Лабораторная №1 Компилятор, компоновщик и динамические библиотеки

Написать программу, использующую три библиотеки: zlib (http://www.zlib.net/), libpng (http://www.libpng.org/pub/png/), freetype (http://www.freetype.org/). Программа должна осуществлять создание png файла с нарисованным в нем текстовым сообщением с помощью шрифта, загруженного из файла *.ttf.

При написании программы должны быть выполнены несколько условий:

1. Программа должна компилироваться с помощью gcc и системы makefile в ОС Linux или WSL.
2. Программа должна быть выполнена в двух версиях:
   a. static - статическая компоновка всех трех библиотек.
   b. dynamic – динамическая загрузка всех трех библиотек в начале работы с помощью dlload.
   c. blob – реализация всего функционала в виде блоба и загрузка его с диска с помощью загрузчика elf-loader (https://code.google.com/p/elf-loader/), который должен быть включен в состав проекта программы.
3. Система Makefile должна поддерживать следующие команды:
   a. make static – компиляция исходников в версии со статической компоновкой;
   b. make dynamic – компиляция исходников в версии с динамической компоновкой;
   c. make blob – компиляция исходников в версии с компоновкой в виде блоба;
   d. make clean – удаление всех бинарных файлов;
   e. make all – удаление старых бинарных файлов и компиляция исходников в версиях static, dynamic и blob;

4. Программа в независимости от типа сборки по результатам своей работы должна не только генерировать png файл с нарисованным текстовым сообщением в нем, но и печатать на экран общее время своей работы и время, затраченное на загрузку системы (от начала main до конца своей работы).

5. Файл со шрифтом, имя png файла и печатаемый текст должны поступать в программу из аргументов командной строки.

6. Программа не должна генерировать исключения или преждевременно завершать свою работу в независимости от аргументов командной строки.

Состав отчета

1. Формулировка задания
2. Проделанная работа
3. Структура каталогов
4. Код makefile
5. Достоинства и недостатки статической и динамической компоновки, оформленные в виде сравнительной таблицы
6. Указания в каких случаях рекомендуется использовать динамическую, а в каких статическую компоновку
7. Выводы

Рекомендации

1. Рекомендуется использовать следующую систему каталогов:
    a. lab1\ - каталог со всеми исходными текстами и скриптами сборки
    b. lab1\zlib\ - исходные тексты и makefile для библиотеки zlib
    c. lab1\freetype\ - исходные тексты и makefile для библиотеки freetype
    d. lab1\libpng\ - исходные тексты и makefile для библиотеки libpng
    e. lab1\app\ - исходные тексты и makefile для приложения
    f. lab1\makefile – корневой makefile

2. Для анализа получившихся ELF файлов рекомендуется использовать утилиту readelf

3. Для удобной работы с программой в версиях static, dynamic и blob рекомендуется использовать суффиксы для имен итоговых бинарных файлов, например: app-static, app-dynamic, app-blob

4. Для использования одного кода для трёх типов компоновки рекомендуется использовать опцию компилятора, например –D_DYNAMIC_ и соответствующие конструкции #ifdef _DYNAMIC_ /#endif для исключения кода в случае статической компоновки.

5. Общий алгоритм приложения с main для динамической компоновки:
    a. Запомнить текущее время во временной переменной
    b. Обработать аргументы командной строки
    c. Загрузить библиотеку zlib
    d. Заполнить таблицу с экспортируемыми функциями из zlib
    e. Загрузить библиотеку libpng
    f. Заполнить таблицу с экспортируемыми функциями из libpng
    g. Заполнить таблицу с импортируемыми функциями для libpng используя таблицу экспортируемых функций zlib
    h. Загрузить библиотеку freetype
    i. Заполнить таблицу с экспортируемыми функциями из freetype
    j. Заполнить таблицу с импортируемыми функциями для freetype используя таблицу экспортируемых функций zlib
    k. Проинициализировать библиотеку libpng
    l. Проинициализировать библиотеку freetype
    m. Распечатать общее время загрузки приложения как текущее время – сохраненное время
    n. Создать картинку png в памяти с помощью функций libpng
    o. Получить указатель на память с пикселями созданной картинки
    p. Нарисовать текст в картинку с помощью функций freetype
    q. Сохранить png картинку

r. Распечатать общее время работы приложения как текущее время – сохраненное время

6. Для типа сборки blob рекомендуется собрать из трех библиотек и основного кода программы файл блоба app.blob, который:
   a. Будет формата ELF
   b. Будет собираться с опциями -pie и -fPIE
   c. Будет обладать одной точкой входа, которая и будет выполнять необходимые действия
   d. Будет без таблиц импорта и экспорта
   e. В качестве аргументов для точки входа будет передаваться структура с параметрами программы (именем файла шрифта, именем png, строкой), а так же таблицей указателей на необходимые функции (fopen, fread, fwrite,…)
7. Для сборки типа blob собирается отдельно блоб с основной функциональностью, а сама программа работает по алгоритму:
   a. Запомнить текущее время во временной переменной
   b. Обработать аргументы командной строки
   c. Сформировать структуру, для передачи ее в блоб
   d. Загрузить блоб с помощью elf-loader
   e. Получить указатель на точку входа в блобе
   f. Вызвать точку входа
   g. Распечатать общее время работы приложения как текущее время – сохраненное время

Ресурсы

Исходные тексты:

- http://www.zlib.net/
- http://www.libpng.org/pub/png/
- http://www.libpng.org/pub/png/
- https://code.google.com/p/elf-loader/

- http://www.freetype.org/
- https://labs.xvid.com/

Теоретические сведения:

- http://habrahabr.ru/post/155201/
- http://habrahabr.ru/post/150327/
- http://habrahabr.ru/post/106107/