

ΑΝΩΤΑΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ
ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ

ΑΕΙ Πειραιά Τ.Τ.

Τμήμα Μηχανικών Ηλεκτρονικών

Υπολογιστικών Συστημάτων

Εργαστήριο Λειτουργικών Συστημάτων

**Εργαστηριακή άσκηση
Πληροφορίες συστήματος**

Επιμέλεια:
Στυλιανός Βουτσινάς
BSc, MSc, PhD(Cand.)

Υπεύθυνος Εργαστηρίου:

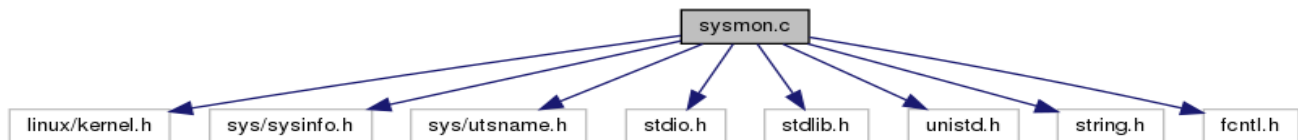
Δρ. Διλιντάς Γεωργιος (Καθηγητής)

Τεκμηρίωση Αρχείων

Τεκμηρίωση Αρχείου sysmon.c

```
#include <linux/kernel.h>
#include <sys/sysinfo.h>
#include <sys/utsname.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <fcntl.h>
```

Διάγραμμα εξάρτησης αρχείου συμπερίληψης για το sysmon.c:



Τεκμηρίωση Συναρτήσεων

int main (void)

Κύρια συνάρτηση του προγράμματος, τυπώνει πληροφορίες από τις δομές sysinfo, utsname και από το αρχείο /proc/cpuinfo.

Επιστρέφει:

το exit code της διεργασίας

Ορισμός στη γραμμή [30](#) του αρχείου [sysmon.c](#).

```
31 {
32  /* Conversion constants. */
33  const long minute = 60;
34  const long hour = minute * 60;
35  const long day = hour * 24;
36  const double megabyte = 1024 * 1024;
37
38  /*
39   struct sysinfo {
40       long uptime;
41       unsigned long loads[3]; // 1, 5, and 15 minute load averages
42       unsigned long totalram; // Total usable main memory size
43       unsigned long freeram; // Available memory size
44       unsigned long sharedram; // Amount of shared memory
45       unsigned long bufferram; // Memory used by buffers
46       unsigned long totalswap; // Total swap space size
47       unsigned long freeswap; // Swap space still available
48       unsigned short procs; // Number of current processes
49       char _f[22]; // Pads structure to 64 bytes
50   };
51  */
52  struct sysinfo si;
53  /*
54   struct utsname {
55       char sysname[]; // Operating system name
56       char nodename[]; // Name within "some implementation-
defined network"
57       char release[]; // Operating system release
58       char version[]; // Operating system version
59       char machine[]; // Hardware identifier
60   #ifdef _GNU_SOURCE
61       char domainname[]; // NIS or YP domain name
62   #endif
63   };
64  */
65  }
```

```

64     */
65     struct utsname sys;
66     sysinfo (&si); // επιστρέφει πληροφορίες για την μνήμη και το φόρτο του
υπολογιστή στην δομή που του έχει δοθεί σαν όρισμα
67     float load = 1<<SI_LOAD_SHIFT;
68     uname(&sys); // επιστρέφει πληροφορίες για το σύστημα στην δομή που του
έχει δοθεί σαν όρισμα
69     printf("*****System statistics*****\n");
70     printf( "%-25s %s\n", "OS type:", sys.sysname );
71     printf( "%-25s %s\n", "System version", sys.version );
72     printf( "%-25s %s\n", "Hostname", sys.nodename );
73     printf( "%-25s %s\n", "Kernel version", sys.release );
74     printf( "%-25s %s\n", "Default shell", getenv("SHELL") );
75     printf( "%-25s %s\n", "User", getenv("USER") );
76
77
78     printf("*****Memory statistics*****\n");
79     printf( "%-25s %5.1f MB\n", "total RAM", si.totalram / megabyte );
80     printf( "%-25s %5.1f MB\n", "free RAM", si.freeram / megabyte );
81     printf( "%-25s %5.1f%\n", "% of free RAM", (((si.freeram / megabyte) /
(si.totalram / megabyte))*100);
82     printf( "%-25s %ld days, %ld:%02ld:%02ld\n", "system uptime",
si.uptime / day, (si.uptime % day) / hour, (si.uptime % hour) / minute, si.uptime
% minute );
83     printf( "%-25s %d\n", "Running Procs", si.procs );
84
85     printf("*****CPU statistics*****\n");
86     printf( "%-25s %s\n", "System Architecture", sys.machine );
87     printf( "%-25s %5.1ld\n", "Number of cores",
sysconf( _SC_NPROCESSORS_ONLN ));
88     printf( "%-25s %5.1f\n", "1min load avg", si.loads[0] / load);
89     printf( "%-25s %5.1f\n", "5min load avg", si.loads[1] / load);
90     printf( "%-25s %5.1f\n", "15min load avg", si.loads[2] / load);
91
92     int fd = open("/proc/cpuinfo", O_RDONLY);
93     char c;
94     char buff[128];
95     int i=0;
96     while(i<128){
97         read(fd,&c,sizeof(char));
98         buff[i]=c;
99         i++;
100     }
101     char* attrs[128];
102     char temp[1024]="";
103     int cpupos=0;
104     int k = parse_string(buff,": \t\n\r\n",attrs);
105
106     for (i=0; i<k; i++){
107         if(strcmp(attrs[i],"vendor_id")==0){
108             printf( "%-25s %s\n", "vendor_id", attrs[i+1]);
109         }
110         if((strcmp(attrs[i],"model")==0)&&(strcmp(attrs[i+1],"name")==0)){
111             cpupos=i+2;
112             while(strcmp(attrs[cpupos],"stepping")!=0){
113                 strcat(temp,attrs[cpupos]);
114                 cpupos++;
115             }
116             printf( "%-25s %s\n", "model name", temp);
117         }
118     }
119     close(fd);
120     return 0;
121 }

```

Το διάγραμμα δείχνει ποιες συναρτήσεις καλούνται από αυτή:



int parse_string (char * *inputString*, char * *Delimiter*, char ** *tokens*)

Μετατρέπει ένα αλφαριθμητικό σε πίνακα αλφαριθμητικών, χωρισμένο με βάση τα διαχωριστικά που έχουν οριστεί

Παράμετροι:

in	<i>inputString</i>	το αρχικό αλφαριθμητικό
in	<i>Delimiter</i>	τα διαχωριστικά
in	<i>tokens</i>	ο τελικός πίνακας το μήκος του πίνακα tokens

Επιστρέφει:

το μήκος του πίνακα tokens

Ορισμός στη γραμμή [16](#) του αρχείου [sysmon.c](#).

```
17 {  
18     int i=0;  
19     tokens[i] = strtok(inputString, Delimiter);  
20     i++;  
21     while ((tokens[i] = strtok(NULL, Delimiter)) != NULL){  
22         i++;  
23     }  
24     return i;  
25 }
```

Το διάγραμμα δείχνει από ποιες συναρτήσεις καλείται αυτή η συνάρτηση:



To αρχείο sysmon.c

```
1 #include <linux/kernel.h>
2 #include <sys/sysinfo.h>
3 #include <sys/utsname.h>
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <unistd.h>
7 #include <string.h>
8 #include <fcntl.h>
16 int parse_string(char *inputString, char *Delimiter, char **tokens)
17 {
18     int i=0;
19     tokens[i] = strtok(inputString, Delimiter);
20     i++;
21     while ((tokens[i] = strtok(NULL, Delimiter)) != NULL){
22         i++;
23     }
24     return i;
25 }
26
30 int main (void)
31 {
32     /* Conversion constants. */
33     const long minute = 60;
34     const long hour = minute * 60;
35     const long day = hour * 24;
36     const double megabyte = 1024 * 1024;
37
38     /*
39     struct sysinfo {
40         long uptime;
41         unsigned long loads[3]; // 1, 5, and 15 minute load averages
42         unsigned long totalram; // Total usable main memory size
43         unsigned long freeram; // Available memory size
44         unsigned long sharedram; // Amount of shared memory
45         unsigned long bufferram; // Memory used by buffers
46         unsigned long totalswap; // Total swap space size
47         unsigned long freeswap; // Swap space still available
48         unsigned short procs; // Number of current processes
49         char _f[22]; // Pads structure to 64 bytes
50     };
51     */
52     struct sysinfo si;
53     /*
54     struct utsname {
55         char sysname[]; // Operating system name
56         char nodename[]; // Name within "some implementation-defined
network"
57         char release[]; // Operating system release
58         char version[]; // Operating system version
59         char machine[]; // Hardware identifier
60         #ifdef _GNU_SOURCE
61         char domainname[]; // NIS or YP domain name
62         #endif
63     };
64     */
65     struct utsname sys;
66     sysinfo (&si); // επιστρέφει πληροφορίες για την μνήμη και το φόρτο του
υπολογιστή στην δομή που του έχει δώσει σαν όρισμα
67     float load = 1<<SI_LOAD_SHIFT;
68     uname(&sys); // επιστρέφει πληροφορίες για το σύστημα στην δομή που του έχει
δωθεί σαν όρισμα
69     printf("*****System statistics*****\n");
70     printf( "%-25s %s\n", "OS type:", sys.sysname );
71     printf( "%-25s %s\n", "System version", sys.version );
72     printf( "%-25s %s\n", "Hostname", sys.nodename );
73     printf( "%-25s %s\n", "Kernel version", sys.release );
74     printf( "%-25s %s\n", "Default shell", getenv("SHELL") );
75     printf( "%-25s %s\n", "User", getenv("USER") );
76
77
78     printf("*****Memory statistics*****\n");
```

```

79     printf( "%-25s %5.1f MB\n", "total RAM", si.totalram / megabyte );
80     printf( "%-25s %5.1f MB\n", "free RAM", si.freeram / megabyte );
81     printf( "%-25s %5.1f%\n", "% of free RAM", ((si.freeram / megabyte) /
(si.totalram / megabyte))*100);
82     printf( "%-25s %ld days, %ld:%02ld:%02ld\n", "system uptime", si.uptime /
day, (si.uptime % day) / hour, (si.uptime % hour) / minute, si.uptime % minute );
83     printf( "%-25s %d\n", "Running Procs", si.procs );
84
85     printf("*****CPU statistics*****\n");
86     printf( "%-25s %s\n", "System Architecture", sys.machine );
87     printf( "%-25s %5.1ld\n", "Number of cores",
sysconf( _SC_NPROCESSORS_ONLN ));
88     printf( "%-25s %5.1f\n", "1min load avg", si.loads[0] / load);
89     printf( "%-25s %5.1f\n", "5min load avg", si.loads[1] / load);
90     printf( "%-25s %5.1f\n", "15min load avg", si.loads[2] / load);
91
92     int fd = open("/proc/cpuinfo", O_RDONLY);
93     char c;
94     char buff[128];
95     int i=0;
96     while(i<128){
97         read(fd,&c,sizeof(char));
98         buff[i]=c;
99         i++;
100    }
101    char* attrs[128];
102    char temp[1024]="";
103    int cpupos=0;
104    int k = parse\_string(buff,": \t\n\r\n",attrs);
105
106    for (i=0; i<k; i++){
107        if(strcmp(attrs[i],"vendor_id")==0){
108            printf( "%-25s %s\n", "vendor_id", attrs[i+1]);
109        }
110        if((strcmp(attrs[i],"model")==0)&&(strcmp(attrs[i+1],"name")==0)){
111            cpupos=i+2;
112            while(strcmp(attrs[cpupos],"stepping")!=0){
113                strcat(temp,attrs[cpupos]);
114                cpupos++;
115            }
116            printf( "%-25s %s\n", "model name", temp);
117        }
118    }
119    close(fd);
120    return 0;
121 }

```
