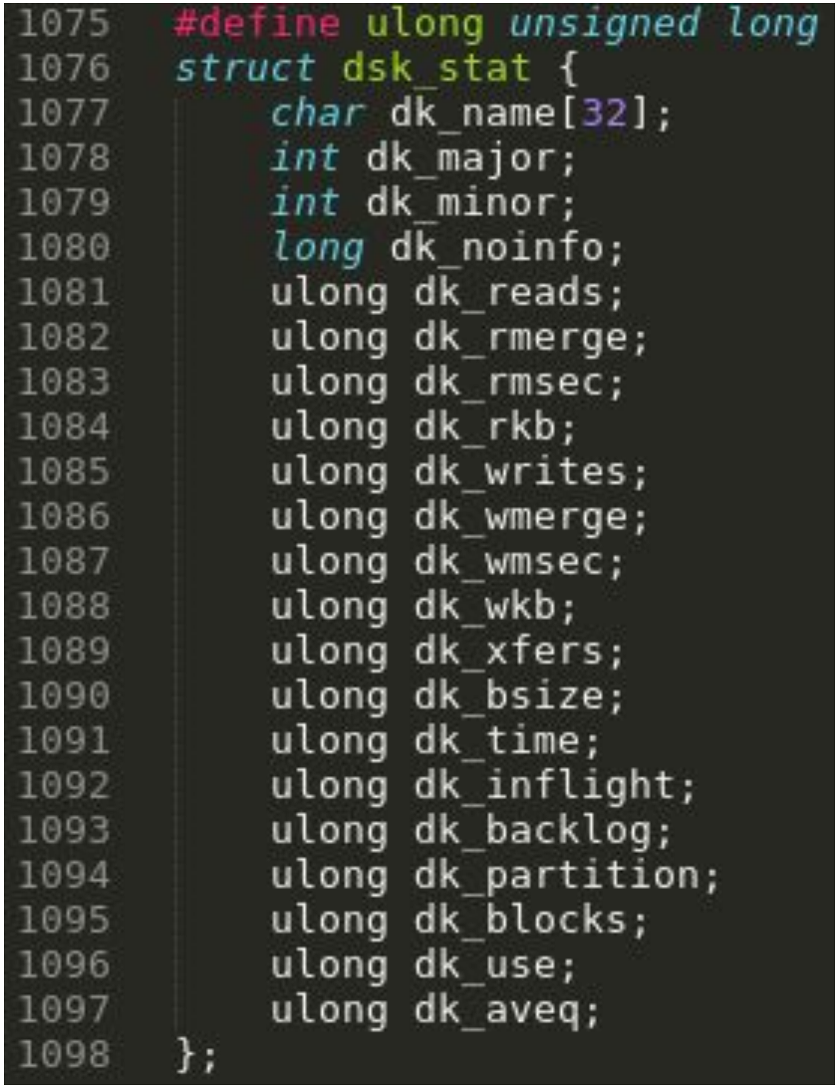


```
void proc_init()
{
    proc[P_CPUINFO].filename = "/proc/cpuinfo";
    proc[P_STAT].filename = "/proc/stat";
    proc[P_VERSION].filename = "/proc/version";
    proc[P_MEMINFO].filename = "/proc/meminfo";
    proc[P_UPTIME].filename = "/proc/uptime";
    proc[P_LOADAVG].filename = "/proc/loadavg";
    proc[P_NFS].filename = "/proc/net/rpc/nfs";
    proc[P_NFSD].filename = "/proc/net/rpc/nfsd";
    proc[P_VMSTAT].filename = "/proc/vmstat";
}
```

Некоторые основные структуры данных для хранения статистики

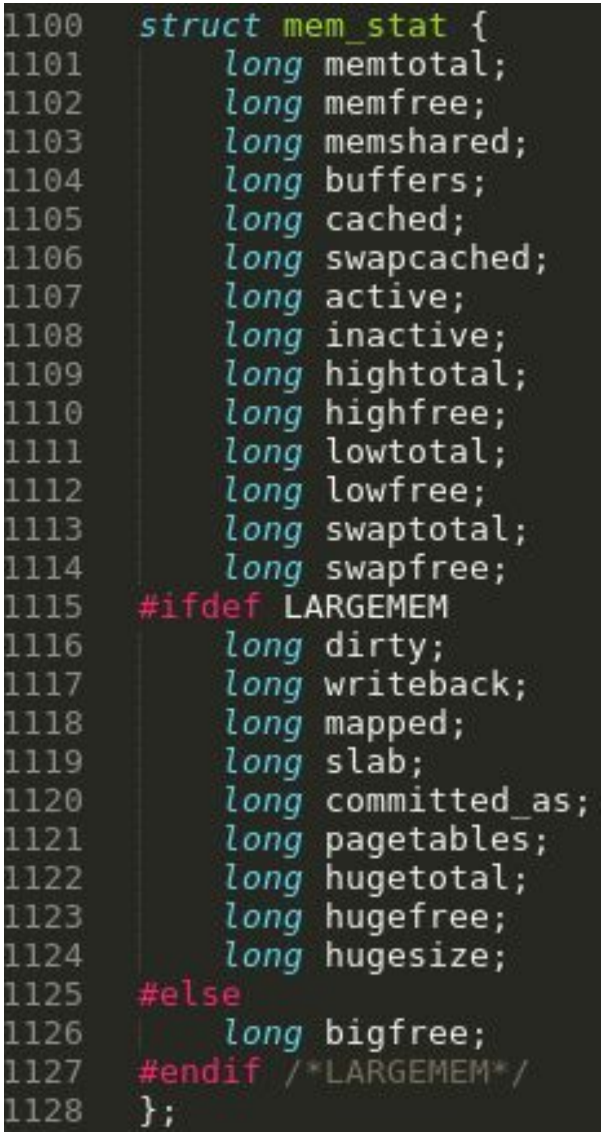


```c
struct cpu_stat {
        long long user;
        long long nice;
        long long sys;
        long long idle;
        long long wait;
        long long irq;
        long long softirq;
        long long steal;
        long long guest;
        long long guest_nice;
        long long intr;
        long long ctxt;
        long long btime;
        long long procs;
        long long running;
        long long blocked;
        float uptime;
        float idletime;
        float mins1;
        float mins5;
        float mins15;
};
```

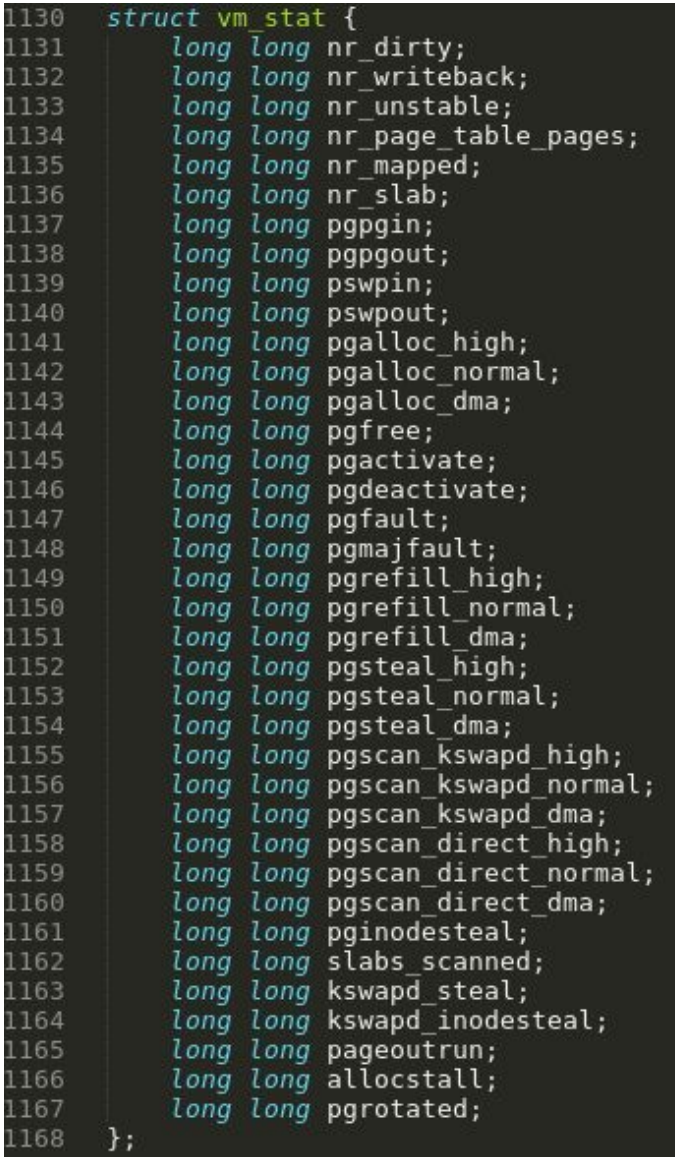


```c
#define ulong unsigned long
struct dsk_stat {
        char dk_name[32];
        int dk_major;
        int dk_minor;
        long dk_noinfo;
        ulong dk_reads;
        ulong dk_rmerge;
        ulong dk_rmsec;
        ulong dk_rkb;
        ulong dk_writes;
        ulong dk_wmerge;
        ulong dk_wmsec;
        ulong dk_wkb;
        ulong dk_xfers;
        ulong dk_bsize;
        ulong dk_time;
        ulong dk_inflight;
        ulong dk_backlog;
        ulong dk_partition;
        ulong dk_blocks;
        ulong dk_use;
        ulong dk_aveq;
};
```

Некоторые основные структуры данных для хранения статистики

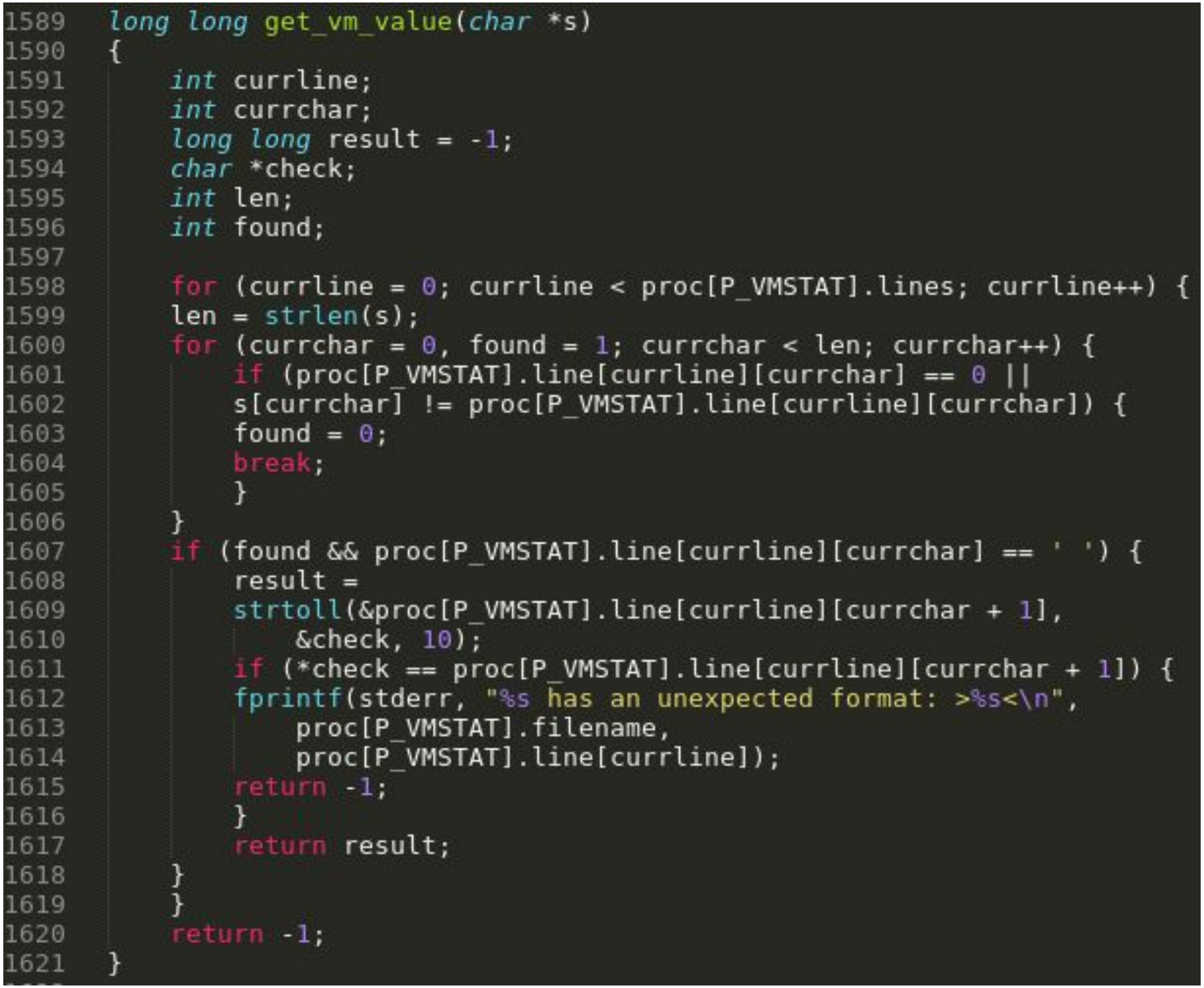


```
struct mem_stat {
    long memtotal;
    long memfree;
    long memshared;
    long buffers;
    long cached;
    long swapcached;
    long active;
    long inactive;
    long hightotal;
    long highfree;
    long lowtotal;
    long lowfree;
    long swaptotal;
    long swapfree;
#ifdef LARGEMEM
    long dirty;
    long writeback;
    long mapped;
    long slab;
    long committed_as;
    long pagetables;
    long hugetotal;
    long hugefree;
    long hugesize;
#else
    long bigfree;
#endif /*LARGEMEM*/
};
```



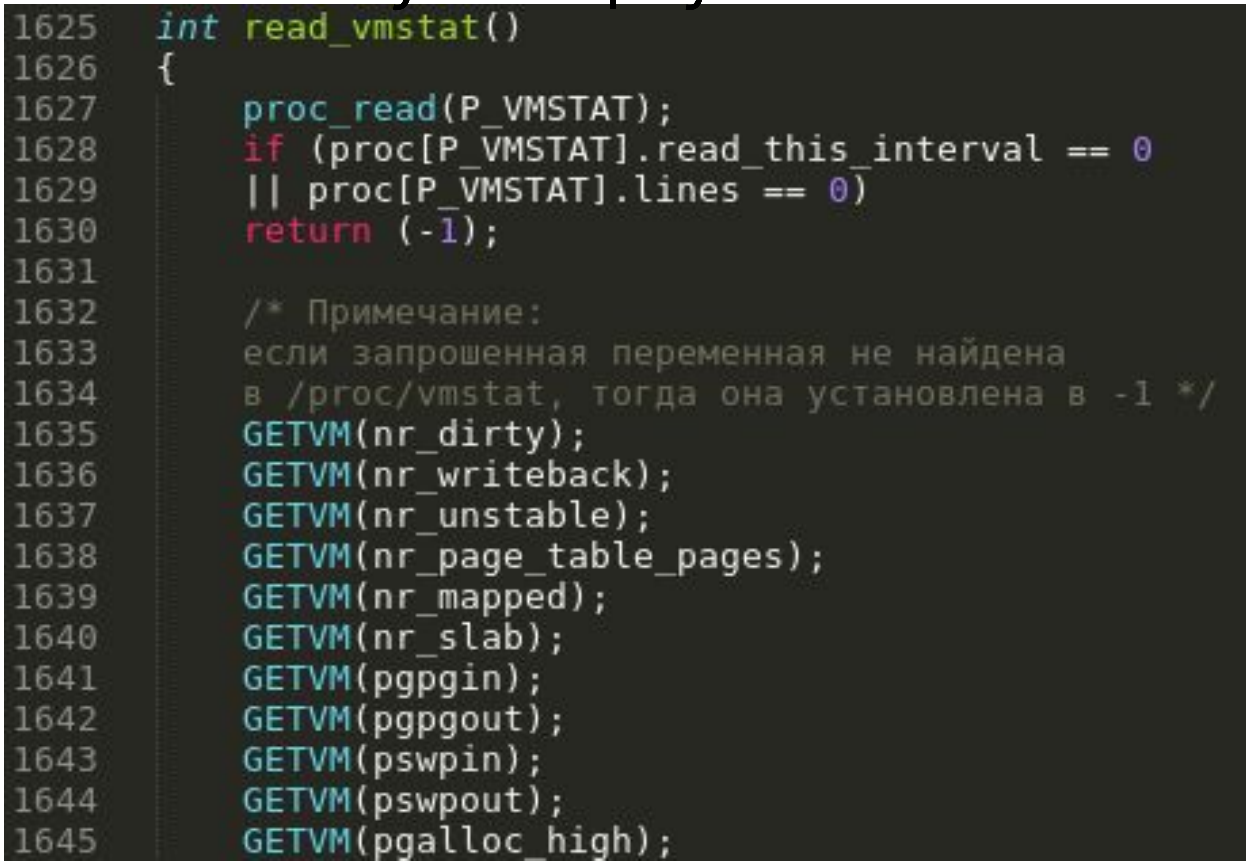
```
struct vm_stat {
    long long nr_dirty;
    long long nr_writeback;
    long long nr_unstable;
    long long nr_page_table_pages;
    long long nr_mapped;
    long long nr_slab;
    long long pgpgin;
    long long pgpgout;
    long long pswpin;
    long long pswpout;
    long long pgalloc_high;
    long long pgalloc_normal;
    long long pgalloc_dma;
    long long pgfree;
    long long pgactivate;
    long long pgdeactivate;
    long long pgfault;
    long long pgmajfault;
    long long pgrefill_high;
    long long pgrefill_normal;
    long long pgrefill_dma;
    long long pgsteal_high;
    long long pgsteal_normal;
    long long pgsteal_dma;
    long long pgscan_kswapd_high;
    long long pgscan_kswapd_normal;
    long long pgscan_kswapd_dma;
    long long pgscan_direct_high;
    long long pgscan_direct_normal;
    long long pgscan_direct_dma;
    long long pginodesteal;
    long long slabs_scanned;
    long long kswapd_steal;
    long long kswapd_inodesteal;
    long long pageoutrun;
    long long allocstall;
    long long pgrotated;
};
```

Собираем статистику о виртуальной памяти

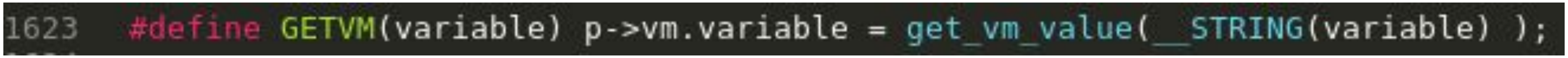


```
long long get_vm_value(char *s)
{
    int currline;
    int currchar;
    long long result = -1;
    char *check;
    int len;
    int found;

    for (currline = 0; currline < proc[P_VMSTAT].lines; currline++) {
    len = strlen(s);
    for (currchar = 0, found = 1; currchar < len; currchar++) {
        if (proc[P_VMSTAT].line[currline][currchar] == 0 ||
        s[currchar] != proc[P_VMSTAT].line[currline][currchar]) {
        found = 0;
        break;
        }
    }
    if (found && proc[P_VMSTAT].line[currline][currchar] == ' ') {
        result =
        strtoll(&proc[P_VMSTAT].line[currline][currchar + 1],
            &check, 10);
        if (*check == proc[P_VMSTAT].line[currline][currchar + 1]) {
        fprintf(stderr, "%s has an unexpected format: >%s<\n",
            proc[P_VMSTAT].filename,
            proc[P_VMSTAT].line[currline]);
        return -1;
        }
        return result;
    }
    }
    return -1;
}
```

Функция read_vmstat() - записывает в vm_stat статистику о виртуальной памяти



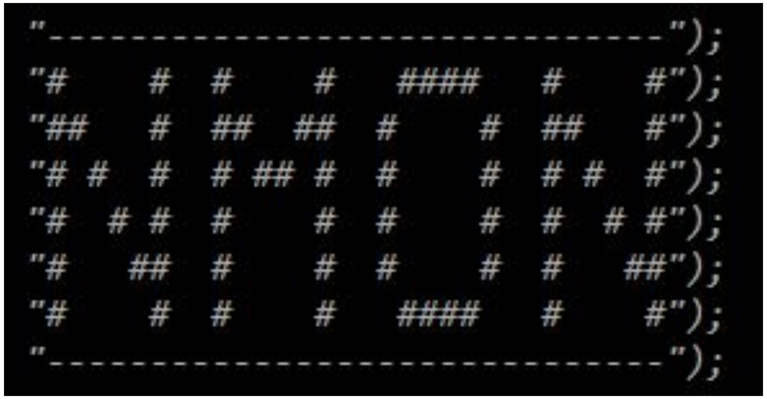
```
int read_vmstat()
{
    proc_read(P_VMSTAT);
    if (proc[P_VMSTAT].read_this_interval == 0
    || proc[P_VMSTAT].lines == 0)
    return (-1);

    /* Примечание:
    если запрошенная переменная не найдена
    в /proc/vmstat, тогда она установлена в -1 */
    GETVM(nr_dirty);
    GETVM(nr_writeback);
    GETVM(nr_unstable);
    GETVM(nr_page_table_pages);
    GETVM(nr_mapped);
    GETVM(nr_slab);
    GETVM(pgpgin);
    GETVM(pgpgout);
    GETVM(pswpin);
    GETVM(pswpout);
    GETVM(pgalloc_high);
```



```
#define GETVM(variable) p->vm.variable = get_vm_value(__STRING(variable) );
```

int main(int argc, char **argv)

- занимает более 4000 строк кода
- “можно разделить на три секции”:
    - получение данных
    - вывод данных
    - обновление данных
- меньше всего содержит комментариев



```
"------------------------------");
"#    #  #    #   ####   #    #");
"##   #  ##  ##  #    #  ##   #");
"# #  #  # ## #  #    #  # #  #");
"#  # #  #    #  #    #  #  # #");
"#   ##  #    #  #    #  #   ##");
"#    #  #    #   ####   #    #");
"------------------------------");
```