

**Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации
Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский технический университет связи и информатики»**

Кафедра Системного программирования

ЛАБОРАТОРНАЯ РАБОТА №1
по дисциплине
ОПЕРАЦИОННЫЕ СИСТЕМЫ
на тему:
«Функции получения системной информации»

Выполнил:
студент ____ Шамсутдинов Р.Ф. ____
(Ф.И.О.)
группа ____ БВТ2201 ____

Проверил:
____ Королькова Т. В. ____
(Ф.И.О., должность преподавателя)

Оценка ____

Дата ____

Введение

Цель работы: получение практических навыков получения системной информации с помощью функций API в Linux.

Задание:

Разработать приложение, обеспечивающее получение следующей системной информации:

- Имя компьютера, имя пользователя;
- Версия операционной системы;
- Системные метрики (не менее 3-х);
- Функции для работы со временем (не менее 2-х);
- Дополнительные API-функции: 4 функции по выбору.

Примечание: для получения системной информации для последнего пункта задания можно воспользоваться как функциями, определенными в системных библиотеках, так, например, считать данные из системных файлов, размещенных в /sys/, либо других директориях.

Ход работы

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/utsname.h>
#include <time.h>
#include <sys/sysinfo.h>
#include <pwd.h>
```

Рис. 1 – подключение заголовочных файлов

```
#define BUFFER_SIZE 256

int main() {
    char hostname[BUFFER_SIZE];
    char username[BUFFER_SIZE];
    struct utsname sys_info;
    time_t current_time;
    struct tm *time_info;
    struct sysinfo si;
```

Рис. 2 – объявление переменных

```

printf("\n=== Системная информация ===\n\n");

// Получение имени компьютера
if (gethostname(hostname, BUFFER_SIZE) == 0) {
    printf("Имя компьютера: %s\n", hostname);
} else {
    printf("Ошибка при получении имени компьютера\n");
}

// Получение имени пользователя
if (getlogin_r(username, BUFFER_SIZE) == 0) {
    printf("Имя пользователя: %s\n", username);
} else {
    printf("Ошибка при получении имени пользователя\n");
}

```

Рис. 3 – получение имени компьютера и имени пользователя

```

// Получение информации об ОС
if (uname(&sys_info) == 0) {
    printf("\n--- Информация об ОС ---\n");
    printf("Операционная система: %s\n", sys_info.sysname);
    printf("Версия ядра: %s\n", sys_info.release);
    printf("Версия ОС: %s\n", sys_info.version);
    printf("Архитектура: %s\n", sys_info.machine);
}

```

Рис. 4 – получение информации о системе

```

// Получение системных метрик
if (sysinfo(&si) == 0) {
    printf("\n--- Системные метрики ---\n");
    printf("Время работы системы: %ld дней, %ld часов, %ld минут\n",
        si.uptime / 86400, (si.uptime % 86400) / 3600, (si.uptime % 3600) / 60);
    printf("Всего ОЗУ: %lu МБ\n", si.totalram / 1024 / 1024);
    printf("Свободно ОЗУ: %lu МБ\n", si.freeram / 1024 / 1024);
}

```

Рис. 5 – получение системных метрик

```
// Работа со временем
time(&current_time);
time_info = localtime(&current_time);
printf("\n--- Информация о времени ---\n");
printf("Текущее локальное время: %s", asctime(time_info));
time_info = gmtime(&current_time);
printf("Текущее время UTC: %s", asctime(time_info));
```

Рис. 6 – получение текущего времени

```
// Дополнительная информация через system()
printf("\n--- Дополнительная информация ---\n");
printf("Информация о процессоре:\n");
system("cat /proc/cpuinfo | grep 'model name' | head -n 1");
printf("\nИнформация о дисковом пространстве:\n");
system("df -h /");
printf("\nСписок активных процессов:\n");
system("ps aux | head -n 2");
printf("\nИнформация о сетевых интерфейсах:\n");
system("ip addr show | grep 'inet ' | head -n 4");

return 0;
}
```

Рис. 7 – получение доп. информации через system()

```

shamsutdinov@virtualbox:~/labs/1$ vim main.c
shamsutdinov@virtualbox:~/labs/1$ gcc main.c -o main && ./main

=== Системная информация ===

Имя компьютера: virtualbox
Имя пользователя: shamsutdinov

--- Информация об ОС ---
Операционная система: Linux
Версия ядра: 6.8.0-40-generic
Версия ОС: #40~22.04.3-Ubuntu SMP PREEMPT_DYNAMIC Tue Jul 30 17:30:19 UTC 2
Архитектура: x86_64

--- Системные метрики ---
Время работы системы: 0 дней, 1 часов, 3 минут
Всего ОЗУ: 7941 МБ
Свободно ОЗУ: 1199 МБ

--- Информация о времени ---
Текущее локальное время: Tue Feb 18 01:29:16 2025
Текущее время UTC: Mon Feb 17 22:29:16 2025

--- Дополнительная информация ---
Информация о процессоре:
model name      : AMD Ryzen 5 6600H with Radeon Graphics

Информация о дисковом пространстве:
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        39G   11G   27G   29% /

Список активных процессов:
USER           PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root             1  0.1  0.1 167852 12892 ?        Ss   00:25   0:05 /sbin/init splash

Информация о сетевых интерфейсах:
  inet 127.0.0.1/8 scope host lo
  inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
shamsutdinov@virtualbox:~/labs/1$

```

Рис. 8 – вывод программы

Заключение

Вывод: проделав работу, мы получили практические навыки получения системной информации с помощью функций API в Linux.