**Московский Авиационный Институт**

**(Национальный Исследовательский Университет)**

**Факультет информационных технологий и прикладной математики**

**Кафедра вычислительной математики и программирования**

**Лабораторная работа №5 по курсу**

**«Операционные системы»**

**СОЗДАНИЕ ДИНАМИЧЕСКИХ БИБЛИОТЕК.  
СОЗДАНИЕ ПРОГРАММ, КОТОРЫЕ ИСПОЛЬЗУЮТ ФУНКЦИИ ДИНАМИЧЕСКИХ БИБЛИОТЕК.**

**Студент: Баранников Степан Алексеевич**

**Группа: М80–201Б-18**

**Вариант: 03**

**Преподаватель: Соколов Андрей Алексеевич**

**Оценка: \_\_\_\_\_\_\_\_\_\_\_**

**Дата: \_\_\_\_\_\_\_\_\_\_\_**

**Подпись: \_\_\_\_\_\_\_\_\_\_\_**

**Москва, 2019**

**1.** **Постановка задачи**

Требуется создать динамическую библиотеку, которая реализует определенный функционал. Далее использовать данную библиотеку 2-мя способами:

1. Во время компиляции (на этапе «линковки»/linking)

2. Во время исполнения программы, подгрузив библиотеку в память с помощью системных вызовов

В конечном итоге, программа должна состоять из следующих частей:

• Динамическая библиотека, реализующая заданных вариантом интерфейс;

• Тестовая программа, которая используют библиотеку, используя знания полученные на этапе компиляции;

• Тестовая программа, которая использует библиотеку, используя только местоположение динамической библиотеки и ее интерфейс.

Провести анализ между обоими типами использования библиотеки.

**Структура данных:** Стек

**Тип данных:** MD5-суммы

**2.Описание**

Интерфейс очереди описан в **head.h**, сами функции очереди описаны в **head.c** (добавления и удаления из стека, первый элемент).

В исходном коде **main.c** предоставляется пример программы без использования библиотеки динамической линковки. Его можно запустить после компиляции с объектным кодом очереди или после линковки при помощи флага **-L** (при этом необходимо добавить путь к динамической библиотеке используя команду в файле **set\_path.sh**).

В исходном коде **main\_dyn.c** предоставляется пример программы с использованием библиотеки динамической линковки **dlfcn.h**. Указатели на функции загружаются функциями **dlsym**. При этом требуется использование дополнительного флага -**ldl** для **dlfcn.h**.

**3.Листинг Makefile**

|  |
| --- |
| LIBRARY\_NAME = head LIBRARY\_FLAGS = -Wall BINARY\_FLAGS = -Wall  .PHONY: all all: dynamic dynamic\_sys  dynamic: dynamic\_lib main.c  gcc $(BINARY\_FLAGS) main.c -l$(LIBRARY\_NAME) -L. -o executable  dynamic\_sys: dynamic\_lib main\_dyn.c  gcc $(BINARY\_FLAGS) main\_dyn.c -ldl -o executable\_dyn  #No dependencies since version control needs to be performed manually dynamic\_lib:  gcc $(LIBRARY\_FLAGS) -c $(LIBRARY\_NAME).c  gcc $(LIBRARY\_FLAGS) -shared -o lib$(LIBRARY\_NAME).so $(LIBRARY\_NAME).o  .PHONY: clean clean:  rm -f \*.o \*.so executable executable\_dyn |

**4.** **Вывод**

В ходе ЛР оказалось, что библиотеки можно компилировать не только на этапе компиляции, после чего линковать с главными файлами, но и компилировать во время рантайма программы, что позволяет нам, например, не компилировать всю библиотеку, а пользоваться лишь нужнымы для нас функциями.