**1. Динамическая загрузка библиотек с dlopen**

* **dlopen(lib\_path, RTLD\_LAZY)**: Загружает динамическую библиотеку по пути lib\_path. Флаг RTLD\_LAZY означает, что функции библиотеки загружаются по мере их использования.
* RTLD\_LAZY (ленивая загрузка)

Разрешение символов происходит только при первом использовании.

Если символ не будет найден при вызове, это приведёт к ошибке.

* **Ошибка загрузки**: Если библиотеку не удалось загрузить, dlopen возвращает NULL, и нужно использовать **dlerror()**, чтобы получить описание ошибки.

void \*handle = dlopen("path\_to\_lib", RTLD\_LAZY);

if (!handle) {

fprintf(stderr, "Ошибка: %s\n", dlerror());

return 1;

}

* **Закрытие библиотеки**: После того как библиотека больше не нужна, её нужно закрыть с помощью **dlclose(handle)**.

dlclose(handle);

**2. Загрузка функций через dlsym**

* **dlsym(handle, "function\_name")**: Загружает указатель на функцию по имени в загруженной библиотеке.
* Нужно кастовать указатель на функцию в соответствующий тип, например:

func\_float3 SinIntegral = (func\_float3)dlsym(handle, "SinIntegral");

func\_float2 Derivative = (func\_float2)dlsym(handle, "Derivative");

* **Обработка ошибок**: Если функция не найдена, **dlsym** вернёт NULL, и нужно снова вызвать **dlerror()**.

КопироватьРедактировать

if (!SinIntegral || !Derivative) {

fprintf(stderr, "Ошибка загрузки функций: %s\n", dlerror());

dlclose(handle);

return 1;

}

**3. Использование typedef для указателей на функции**

* **typedef** используется для удобства, чтобы задать тип указателя на функцию. Например:

typedef float (\*func\_float3)(float, float, float);

typedef float (\*func\_float2)(float, float);

* Это упрощает работу с динамическими функциями, поскольку мы создаём псевдонимы для типов указателей на функции.

**4. Обработка ввода с strtok и strtof**

* **strtok** используется для разбиения строки на токены. Важно правильно управлять разделителями и проверять, что каждый токен соответствует нужному типу.

char\* token = strtok(input, " ");

* **strtof** — преобразует строку в число с плавающей запятой. Используйте **endptr**, чтобы отслеживать, что строка корректно преобразована.

float num = strtof(token, &endptr);

if (\*endptr != '\0') {

printf("Ошибка: '%s' не является числом\n", token);

}

**5. Переключение между библиотеками**

* Для переключения между двумя библиотеками достаточно использовать логическое выражение, чтобы выбрать нужную библиотеку:

current\_lib = (current\_lib == 1) ? 2 : 1;

* После этого нужно загрузить новую библиотеку с **dlopen** и закрыть старую с **dlclose**.

**Ключевые моменты:**

* **dlopen**: Для загрузки библиотеки.
* **dlsym**: Для получения указателей на функции из библиотеки.
* **dlclose**: Для закрытия библиотеки.
* **typedef**: Для создания удобных типов для указателей на функции.
* **strtok** и **strtof**: Для обработки ввода и преобразования строк в числа.
* **Обработка ошибок**: Важно проверять ошибки с **dlerror()** и **endptr**.