

# Лабораторная работа №1

## Функции получения системной информации

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

В языках программирования C и C++ `unistd` – это заголовочный файл, обеспечивающий доступ к API операционной системы POSIX. Он определен стандартом POSIX.1, основой спецификации Single Unix, и поэтому должен быть доступен в любой POSIX-совместимой операционной системе и компиляторе, который включает Unix и Unix-подобные операционные системы, такие как варианты GNU, дистрибутивы Linux и BSD и macOS, а также компиляторы, такие как GCC и LLVM.

С помощью `<unistd.h>` можно получить системную информацию в Linux-системах.

Примеры функций из `unistd`, предоставляющих системную информацию:

**int gethostname(char \*name, size\_t len);**// - получить имя компьютера

**int getlogin\_r(char \*name, size\_t len);**// - получить имя пользователя

**long sysconf(int name);**// - получить системные метрики

Используя системный вызов **uname** можно получить данные о версии ОС и архитектуре, в которой выполняется программа. Результат возвращается в структуре `utsname`:

**int uname(struct utsname \*buf);**// - получить версию операционной системы

Структура `utsname` определена в `<sys/utsname.h>` и содержит следующие поля:

**char \* sysname;**// — имя операционной системы

**char \* nodename;**// — имя машины

**char \* release;**// — выпуск операционной системы

**char \* version;**// — версия операционной системы

**char \* machine;**// — архитектура команд процессора

Используя `<time.h>`, можно получить системное время для Linux:

**time\_t time(time\_t \*timer);**// — вычисляет текущее время и возвращает время в формате `time_t`

**struct tm \*localtime(const time\_t \*timer);**// - значение таймера разбивается на структуру `tm` и выражается в местном часовом поясе

**struct tm \*gmtime(const time\_t \*timer);**// - значение таймера разбивается на структуру `tm` и выражается в местном часовом поясе

**char \*asctime(const struct tm \*timeptr);**// - возвращает указатель на строку, которая представляет день и время структуры `timeptr`.

Для работы с системными цветами используйте возможности библиотеки **ncurses**.

## Задание на лабораторную работу

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.

## СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист
2. Наименование лабораторной работы, ее цель.
3. Примеры работы приложения (результаты и тексты программ).