**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Операционные системы»

**Лабораторная работа № 5**

Тема: dll, run time linking, direct linking

Студент: Гамов Павел

Группа: 80-207

Преподаватель: Миронов Е.С.

Дата:

Оценка:

Москва, 2019

**1.**               **Постановка задачи**

Задача – реализовать стек и создать из него динамическую библиотеку. Создать программы, в которых мы будем использовать неявное и явное использование динамической библиотеки.

**2.**           **Описание программы**

Реализуем функционал стека, пару функций pop, push, print, create, delete. Дальше скомпилируем его сначала в .о, потом в .so, что является форматом динамических библиотек. В папке Dlink используем неявное использование, в папке Dload явное, что это означает? В первом случае библиотека загружается в память, где доступна всем используемым процессам, таким образом нет необходимости загружать копию в каждый отдельный пейдж, что существенно снижает затраты памяти на использование огромных библиотек. Во втором случае нам придется явно указывать откуда и какую функцию нам придется подгружать, таким образом мы переписываем функции, где сначала подгружаем нужную функцию, передаем ей аргументы, дальше возвращаем результат, перед этим отгружаем библиотеку обратно. Важный аспект, что библиотеку полностью выгрузить мы не сможем, только освободить. Так как из-за аспектов каждой операционной системы полная выгрузка происходит где-то за кадром, потому что внезапно может возникнуть процесс, который будет ее использовать. Поэтому это бывает хорошо, когда ресурс используется часто, но плохо, когда очень редко и, следовательно, программист теряет контроль над ресурсами.

Также важным плюсом использования dll при больших проектах, то что при изменении внутренних алгоритмов не надо перекомпилировать проект полностью, достаточно скомпилировать заново только библиотеку.

**3.**           **Выводы**

Каких-то значительных и очевидных минусов у данной системы нет, даже больше плюсов как по мне. Так что часто проще довериться разработчикам ПО, потому что эти люди знают, что делают.

Из минусов могу отметить, что явная загрузка функционала достаточно долгая, так что часто проще заранее продумать и подгрузить нужные функции, а дальше освободить их при завершении программы.

Также существует проблема совместимости функций С++, где компилятор меняет название функций, да так что приходится компилировать со словом extern “C”.